COMMUNICATION IN A WORLD OF PERVASIVE SURVEILLANCE

Sources and methods:
Counter-strategies against pervasive surveillance architecture

Jacob R. Appelbaum

Communication in a world of pervasive surveillance

Sources and methods:

Counter-strategies against pervasive surveillance architecture

Jacob R. Appelbaum

Copyright: © Jacob R. Appelbaum
E-mail: jacob@appelbaum.net
Website: https://www.appelbaum.net/
Thesis typeset with LaTeX and BibTeX.

First edition, second printing: March 2022

This research was supported by the Netherlands Organisation for Scientific Research (NWO) grant 639.073.005 ("Computational number theory in cryptography and cryptanalysis").

Printed by Gildeprint Drukkerijen, Enschede, The Netherlands.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-5471-3
NUR: 980

Cover composition: anonymous.

Communication in a world of pervasive surveillance

Sources and methods:
Counter-strategies against pervasive surveillance architecture

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus Prof. dr. ir. F.P.T. Baaijens, voor een commissie
aangewezen door het College voor Promoties, in het openbaar te verdedigen
op 25 maart 2022 om 16:00 uur.

door

Jacob R. Appelbaum

geboren te [redacted], Californië, Verenigde Staten

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. M.G.J. van den Brand
1e promotor:	prof.dr. D.J. Bernstein
2e promotor:	prof.dr. T. Lange
leden:	dr. B.M.M. de Weger
	prof.dr. P. Rogaway (UC Davis, Computer Science)
	prof.dr. N. Koblitz (University of Washington, Mathematics)
	prof.dr. C. Grothoff (Bern University of Applied Sciences, Engineering and Information Technology)
	prof.dr.ir. J. Schwenk (Ruhr-University Bochum, Electrical Engineering)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Preface

> *"If this be treason, make the most of it!"*
>
> — Patrick Henry*, on his* ***twenty-ninth*** *birthday*

This thesis is the culmination of more than a decade of research into the topic of surveillance and the uses of data collected through surveillance. The research that follows includes discussions with insiders and analysis of both previously published and unpublished information. We still lack full information on many topics, notably the names of perpetrators. This has several causes: the nature of the topics covered, the legally threatening markings on many documents, and the political power of those who would suppress publication. There can be considerable personal consequences for following this direction of research. Several colleagues face serious legal, political, social, and health issues resulting from their participation in, and contributions to, this research topic.

Many aspects of this research started as investigative journalism rather than science. Documents first published by news organizations under the byline of the author of this thesis are reproduced here in full and credited appropriately. Sensitive, classified, or otherwise secret internal documents are provided to ensure that their content is witnessed firsthand, to make them freely accessible on the Internet and in libraries, and to ensure that they are not erased from history.

The perspective in this thesis is necessarily dominated by the United States of America, whose activities impact nearly every person on planet Earth. The focus on America is deeply political: it is the moral duty of every citizen of the United States to address serious faults in policy and to assist in the process of accountability. Democratic discussion covering technical and non-technical topics of various government or corporate activities is important and necessary. The evidence and findings discussed in this thesis touch on myriad controversial issues ranging from political spying on world leaders to drone assassination of human beings who faced no legal charges and are afforded no day in court.

The sheer number of the surveillance systems that we document in subsequent chapters reflects the industrial scale of data collection in the twenty-first century. We hope that future researchers will take up the challenge of addressing each covert program as a research subject to fully and completely explore, and to freely share their findings with the wider world in the spirit of open academic discussion. This kind of basic research is crucial to anti-surveillance software and hardware development. One example is the general idea of the Mix network (mixnet), an anonymity mechanism designed to withstand very powerful adversaries who possess a long memory. How might the evolution of mixnets be shaped by understanding the concrete systems that attack privacy and anonymity infrastructure? Researchers may even feel inspired to build their own countermeasures, and perhaps full solutions, that encompass more than the purely technical. We offer several examples of such solutions in the chapters that follow. By applying mathematics and computer science to build countermeasures to surveillance systems, we can protect people individually and at scale, reducing these systems to historical footnotes.

Mass surveillance programs present a temptation so great that even very intelligent people imagine the trade-offs to be worthwhile. Many people cannot imagine a future in which their government is blatantly corrupt, or has indeed collapsed. Yet history teaches unambiguously that such changes may come quickly, unexpectedly, and those who seek to exploit the entropic nature of the situation will use all technical, social, economic, and political levers to accomplish their goals. This knowledge should, but often does not, temper support for mass surveillance; this is a blind spot that is not to be dismissed lightly.

The machinery of mass surveillance is simply too dangerous to be allowed to exist. We must work to ensure that no one will be able to say that they did not know, or that they were not warned. We must use all of the tools in our toolbox – economic, social, cultural, political, and of course, cryptographic – to blind targeted and mass surveillance adversaries. The goal is justice [Pon11, "The method is transparency, the goal is justice."] and this thesis encourages a method of designing, building, deploying, and using cryptographic protocols centered around human liberty to ensure it.

Acknowledgments

I wish to express love, appreciation, and gratitude to my wife and my family around the world. Extra special thanks to Leif Ryge who always challenges me to do better.

Sincere thanks to Daniel J. Bernstein and Tanja Lange; you are great advisors. I am sincerely humbled and deeply appreciative for your support and encouragement. I feel that there are no adequate words in any language to express my appreciation for how you have helped change my life for the better. Still, I will try: thank you very much.

Thanks to my committee members Benne de Weger, Phil Rogaway, Neal Koblitz, Christian Grothoff, and Jörg Schwenk for their valuable feedback, and to Mark van den Brand for chairing the committee. Thank you again, Daniel J. Bernstein and Tanja Lange for bringing this special committee together.

Thank you again to Benne de Weger. You were welcoming and encouraging of my interests in cryptography more than a decade ago when we worked together on the practical exploitation of MD5 in the wild.

During my time at TU/e, I feel extremely privileged to have shared some thoughtful discussions with Phil Rogaway during his visit to Eindhoven and when we overlapped at conferences such as Asiacrypt. You are one of the most humble and kind people I have ever had the pleasure of meeting, and you set a standard by personal example that is difficult to capture in words. You have been kind, insightful, and deeply encouraging of formalizing cryptographic notions and in being extremely rigorous in those definitions. Thank you again Phil.

When working at the University of Washington around a decade ago, Neal Koblitz and I shared a memorable lunch. You encouraged me to focus on mathematics, and to move away from the captured interests of corporate and government funding of computer science. I am glad that I listened to your advice Neal, and it has changed my life for the better. Thank you for taking the time to guide me at a critical time in my life.

Thank you again Christian Grothoff for encouraging me to think critically about society, about the role of Free Software, and about how we might all contribute to the world in a positive manner. Especially I wish to thank you for your long and detailed feedback on every single page of this thesis; it is a stronger work because of your attention to detail.

Thank you to all of my colleagues in Eindhoven – working with you has been humbling and respectful, I am in awe of the incredible research and work that comes from the CC and CI groups at TU/e. Thank you to all of my coauthors, especially the coauthors of the final four chapters of this thesis. Thank you to my Master's students Björn Ruytenberg and Peter Wu for the privilege of supervising you both; I am humbled by your generosity and your talent, thank you both for the time we spent working together.

I could not have finished without support from many people. Thank you to all who wish to remain anonymous. Thank you to each of the people in the following alphabetically sorted list. Thank you for your positive influence in my life during my PhD research time. Especially thanks to those who encouraged me to pursue and to complete a PhD at TU/e:

Aaron Gibson, Adam Langley, Ai Weiwei, Alex Le Heux, Andras Kristof, Andreas Hülsing, Andy Müller-Maguhn, Angel Alonzo, Angela Richter, Anonymous, Berit Gilma, Bill Binney, Brennan Novak, Brian Aker, Bruce Leidl, Caspar Bowden, Chitchanok Chuengsatiansup, Chloe Martindale, Christian David, Christine Aker, Christopher Sheats, Christy Lange, Daniel Ellsberg, Daniel Neves, Daniel Yeow, David Ahmed, David Fine, David McKinney, David Miranda, David Robinson, David Stainton, Edward Snowden, Emmett Corman, Erik Visse-Martindale, Eva Blum-Dumontet, Eva Infeld, Felicity Ruby, Florian Weber, Frank Rieger, Garen Kessel, Gavin MacFadyen, Geraldine De Bastion, Gerhardt Isringhaus, Glenn Greenwald, Gustavo Banegas, Harald Welte, Harry Halpin, Holger Stark, Ian Ryge, Ingmar Zahorsky, Ingy döt Net, J. Alex Halderman, Jan Bultmann, Janine Römer, Jean Peters, Jeff Burdges, Jérémie Zimmermann, Jesús-Javier Chi-Domínguez, Joe Dibee, Joe Joe Wong, Johannes Wahlstrom, John Gilmore, John Goetz, John Perry Barlow, Jonathan Levin, Jonathan Wilkins, Juan Branco, Juan Passarelli, Julian Assange, Karsten Nohl, Kate Young, Katharina Wurdack, Kathrin Hövelmanns, Katie Miranda, Kelly Caine, Kent Bozlinski, Kimberley Thissen, Kit Smeets, Laura Brown, Laura Poitras, Leon Lupo, Lorenz, Luca Barbeni, Lucky Green, Maike Welte, Mairon Mahzoun, Malek Azaz, Marc Bruyere, Marcel Rosenbach, Margaret Kunstler, Marianne Le Heux, Mark Atwood, MC, Michael Ellsberg, Michael Ratner, Mitch Altman, Monika Ermert, Morgan Weaver, Nadia Heninger, Nadim Kobeissi, Nadja Vancauwenberghe, Naomi Colvin, Patrick D. Anderson, Pepijn Le Heux, Pepper Aker, Peter Gilmore, Peter Todd, Renata Ávila, Richard Stallman, Robin Kwant, Roger Corman, Rop Gonggrijp, S, Scott Ludlam, Sean Bonner, Sean Vegezzi, Soren Ragsdale, Stefania Maurizi, Stella Morris, Stewart Smith, Suelette Dreyfus, Sven Guckes, Tal Siano, Tatiana Bazzichelli, Terry James Albury, Tillmann Heier, Tim Jenkin, Tim Kuijsten, Tomer Ashur, Trevor Paglen, Vera Wilde, Virgil Griffith, Will Scott, and Yazz Atlas.

In Berlin there is a statue of Heinrich Heine that I visited often during my research time. The statue quotes Heine as observing the motivation and underlying cause of struggles of his era; his observation still holds true today:

> "*Wir ergreifen keine Idee, sondern die Idee ergreift uns und Knechtet uns und peitscht uns in die Arena hinein, dass Wir wie gezwungene Gladiatoren für sie kämpfen.*" [1]

Jacob R. Appelbaum, Berlin, March 2022

---

[1] *"We do not seize an idea, but the idea seizes us and subjugates us and whips us into the arena so that we fight for it like forced gladiators."*

Contents

Preface v

Acknowledgments vii

Contents ix

List of Figures xii

List of Tables xiv

List of Algorithms xv

1 Introduction 1
- 1.1 A fifth level of ethics in mathematics 4
- 1.2 Thinking about the future 8
- 1.3 Organization of this thesis 13

2 Background on network protocols 15
- 2.1 Free Software, Open Hardware, Operational Security 17
- 2.2 Layers of the Internet 18
- 2.3 Ethernet networks and the Internet Protocols 18
- 2.4 The Domain Name System 19
- 2.5 Multicast Domain Name System (mDNS) 19
- 2.6 Hypertext Transport Protocol (HTTP) 20
- 2.7 Transport Layer Security (TLS) 20
- 2.8 Virtual Private Networks (VPN) 20

3 Background on cryptography 23
- 3.1 Mathematics as informational self-defense 25
- 3.2 Notation 25
- 3.3 Hashing 26
- 3.4 Symmetric Encryption: block cipher 27
- 3.5 Symmetric Encryption: stream cipher 27
- 3.6 Message Authentication Code 27
- 3.7 Authenticated-Encryption with Associated-Data (AEAD) 28
- 3.8 Non-interactive Key Exchange (NIKE) 29

3.9 Verification of public keys 30
3.10 Signatures 31
3.11 Protocols from building blocks 33

4 The Adversary **35**
4.1 Zersetzung or Dirty Tricks? 43
4.2 Foundational events and disclosures in surveillance 48
4.3 Summer of Snowden and the post-Snowden Era 70
4.4 Standardization of cryptographic sabotage 81
4.5 XKeyscore 85
4.6 ANT catalog 92
4.7 Conclusions 146

5 The GNU name system **147**
5.1 Introduction 149
5.2 Background: Domain Name System (DNS) 150
5.3 Security goals 151
5.4 Exemplary Attacker: The NSA's MORECOWBELL and QUANTUMDNS programs 152
5.5 Adversary Model 155
5.6 Domain Name System Security Extensions (DNSSEC) 157
5.7 Query name minimization 158
5.8 DNS-over-TLS 159
5.9 DNSCurve 160
5.10 Confidential DNS 161
5.11 Namecoin 163
5.12 The GNU name system 164
5.13 Assessment 166
5.14 Conclusions 167

6 Tiny WireGuard Tweak **169**
6.1 Introduction 171
6.2 Realistic adversary concerns 171
6.3 WireGuard overview 172
6.4 Traffic analysis 174
6.5 Security and privacy issues 178
6.6 Blinding flows against mass surveillance 181
6.7 Conclusions 183

7 Vula **185**
7.1 Introduction 187
7.2 Background and related work 188
7.3 Threat Model and design considerations 192
7.4 Detailed Protocol Description 198
7.5 Performance 211
7.6 Security Evaluation 213
7.7 Conclusions 218

8 REUNION **219**
8.1 Introduction 221
8.2 Background and related work 222
8.3 Introducing REUNION 230
8.4 Threat Model 243
8.5 Security Evaluation 244
8.6 Implementations 248
8.7 Future Work 249
8.8 Conclusions 249

Bibliography **251**

General index **309**

Glossary **327**

Summary **331**

Curriculum Vitae **333**

List of Figures

1.1 U.S. domestic FISA intercept of OTR messages without decryption 10

3.1 Example: Mallory in the middle 30

4.1 TEMPORA: Big data ... Big access 38
4.2 What is TEMPORA? 39
4.3 SORM network floor map 40
4.4 JTRIG: identifying & exploiting fracture points 43
4.5 JTRIG: DISRUPTION Operational Playbook 44
4.6 JTRIG: Map of technologies to message delivery 45
4.7 JTRIG: EFFECTS: Definition 46
4.8 JTRIG: Discredit a target 47
4.9 JTRIG: Discredit a company 48
4.10 Identity intelligence 50
4.11 CounterSpy magazine statement of motivation 52
4.12 What is XKEYSCORE? 55
4.13 Where is X-KEYSCORE 56
4.14 NSA domestic surveillance: AT&T room 641A 59
4.15 Central Intelligence Agency Information Operations Center logo 62
4.16 Athena Technology Overview 65
4.17 Central Intelligence Agency partial organizational chart 69
4.18 Hardware implant added to Andy Müller-Maguhn’s IP19 Cryptophone . . . 71
4.19 PRISM: FAA702 Operations: Two Types of Collection 74
4.20 PRISM: FAA702 Operations: Why Use Both: PRISM vs. Upstream 75
4.21 Geography of FAIRVIEW 76
4.25 Dates When PRISM Collection Began For Each Provider 76
4.22 "SSL added and removed here! :-)" 77
4.23 PRISM Tasking Process 78
4.24 PRISM Collection Dataflow 79
4.26 (S//SI/REL) XKEYSCORE definition 85
4.27 IRONCHEF 93
4.28 DEITYBOUNCE 94
4.29 GODSURGE 95
4.30 JETPLOW 97
4.31 HALLUXWATER 98
4.32 FEEDTROUGH 99

4.33 GOURMETTROUGH . . . 100
4.34 SOUFFLETROUGH . . . 101
4.35 HEADWATER . . . 102
4.36 SIERRAMONTANA . . . 104
4.37 SCHOOLMONTANA . . . 105
4.38 STUCCOMONTANA . . . 106
4.39 CTX4000 . . . 107
4.40 LOUDAUTO . . . 108
4.41 NIGHTWATCH . . . 109
4.42 PHOTOANGLO . . . 110
4.43 TAWDRYYARD . . . 111
4.44 CROSSBEAM . . . 112
4.45 CANDYGRAM . . . 113
4.46 CYCLONE Hx9 . . . 114
4.47 EBSR . . . 115
4.48 ENTOURAGE . . . 116
4.49 GENESIS . . . 117
4.50 NEBULA . . . 118
4.51 TYPHON-HX . . . 119
4.52 WATERWITCH . . . 120
4.53 NIGHTSTAND . . . 121
4.54 SPARROW II . . . 122
4.55 DROPOUTJEEP . . . 123
4.56 GOPHERSET . . . 124
4.57 MONKEYCALENDAR . . . 125
4.58 TOTECHASER . . . 126
4.59 PICASSO . . . 127
4.60 TOTEGHOSTLY . . . 128
4.61 RAGEMASTER . . . 129
4.62 SURLYSPAWN . . . 130
4.63 GINSU . . . 131
4.64 IRATEMONK . . . 132
4.65 SWAP . . . 133
4.66 WISTFULTOLL . . . 134
4.67 HOWLERMONKEY . . . 135
4.68 JUNIORMINT . . . 136
4.69 MAESTRO-II . . . 137
4.70 SOMBERKNAVE . . . 138
4.71 TRINITY . . . 139
4.72 COTTONMOUTH-I . . . 140
4.73 COTTONMOUTH-II . . . 141
4.74 COTTONMOUTH-III . . . 142
4.75 FIREWALK . . . 143

5.1 Resolving the name `www.example.com` with DNS . . . 151
5.2 NSA's MORECOWBELL infrastructure . . . 153
5.3 MORECOWBELL: A Covert HTTP/DNS Monitoring System . . . 154

5.4 What is MORECOWBELL . . . 154
5.5 How does MORECOWBELL work? . . . 155
5.6 "Benefits" of MORECOWBELL . . . 155
5.7 NSA's QUANTUMTHEORY . . . 156
5.8 NSA's QUANTUMDNS . . . 156
5.9 Resolving the name `www.example.com` with DNS and DNSSEC . . . 158
5.10 DNS query name minimization example . . . 159
5.11 DNSCurve resolution example . . . 162
5.12 Opportunistic Confidential DNS name resolution example . . . 163
5.13 Namecoin . . . 164
5.14 The GNU Name System . . . 165

6.1 Informal protocol narration of WireGuard handshake . . . 173
6.2 Flow graph between two WireGuard peers as seen in Wireshark . . . 176
6.3 148 byte initiator packet payload . . . 177
6.4 92 byte responder packet payload . . . 177
6.5 64 byte Cookie packet payload . . . 177
6.6 Variable length (32 up to $\infty + 16$) byte data packet payload . . . 178
6.7 Informal protocol narration of sending and receiving an initiator packet. . . 178
6.8 Tweaked initiator packet (in bytes) . . . 182

7.1 Vula peers as shown by `vula peer -show` . . . 195
7.2 Incoming descriptor processing state engine . . . 200
7.3 FLENT measurement graph (AMD64 and AARCH4) . . . 207
7.7 FLENT measurement graph with and without WireGuard . . . 212
7.8 FLENT measurement graph with and without WireGuard . . . 213

8.1 PANDA full exchange . . . 224

List of Tables

4.1 Alleged and confirmed NSA domestic interception locations . . . 64

5.1 Comparison of the defenses offered by the various designs and their relative deployment complexity. . . . 167

6.1 Theoretical maximum sizes for UDP payloads . . . 178

7.1 Comparison of properties . . . 191
7.2 Vula descriptor key, value pair examples . . . 199
7.3 Python CSIDH-512 performance (x86_64) . . . 211

7.4 Python CSIDH-512 performance (AARCH64, ARMV7L) 211
7.5 Python CSIDH-512 performance (PPC64LE, RISCV64) 211

8.1 Adversary attack strategies 245
8.2 Evaluation of Adversary attack strategies 246
8.3 Per-user and total network bandwidth cost estimates per epoch 248

List of Algorithms

6.1 Simplified WireGuard key agreement process 175
6.2 Extract Initiator's Long-term Static Key Pair 180
6.3 Extract User U_n's Long-term Static Key Pair 180
8.1 PANDA protocol 225
8.2 REUNION protocol 234

CHAPTER 1

Introduction

> "*Wer die Wahrheit nicht weiß, der ist bloß ein Dummkopf. Aber wer sie weiß und sie eine Lüge nennt, der ist ein Verbrecher.*" [1]
>
> — Bertold Brecht, ***Das Leben des Galilei***, *Seite 71*

Electronic surveillance systems, in their twenty-first century totality, create an environment of *pervasive surveillance* where most, if not all, communications channels are monitored in some capacity. Sociologists and other academic researchers define surveillance in many different ways [Mar15]. We consider the definition from Lyon from Surveillance Studies: "*any systematic, routine, and focused attention to personal details for a given purpose (such as management, influence, or entitlement)*" [Lyo14]. Today's Internet is the primary terrain of struggle [GBC11, Kat90, Her00, Ziz08, Cun15, GE07] between those committed to attacking electronic communications, whether in targeted [Bam16] surveillance of individuals or indiscriminate mass surveillance [Eur18, Eur78, Eur06, Eur84, Eur10, Eur87, Eur15, Eur16] of whole populations, and those committed to securing communications from attack.

The two most prevalent surveillance adversaries are state [Gre14b] and corporate [Zub19, Int21a, Int21b] actors, though in some situations there is no meaningful distinction between these. *Fusion Centers* [Wik21i] for example, are an American domestic intelligence apparatus that aggregates data provided by government agencies, corporations, and private persons, resulting at times in Americans being persecuted for engaging in constitutionally protected activities. Surveillance data of all kinds collected from other terrains [Goo21, War15b] readily merges into the Internet's IP traffic flows. This collection is not merely through passive observation of our communications, but also through active interaction and exploitation, along with analysis of behavioral data, other systems data, and data at rest. To name just a few examples:

- In-person, face-to-face meetings when personal or professional electronic equipment is present in the same room [ATL06, CCTM16].
- Targeted and mass surveillance of telephone metadata and call content [SM13, GS14].
- Targeted and mass surveillance of postal mail [Nix13].
- Public and private video surveillance, especially when used in tandem with machine learning for identification based on height, gait, and/or facial structure among others [EKGBSBA16].
- Stylometry of written text to identify anonymous authors [BAG12].
- Analysis of video and images of biological structures such as veins, ear shape, as well as of body modifications such as piercings and tattoos [RP14].

As new sources of data become available in nearly every realm of life, we find new surveillance tools being designed to exploit them. Understanding these surveillance practices is critical for building defenses.

It is now commonly understood that the US Government does "*kill people based on metadata*" [Col14] including children [Sca13a, Bon13, Kri19, AR21], intentionally [2] and unintentionally. The state's capacity for violence is enhanced with additional surveillance capabilities. Historical as well as contemporary use of data and metadata to socially sort [Lyo03] has enabled human rights abuses such as persecuting political refugees [vZCMS17, DNI21], assassinations [Col14] and genocide [Bla12].

---

[1] "*He who does not know the truth is merely a fool. But whoever knows it and calls it a lie is a criminal.*"

[2] The President of The United States of America is directly involved in some assassination decisions [Poi14, Par15], something of an explicit concern [Ken11] to the founders of the country.

Modern proponents of both targeted and mass surveillance regularly claim that granting authorities surveillance powers will help to prevent terrorist acts. We know that while this is sometimes true [EM13, BSSC14], it is often false, with disastrous consequences [GRS14, Rot15]. We also know that the existence of interception capabilities puts both the operators [Bam16] and users of communication infrastructure at direct risk, and that the same surveillance methods intended for terrorists are diverted to targeting democratically elected leaders [JAS13]. This leads us to ask: In order to protect our societies from terrorist acts, must we leave ourselves vulnerable? Is it worth the trade-off to occasionally catch the least competent would-be terrorists, corrupt officials, spies, criminals, and thieves? The questions themselves seem absurd when the answer promotes criminality of all kinds: corporate espionage, economic warfare, government espionage, human-rights violations, lawfare, so-called "targeted killings" (assassinations), untargeted killings, etc. Yet an affirmative answer to those questions is an observable national policy in countries around the world.

The deployment of standardized communications protocols in the last century made it possible to perform surveillance in a highly automated fashion. We investigate some of these surveillance systems extensively with help from documents exposed by whistleblowers, known and unknown, or other anonymous insiders. We compare the intentions and stated beliefs of surveillance adversaries with those of protocol designers, who in recent years have belatedly started to introduce the term surveillance, and later mass surveillance, into Internet-related protocol publications [FT14, BSJ$^+$15a].

1.1 — A fifth level of ethics in mathematics

Consider the following definition of resistance from then political journalist Ulrike Meinhof [Mei68]:

> " *Protest ist, wenn ich sage, das und das paßt mir nicht. Widerstand ist, wenn ich dafür sorge, daß das, was mir nicht paßt, nicht länger geschieht. Protest ist, wenn ich sage, ich mache nicht mehr mit. Widerstand ist, wenn ich dafür sorge, daß alle andern auch nicht mehr mitmachen.*" [3]

Resistance is a matter of context, and a matter of personal choice – it is obviously ethical and moral for oppressed people [Fan64, Gut94] of the earth to rise up against their oppressors. In the German context, Meinhof largely discredited herself by becoming an active revolutionary as a first generation member of the Red Army Faction [Aus09] which used violence in attempts to achieve their political aims. We wish to differentiate her stated analysis from her concrete violent actions. We seek not to advocate for violence, nor to use violence, nor to eliminate the state. [4] Rather, we seek to ensure that the state lives up to its constitutional promises of protecting individual and societal security. Unlike Meinhof's acts of violence which amounted to terrorism, cryptography by contrast allows for resistance in a non-violent manner to the benefit of everyone except the ones who are spying [AAMMZ12] on us. Whistleblower Edward Snowden has observed this succinctly while paraphrasing [Fri14] former American President Thomas Jefferson:

---

[3] *"Protest is when I say such and such does not suit me. Resistance is when I make sure that what I do not like no longer happens. Protest is when I say, I will no longer participate. Resistance is when I make sure that all the others also no longer participate."*

[4] In cryptographic protocols and primitives, we should absolutely *eliminate the state* as observed by the authors of SPHINCS [BHH$^+$15].

> *While I pray that public awareness and debate will lead to reform, bear in mind that the policies of men change in time, and even the Constitution is subverted when the appetites of power demand it. In words from history: Let us speak no more of faith in man, but bind him down from mischief by the chains of cryptography.*

When discussing resistance to mass surveillance through applied cryptography, we reflect on the topic of ethics in mathematics (EiM). The EiM discourse defines **Four Levels of Ethical Engagement** [Mau18]:

> "*Level 0: Believing there is no ethics in mathematics.*
> *Level 1: Realizing there are ethical issues inherent in mathematics.*
> *Level 2: Doing something: speaking out to other mathematicians.*
> *Level 3: Taking a seat at the tables of power.*
> *Level 4: Calling out the bad mathematics of others.*"

What exactly is meant by *bad mathematics*? We define two different kinds of bad mathematics.

The first definition of bad mathematics encompasses incorrect, wrong, or otherwise overstated claims of security.

Criticism of the first kind of bad mathematics is well represented by the *Another Look* series of papers [KM19] by Koblitz and Menezes. Over the last two decades, the pair have produced a series of papers focused on this kind of bad mathematics. They observe that mathematics purporting to prove some kind of security sometimes becomes a way to mislead. Mathematics of this kind is regularly used to narrow an audience to include only those who self-select as having specialized mathematics training. Who will challenge a protocol or a cryptographic design if it rests on a proof of security? Those who feel confident in their understanding of the mathematics, of course. This kind of mathematics often acts as rhetorical sleight-of-hand in other, non-mathematics discourses.

How might proofs of security turn into something else entirely? In their *Another look at HMAC* [KM13] paper, they raise differences between the original claims of security for hash-based message authentication code (HMAC), and then the resulting final HMAC construction that was standardized by a national standards body. Remarkably, they find that the proof for the standard is weaker than expected, with the abstract mathematical proof for HMAC serving as a distraction from the actual standardized use case.

The second definition of bad mathematics involves math that is understood to be generally correct, but is used for nefarious political purposes or financial self-interest. Only a handful of mathematicians and computer scientists publicly refuse to take funding from intelligence or military sources, a much smaller set of people than those who accept such funding. Mathematicians and computer scientists tend to avoid criticism of this second kind of bad mathematics. There are exceptions: in computer science, Rogaway's refusal to license his cryptography for military use [5], and in mathematics, Koblitz's refusal to take money from intelligence agencies [6]. Exceptions aside, many others are "*on the take and loving it*" [Ass07]. It pays very well to pretend that there are no ethical or moral considerations in the world of mathematics and computer science.

It is in this context that we consider Rogaway's seminal work, the Moral Character of Cryptographic Work [Rog15] where he observes that cryptographic work is usually applied towards a *goal* and that there are *implicit politics*. It is a level-two critique: it raises important ethical issues to others in the field. An example of a level-three engagement is when Cedric Villani joined [Pai17] the French parliament. A level-four write-up is represented by the analysis and attacks presented in LOGJAM [ABD⁺15], which apparently caused quite a stir [HH15] at Fort Meade [7]. This thesis proposes and establishes a fifth level: *Level 5: Active resistance against bad mathematics.*

The goal of the cryptographic work in this thesis is not merely to protest but to build systems that may be deployed today, as an act of resistance, to thwart surveillance adversaries regardless of their claimed political legitimacy. Systems that may be legitimate under one authority are easily repurposed by another, illegitimate authority [Bla12], often invisibly. States have used and continue to use their surveillance capabilities to commit human rights abuses that would involve drastically different economic and political costs without mass surveillance. This work uses cryptography to protect individual liberty, while aspiring to a broader goal of achieving societal liberty. The implicit politics match the explicit: there shall be no compromises with the surveillance adversaries; we will make adversaries work for data, raising their economic costs.

When proposing designs to resist surveillance, we have found that Kerckhoffs' principle [Ker83, CGSN20] and its reformulation as Shannon's maxim [Sha49] are necessary, but not sufficient, protocol design considerations. The core protocols of the Internet were historically designed with non-surveillance adversaries in mind. The protocols were generally designed to survive various unintentional failure modes or to prevent uninvolved third parties from interfering unless they were in a position to perform surveillance. To the extent that many Internet protocols offer any kind of protection, it is usually of a trivial kind, such as ensuring that a non-surveillance adversary cannot interfere with communications through straightforward guessing. Two examples are: transaction IDs used in DNS queries, and TCP/IP sequence numbers used for connection establishment and teardown. A surveillance adversary may easily observe the DNS transaction ID and spoof a response based on its position in the network. Similarly, a surveillance adversary may tear down any TCP/IP connection that it has seen, as well as easily impersonate either system to the other. In both cases, a simple security analysis reveals that information useful for interference can be easily learned from observing the communication directly. Evident in both issues is a kind of naiveté in the early Internet's design: surveillance is not a problem, or if it is a problem, it is someone else's problem. We now know better: project BULLRUN, as mentioned in Section 4.3, makes clear that protocol design is subject to active manipulation with the explicit goal of enabling surveillance without disclosing this goal. How do they accomplish their goals with project BULLRUN? One way is that United States National Security Agency (NSA) participates in Internet Engineering

---

[5] See `https://web.cs.ucdavis.edu/~rogaway/ocb/license2.pdf`

[6] `https://www.washington.edu/news/2007/11/08/neal-koblitz-deciphering-the-cryptographer/`

[7] The National Security Agency headquarters is located at Fort Meade, Maryland, United States.

Task Force (IETF) community protocol standardization meetings with the explicit goal of sabotaging protocol security to enhance NSA surveillance capabilities [8].

Very few people outside the field of surveillance studies [Lyo07] understand the *practices of surveillance* unless they are engaged in ordering, performing, or evaluating surveillance. Even fewer people understand the technical matters involved. In news reporting about surveillance, technical details are regularly censored, the names of those carrying out mass spying are redacted, and the final surveillance "*products*" are almost never revealed to the world. Surveillance is used so regularly in an illegal manner that laundering legally inadmissible surveillance data for use in courts is euphemistically termed *Parallel Construction* [SC13, Mas14] in internal U.S. government agency documents. Similarly, LOVEINT [Pet13b, Gor13] is when surveillance capabilities are abused for personal sexual and/or romantic reasons. SEXINT [GGG13, Gra13] is similar to LOVEINT, except it is not considered abuse: the purpose of SEXINT is to find sexual preferences and proclivities for use as blackmail against a target.

The implicit limits and politics of informing the public through the news media are clear: the public should not actually understand the sources and methods in a technical sense [9]. This leaves society at a disadvantage when it comes to designing, building, and deploying countermeasure defenses. In this thesis, we reject this artificial social propriety, and in fact we reject censorship explicitly. The thesis includes facts and evidence that will upset people and companies, especially those in government circles. Serious efforts were made to conceal the data published here from the public, including over-classification, secret interpretations of laws and executive orders, and outright lies to the US Congress under oath [Wea16, ARAHS19]. Agents and assets of the US government have taken steps to mislead the public about the authenticity of leaked classified documents.

This thesis concerns itself first with analyzing realities of modern technical surveillance from state and other adversaries. We do this by examining public reporting and leaked classified documents to understand how insiders speak to each other [Jul06] and what they consider to be their own capabilities. When possible, we show the technical details of surveillance programs and how various programs interoperate; when only a limited amount of information is available to the general public, we attempt to reconstruct how surveillance programs might work.

In the course of studying the technical manifestations of surveillance, we find that merely speaking out in protest against data collection is largely ineffective. To resist surveillance, we must do more than simply criticize surveillance practices. We must design, create, and deploy hardware and software that thwarts surveillance using applied cryptography, and finally these solutions must become the defaults for otherwise unaware users. Only in this way can we avoid a computer literacy divide that excludes non-technical people from enjoying basic civil liberties.

We offer one path to resisting a variety of dystopian situations that mass surveillance brings us. The GDPR [Alb16] offers another: a combination of policy measures and data minimization as an explicit Privacy by Design [DFM01,Cav09,GTD15] systems goal. There are many other ways to contribute [Hug93, Jul06, Ell03, Gre12, CNE+14, Ell17, PK17,Wik21b] to the struggle against surveillance, but our plan of practical cryptographic resistance is the purpose of this thesis.

---

[8] Discussions with insiders confirmed what is claimed in as of yet unpublished classified documents from the Snowden archive and other sources.

[9] Worse, some journalists argue that because readers or viewers do not understand this technology, it need not be discussed. In most cases the journalists involved might as well be referring to themselves, not to any sort of scientific analysis of readers or viewers. Some of these journalists may even be part of a modern Operation Mockingbird [Sen96], while others are openly known as former law-enforcement officials or even as retired intelligence professionals.

The following chapters strive to bring forth deployable proposals that address specific surveillance programs and offer realistic methods of resistance.

1.2 — Thinking about the future

Data collection performed today is often not considered as a liability for tomorrow. Consider the following example of the top tier thinking by those who deploy and run our mass surveillance systems: During an invitation-only event [10] at a castle in the German countryside, a former director of the United States of America's National Security Agency (DIRNSA) held a speech. He advocated in favor of the previously secret policies and covert programs exposed by Edward Snowden [GMP13], an American whistleblower. The audience was skeptical because of Germany's history with surveillance and authoritarianism. A member of the audience asked, "What about those who come after you?" The response from the then retired director was succinct: "No one comes after us." This is a perplexing view on the liability inherent in mass surveillance to say the least. Even if one trusts the current authorities, is it really reasonable to also trust all future authorities in control of sensitive data? [11] Another audience member summarized this well with a controversial but accurate reference to German history: "Ah, a Thousand Year *Reich*, then?" Another attendee, a well known journalist, stood up and announced that he had a gift for the former DIRNSA. He held in his hands an unfurled poster [DEH35] of the famous Deutsche Hollerith Maschinen Gesellschaft (DEHOMAG) [Wik21f], an IBM subsidiary in Germany before and during the Second World War, which later became IBM Deutschland. DEHOMAG designed [LM94], built [ARBO04], sold [Pau03], deployed [Bla04], supplied materials to [Bla12], and serviced [Mun20] the first major mass surveillance systems deployed in Nazi Germany, including in forced labor areas, concentration camps, and extermination camps [Bla12]. The DEHOMAG system used punch cards which often featured Nazi iconography. These punch card systems were regularly used to organize property seizures and deportations as part of the Holocaust, thereby enabling [Bla12] genocide.

This poster, in the dour German propaganda style of the era, was an actual advertisement from 1935, featuring a large, single watchful eye looking down onto a factory with a punch card as background. The poster said "*Übersicht mit Hollerith Lochkarten*" [12]. The surveillance implications were a selling point.

The talk ended with a question-and-answer session that quickly devolved into loud, intense arguing [13]. It was quite clear that there was no consensus that a state could in-fact guarantee that there would be no one else to come after the current state. The problems resulting from long-term retention of data from mass and targeted surveillance can be likened to the difficulties of storing waste products [SBF15] from nuclear power generation.

---

[10] Private event, personally participated and witnessed in 2014.

[11] A separate issue is that the data collected by current and future authorities can also be obtained by fourth parties. The idea that this data is being kept secure is unserious, as evidenced by examples from the Snowden archive. The author of this thesis has read unredacted communication intercepts such as the one seen in Figure 1.1 while working on the Snowden archive in his capacity as a journalist.

[12] Literally: "Overview with Hollerith punch cards" though in the image context with a giant floating eye it can reasonably be understood as "knowing where everything is."

[13] After this meeting, which was supposedly covered under the Chatham House Rules, the former DIRNSA, who had addressed the thesis author out of the blue by name during his speech and was apparently very

Only a few years after this speech, exactly the same concern about data retention played out in a widely reported geopolitical event. Consider the situation of the American war [Wik10b] in Afghanistan. After twenty years of seemingly endless war [Ass11], the U.S. has finally, predictably, lost the war [14] in 2021, and with it, control of the deployed biometric surveillance systems. ID cards are a surveillance tool [Lyo09], and what was left behind goes far beyond simple ID cards. The US-led coalition reportedly abandoned a *full-spectrum identity intelligence system* that included detailed profiles of persons who worked for the American or other coalition forces in any capacity. There was reportedly data about their friends, families, favorite foods, and biometric identifiers. Their political leanings are probably obvious to anyone with access to their profiles. This and other surveillance programs are now in the hands of the Taliban [Hu21]. Who comes after us, indeed!

The United States holds [Bam12] a great deal of data in Massive Data Repository (MDR) [15] locations. The capacity of MDR sites such as the NSA's Bluffdale Utah [Bam12] is limited only by power, space, and cooling. In particular, the MDR datasets include pilfered encrypted Internet traffic such as the traffic passing through the XKeyscore [Gre13d, Gal14b, AGG+14b] surveillance system. There are various rules governing what is *selected* for long-term data retention in their *corporate repositories*. One example is that some traffic which is considered entropic by a standard Shannon Entropy estimate [16] is *selected* from the network in real time and saved to a database, preserving it for cryptanalysis using future technology. Countermeasures for this long-term strategy will involve not merely encryption, but encryption that resists quantum computers.

If the arrival of quantum computers is actually on the horizon as hypothesized by many [BL17], we should expect that the data stored in the MDRs and other locations will be attacked by whoever holds access to it. The NSA Cryptographic Exploitation Solutions (CES) is one group that has both access to the data and techniques for recovering plaintext. Many other groups like the CES exist with varying levels of competence. There are very few public documents revealing their capabilities, but in late 2014 the Foreign Intelligence Surveillance Act (FISA) intercept reports [AGG+14d, AGG+14c] from the PRISM

---

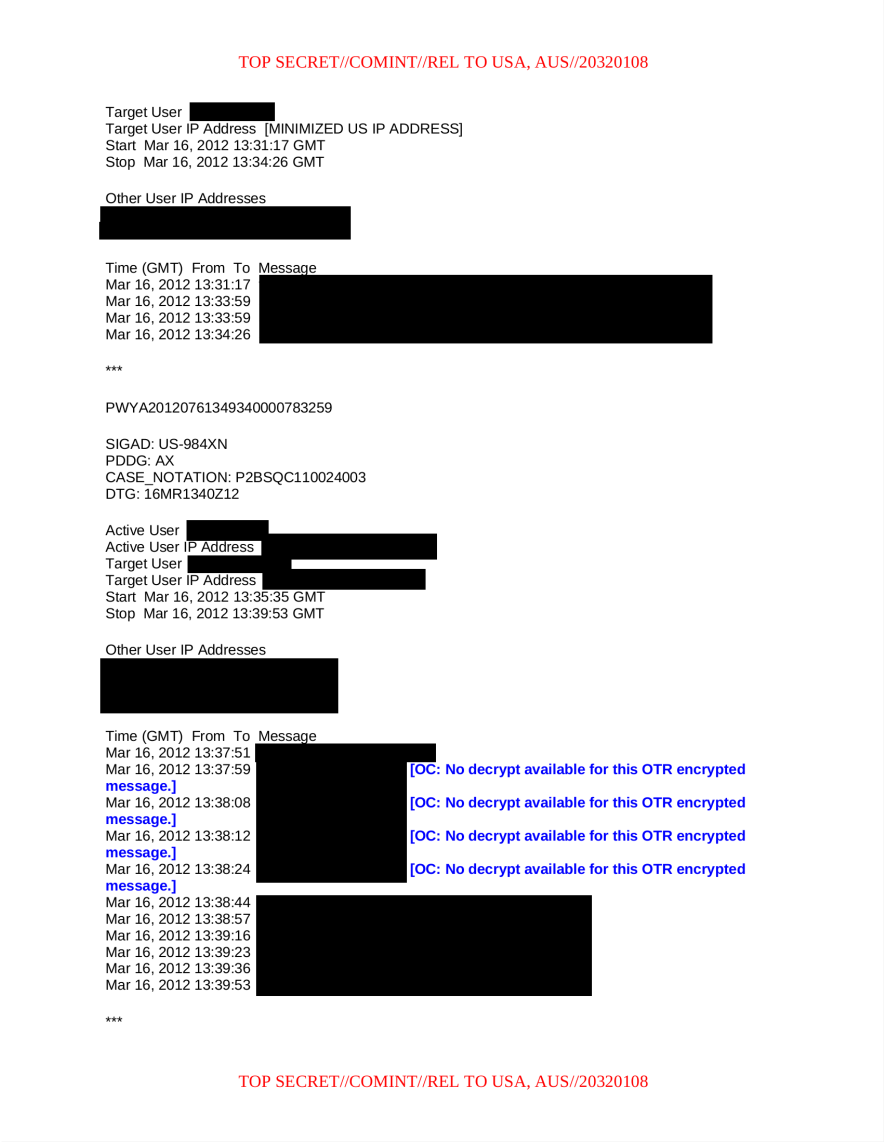
unhappy. He complained to the thesis author's then-employer and their government funders in an attempt to have the thesis author fired for daring to debate him. One supposes he was also unhappy that he had lost the crowd with his childish arrogance and trivial analysis that was easily and clearly disputed. Later the thesis author was informally censured by executives at his former employer and admonished to never "do that again or else." This discussion made clear that *all Tor traffic* is fair game and is collected under EO12333 [Rea81, Jay21] because "terrorists and other extremists use it." The worthlessness of privacy by policy to constrain surveillance adversaries is clear in violations of policy, big and small.

[14] If only the United States of America had not started the Afghan or Iraq wars! If only America could have lost the war twenty years earlier! Between 171,000 and 360,000 [Wik21e] humans were killed in the American war in Afghanistan and between 151,000 and 1,033,000 [Wik21d] in the *most recent* American war in Iraq.

[15] Later after exposure of this term, it was reformulated as Mission Data Repositories [Hog15] as the term "massive" made it difficult for the NSA to deny they were engaged in mass surveillance. We use the original term to emphasize the size and scale.

[16] This statement is based in part on an analysis of as of yet unpublished XKeyscore source code that performs a Shannon Entropy estimate. Some kinds of Internet traffic that is considered entropic is recorded for later analysis.

(*SigAd* US-984XN) interception program, discussed in Section 4.3.1, were published by Der Spiegel, as shown in Figure 1.1. The intercepts show that both the metadata and content of a target's communications are surveillance goals. Der Spiegel made the editorial decision to redact content not protected by encryption.



TOP SECRET//COMINT//REL TO USA, AUS//20320108

```
Target User [redacted]
Target User IP Address  [MINIMIZED US IP ADDRESS]
Start  Mar 16, 2012 13:31:17 GMT
Stop  Mar 16, 2012 13:34:26 GMT

Other User IP Addresses
[redacted]

Time (GMT)  From  To  Message
Mar 16, 2012 13:31:17 [redacted]
Mar 16, 2012 13:33:59
Mar 16, 2012 13:33:59
Mar 16, 2012 13:34:26

***

PWYA201207613493400007832259

SIGAD: US-984XN
PDDG: AX
CASE_NOTATION: P2BSQC110024003
DTG: 16MR1340Z12

Active User [redacted]
Active User IP Address [redacted]
Target User [redacted]
Target User IP Address [redacted]
Start  Mar 16, 2012 13:35:35 GMT
Stop  Mar 16, 2012 13:39:53 GMT

Other User IP Addresses
[redacted]

Time (GMT)  From  To  Message
Mar 16, 2012 13:37:51 [redacted]
Mar 16, 2012 13:37:59  [OC: No decrypt available for this OTR encrypted
message.]
Mar 16, 2012 13:38:08  [OC: No decrypt available for this OTR encrypted
message.]
Mar 16, 2012 13:38:12  [OC: No decrypt available for this OTR encrypted
message.]
Mar 16, 2012 13:38:24  [OC: No decrypt available for this OTR encrypted
message.]
Mar 16, 2012 13:38:44 [redacted]
Mar 16, 2012 13:38:57
Mar 16, 2012 13:39:16
Mar 16, 2012 13:39:23
Mar 16, 2012 13:39:36
Mar 16, 2012 13:39:53

***
```

TOP SECRET//COMINT//REL TO USA, AUS//20320108

Figure 1.1: U.S. domestic FISA intercept of OTR messages without decryption
Courtesy of Der Spiegel [AGG+14d, AGG+14c].

Notably, the CES was unable to recover the plaintext of the encrypted messages in this intercept. The targeted person was using, though not consistently, the Off-the-Record

[BGB04] (OTR) end-to-end encryption protocol. This protocol provides a number of security properties such as confidentiality, authenticity, and deniability. It works with various Internet chat protocols as long as each party has OTR-aware software. The leaked FISA intercept conclusively documents that NSA's CES group, as late as 2014, was unable to break the cryptographic design of OTR to recover the messages of the targeted user. In this particular OTR implementation, a 1536-bit prime is used for modular exponentiation to agree on session keys, and those keys are used to encrypt messages with AES. These intercepts and the related reporting around them confirm again what many cryptographers have long believed, that mathematics contains fundamentally hard problems that are not solvable merely because an adversary has huge financial and computing resources. Though we expect the NSA CES group and other similar groups to attack any and all encrypted traffic of interest to them, we know that strong, properly implemented cryptography is the thing that certainly stops them from recovering plaintexts. In this case, we know that the design of the OTR protocol counted as strong in 2014.

Mass surveillance data sets, encrypted or not, allow for retroactive searches of the pattern of life [Fra17] for targeted individuals or groups. These people may be found by simple keyword association or other automated search systems. It is important that we plan for this moment in history, as it will be possible to attack the ciphertext of any intercepted dissident who has successfully used encryption without post-quantum guarantees. Just as Nelson Mandela was caught [Joh90] with help from the American CIA, future Mandelas will be caught with the help of the CIA, NSA, or others if care is not taken to prevent this obvious outcome. The United States of America's Federal Bureau of Investigation (FBI), whose headquarters are still named after J. Edgar Hoover whose unconstitutional [Ell14, Med14] activities are notorious, is one of the so-called "*corporate customers*" of mass surveillance data from the NSA and the CIA. Their own surveillance efforts are also substantial. For the American exceptionalism [Mad98] advocates who do not worry about foreign nationals or their rights and liberties, it is instructive to consider the history of COINTELPRO [Dav92, Sai02, Med14, DW01].

COINTELPRO was a series of secret programs run by the FBI [17] to disrupt politically protected activities in and outside of the United States. The FBI is a lawless institution [Ger19]. As part of that history, the FBI has used or attempted to use compromising material, or kompromat [HL21] [18] to destroy so-called *internal enemies* [Bro04]. The FBI targeted civil rights leader Martin Luther King Jr. [Com75, Mar18] by encouraging him to commit suicide after alleging his involvement in a sexual affair and later likely being involved in his assassination [KCBSN21]; it subjected author Ernest Hemingway to such heavy surveillance as to contribute to his suicide [Mar06, The93, Mod08]; it involved itself directly with a Chicago police raid where Black Panther leader Fred Hampton was assassinated [Haa11, CVW02] after being drugged by a police informant; and it participated with the New York Police Department and the CIA in the assassination of civil rights activist Malcolm X [DP03, Mia21]. FBI targeting of the leadership of the Black Panther domestic political movement was particularly cruel [Jam09]. Almost no one has been held accountable in any meaningful sense for the FBI's contribution to the deaths of those people; this is almost certainly because of institutional racism against people of color as well as a matter of economic class. Women such as political activists Angela Davis and Assata Shakur [19] are especially rare in that they are still alive.

---

[17] A former FBI agent introduced me to Howard Zinn's nickname for the FBI: the Federal Bureau of Intimidation

[18] Long practiced by American intelligence; originally from Russian компромат

A black person will rarely find justice in American courts, especially if they are a woman, and definitely if they are economically disadvantaged or politically active. These abuses from the FBI continue and grow easier with the passing of time thanks to technological advances in surveillance and data gathering of all kinds. FBI agents [Rei21] who dare to tell the public about modern COINTELPRO-style operations are gagged, arrested, jailed, stripped of their pensions, and their court proceedings are essentially kept secret. Many other Americans have suffered similar injustices and simply were not prominent enough to be the subject of serious individual research projects. The FBI has not [Baz12, Uni13a, Rei21] fundamentally changed their practices, nor has the United States Department of Justice [Ber02, SB11, Pil11, Ree12, Can13, Rei21] under which it operates.

The often brutal suppression [Dav92, DW01, Gra16, Baz12] of American democratic dissent demonstrates that traditional methods [Tim02, Sha10] of resistance will fall apart under a regime of mass surveillance. Responding to this is politically complicated and requires extreme technical sophistication. There is an urgent need for a modern version of the famous 1926 Victor Serge book [Ser05] that was created by studying the archives of the Okhrana, the former Tsarist Russian secret police. This thesis is not that book. There is additionally an urgent need for a comprehensive project to catalog known surveillance programs and activities as revealed by original internal documents. Such a project should analyze related uses of surveillance data, and should be mapped to political geography and indexed by time for easy study. And we need data about secret cryptography [20] in order to understand the cryptanalytic protection techniques used by those who also break cryptography.

Cryptography presents a barrier to the NSA and similar agencies if a message is simply viewed in isolation, and that tells us that in mathematics there is hope for the world's privacy and security. To put this another way, it is easier to break into a remote system and steal the relevant cryptographic keys than to break strong cryptographic protocols. In the FISA intercept example, NSA largely functions as a *passive* adversary. However, we see later in this thesis that NSA and similar adversaries are also *active* adversaries. Such adversaries may try to bypass encryption protection by breaking into running computer systems. There is hope for security of communications and even for security of devices generally, but the details are usually a matter of operational security, a matter largely outside of cryptography but also seemingly impossible without cryptography.

As remarked in the Cypherpunks book [AAMMZ12]: "*Cryptography is the ultimate form of non-violent direct action... Strong cryptography can resist an unlimited application of violence. No amount of coercive force will ever solve a math problem.*" We hope that cryptographers, computer scientists, hardware engineers, mathematicians, and surveillance studies academics will continue to accept the responsibility their education brings them and work together to help all people actively resist targeted and mass surveillance.

---

[19] Assata Shakur lives in exile in Cuba and continues to be sought by bounty hunters from the United States to this day. She remarks that this is not dissimilar to being treated as a fugitive slave.

[20] An example is *Suite-A cryptography* or *Type-1 cryptography*, so designated by the NSA. The NSA now calls this the Commercial National Security Algorithm Suite (CNSA). An example is the secret block cipher BATON [Wik18].

1.3 — Organization of this thesis

This thesis is organized into three parts. The first part, Chapters 1 – 3, describe the background information required to understand the original research in the second and third parts.

The second part, Chapter 4, outlines the Adversary's intents, capabilities, funding, legal obligations, violations of those obligations, and more to help readers define a relevant threat model. We present and reference various kinds of evidence, including but not limited to internal classified U.S. government agency documents, which have been leaked to the journalists so that they may publish facts which are in the public interest.

The third and final part, Chapters 5 – 8, consists of constructive protocol designs and software implementations attempting to counter some of the tactics and strategies from Chapter 4.

Chapter 2 outlines the background on network protocols common to all research.

Chapter 3 outlines the background on cryptography common to all research.

Chapter 4 reviews historical, political, economic, and technical adversarial capabilities. This chapter includes previously published leaked documents that are from works that the author has written about in his role as a journalist with Der Spiegel, NDR, Le Monde, WikiLeaks, and others.
We discuss passive and active surveillance by a variety of Adversaries, including but not limited to active hacking, deploying custom hardware implants, and automated infection with malware. We furthermore discuss source code from an important NSA mass surveillance program.

Chapter 5 describes the DNS ecosystem. This work was published at the NDSS 2017 DNS Privacy Workshop DPRIV17 as joint paper [GWEA18b] entitled *Towards Secure Name Resolution on the Internet* with Christian Grothoff, Matthias Wachs, and Monika Ermert. The Domain Name System (DNS) provides crucial name resolution functions for most Internet services. As a result, DNS traffic provides an important attack vector for spy agencies, as demonstrated by the QUANTUMDNS and MORECOWBELL programs of the NSA. This chapter reviews how DNS works, and explains alternative methods designed to improve the security and privacy of domain name lookups for the future Internet.

Chapter 6 examines a Tiny WireGuard Tweak. This work was published at Africacrypt 2019 as joint paper [AMW19] entitled *Tiny WireGuard Tweak* with Chloe Martindale and Peter Wu. We show that a future adversary with access to a quantum computer, historic network traffic protected by WireGuard, and knowledge of a WireGuard user's long-term static public key can likely decrypt many of the WireGuard user's historic messages. We propose a simple, efficient alteration to the WireGuard protocol that mitigates this vulnerability, with negligible additional computational and memory costs. Our changes add zero additional bytes of data to the wire format of the WireGuard protocol. Our alteration provides transitional post-quantum security for any WireGuard user who does not publish their long-term static public key – it should be exchanged out-of-band. Users who wish to adopt this tweak must deploy the new protocol and then generate fresh keys.

Chapter 7 introduces the Vula protocol. This work, entitled *Vula: automatic local area network encryption*, was previously unpublished. It is joint work with Leif Ryge. We introduce Vula, a protocol and suite of Free Software tools for automatically protecting network traffic between hosts in the same Local Area Network (LAN). Without any configuration, or any user awareness, Vula automatically blinds passive adversaries. With user awareness and a small amount of interaction, it also protects connections using *.local* hostnames, or any other user supplied domain, against active adversaries. The protocol additionally provides protection against a passive adversary who is recording traffic today and who may have a quantum computer tomorrow. Vula's protections persist with network topology changes which occur naturally over time, allowing users to maintain cryptographic assurances while roaming between different LANs. Our GNU/Linux Free Software implementation operates without requiring centralized administration, specialized network equipment, or significant performance penalties.

Chapter 8 proposes a new way to rendezvous with REUNION. This work, entitled *REUNION*, was previously unpublished. It is joint work with Johan Kjær, David Robinson, Leif Ryge, Kit Smeets, and David Stainton. We introduce REUNION, a privacy-preserving rendezvous protocol and suite of Free Software tools for privacy preserving rendezvous. Communication requires context. In digital communication, this context usually includes long-term persistent identifiers, the metadata of which is often of interest to third parties. In light of this, we consider the problem of how to establish secure networked communication as a follow-up to a physical, offline meeting without computers. REUNION participants wishing to rendezvous online need only share a passphrase of their choice. REUNION provides forward-secret message confidentiality against active adversaries who participate in the protocol with or without a quantum computer. It additionally provides forward-secret metadata confidentiality against passive adversaries who obtain complete protocol message transcripts, with or without a quantum computer. Forward-secret metadata confidentiality against active adversaries is also provided, but with the caveat that it can be lost if the participating adversary obtains a quantum computer in the future. We consider several deployment scenarios and release a Free Software implementation. We additionally introduce a protocol model for REUNION using the Verifpal [Kob19] (versions verifpal-v0.23.0 and verifpal-v0.26.0) verification project. The formal verification of REUNION was done with several variants of the protocol model that took a range of time to complete. The fastest verification took approximately 0.5 core hours on a modern Intel CPU (Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz), and variants took significantly longer. The longest single property (confidentiality) verification with verifpal-v0.26.1 took roughly $158,469$ core hours (dual-socket AMD EPYC 7451 24-Core Processor), with verifpal-v0.23.0 $24,632$ core hours (dual-socket AMD EPYC 7742), and verifpal-v0.26.0 $136,881$ core hours (dual-socket AMD EPYC 7742). The queries for properties of the system as described in the model were proven with Verifpal.

CHAPTER 2

Background on network protocols

> *"The smart way to keep people passive and obedient is to strictly limit the spectrum of acceptable opinion, but allow very lively debate within that spectrum – even encourage the more critical and dissident views. That gives people the sense that there's free thinking going on, while all the time the presuppositions of the system are being reinforced by the limits put on the range of the debate."*
>
> — Noam Chomsky, *in his book* ***The Common Good*** *[CBN98]*

This thesis is primarily focused on the Internet, surveillance of the Internet and other networks, and using the Internet and other networks to securely communicate. In this chapter, we introduce each of the technologies assumed to be understood for work introduced in subsequent chapters. We first introduce a commonly understood model of the networks such as the Internet in Section 2.2. We introduce common physical infrastructure and the core Internet Protocol in Section 2.3. Section 2.4 discusses hostnames such as `example.com`, how they work, and how they're used on the Internet. Section 2.5 is a related topic of how hostnames such as `host.local` work on local, perhaps offline, networks. For a review of adversarial technical capabilities and political goals we refer the reader to the next chapter 4.

2.1 — Free Software, Open Hardware, Operational Security

All of the cryptography in this thesis is relevant only if users have a meaningful ability to physically guard, and consensually perform non-adversarial forensics on their own systems. While reverse engineering binaries is possible, to meaningfully be able to inspect software, we consider it a prerequisite that source code is not only available, but it should also be Free Software [Sta02]. The Four Freedoms as defined [FSF90] by the Free Software Foundation are an important part of a practical operational security posture:

- *"The freedom to run the program as you wish, for any purpose (freedom 0)."*
- *"The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this."*
- *"The freedom to redistribute copies so you can help others (freedom 2)."*
- *"The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this."* [FSF90]

Free Software alone is not sufficient. Professional cryptographic and security review is almost certainly also required for both software and hardware. Similarly, we consider that hardware requires the same level of transparency, which is nearly impossible even when it is Free or Open hardware [Söd11] as the problem of verification of hardware, like software, is non-trivial. The adversaries involved are known to use custom hardware implants [1] as seen in Figure 4.18 to otherwise secure systems which adds a concern about not only designing and building secure systems, the systems must also be physically guarded so as to resist tampering or modification.

---

[1] The journalist targeted with this [Gre20, MM18] specific hardware backdoor, or implant, worked with the author of this thesis on a number of journalistic publications during which they were both monitored with this implant as well as experiencing other disruptive activities [DNI21].

2.2 — Layers of the Internet

The Open Systems Interconnect (OSI) [Sta87] model of protocols for communication is an abstraction to help understand and discuss different protocols and their interdependence. Each layer provides standard functionality to build upon as a protocol implementer and each layer is built on the layers below. The seven layers of the model are as follows:

7. Application layer
6. Presentation layer
5. Session layer
4. Transport layer
3. Network layer
2. Data link layer
1. Physical layer

Section 2.3 discusses how the bottom three layers are combined together: on layer one, the Physical layer, and layer two, the Data link layer, technology is used to provide connectivity to layer three, the network layer protocols. Chapter 7.2 discusses layer two protocols such as Ethernet Address Resolution Protocol (ARP) [Plu82] from a security perspective. Later chapters examine protocols at layer four, the Transport layer, such as Transport Layer Security (TLS) [DA99, Res00], and the Domain Name System (DNS) is presented in Section 2.4 and it sits between layer four and five. Layer five is seen in long-lasting HyperText Transfer Protocol (HTTP)) [BLFF96] communication between a web browser and a web server as mentioned later in this chapter. Layer six, is concerned with matters such as character encoding and serialization, also found in HTTP. Layer seven, the application layer, is used for representing things such as an end user's web browser or email client.

These layers are not strict categories and should be considered as loose abstractions. There is an old joke about the existence of eighth, ninth, and tenth layers of the OSI model; we discuss issues at those layers and below in Chapter 4.

2.3 — Ethernet networks and the Internet Protocols

Understanding *the practices of surveillance* used to collect Internet Protocol (IP) [Pos80a, DH98] datagrams [Hor84, Cra96] transmitted over wired and wireless networks such as Ethernet [MB76, Spu00] networks is required to begin to mount suitable defenses to this very real and existing problem space.

Ethernet networks are common short-distance networking technology that is widely deployed in home, business, and government use. Ethernet communication is performed by turning communications data into pieces of data called Ethernet frames for communication between hosts on the same Ethernet segment. These Ethernet frames typically include information for other protocols such as IP [Pos80a, DH98] or TCP/IP. Ethernet networks have 48 bit addressing, and IP datagrams [Hor84, Cra96] use either 32 bit (IPv4) or 128 bit (IPv6) addresses. To translate between IP addresses and Ethernet addresses, the

Address Resolution Protocol (ARP) [Plu82] is used in Ethernet network segments. The use of IP datagrams allows for more complicated end-to-end protocols to be constructed such as the Transmission Control Protocol (TCP) [Pos81c, ZEE16, Ste94] for reliable stream oriented communication, the User Datagram Protocol (UDP) [Pos80b, SHD91] for unreliable datagram oriented communication, and the Internet Control Message Protocol (ICMP) [Pos81a, Pos81b] used for in-band signaling. The combination of these protocols with IP datagrams is often called TCP/IP, and the datagrams containing TCP, UDP, or ICMP are commonly referred to as IP packets. The use of IP packets that include TCP, UDP, and ICMP allow parties to exchange IP addresses with some application layer protocol, e.g., with the Domain Name System (DNS) as explained in Section 2.4, and then initiate communication with TCP/IP.

With regard to cryptographic notions of security, TCP/IP and the related ARP traffic in Ethernet networks is essentially unprotected from the perspective of a surveillance adversary. With the exception of SYN cookies [Ber97] which were introduced as a security mechanism to ensure availability, there are essentially no security goals achieved by default against an active adversary by the ARP, IP, TCP, UDP, and ICMP protocols, save availability. At best, we see random counters where some implementations were not so random [Zal01]. Later attempts to retrofit TCP/IP with security enhancements have largely failed to be meaningfully deployed as a new default. Higher level protocols such as HTTPS have had much more success with intentionally improving the security considerations.

2.4 — The Domain Name System

The need to remember an IP address for an Internet service is usually delegated to the Domain Name System (DNS) [Moc89, EMUM90]. The DNS allows a machine or any other person to remember a hierarchically structured name (e.g. `example.com`) rather than remembering a 32 or 128 bit address. It also allows for other types of data [Ros93, DVGD96] to be retrieved from a name alone. As an example, Laura remembers the name `example.com` rather than the IP address `192.0.2.1`. She uses DNS as implemented by her operating system to *resolve* the name to the current IP address. The first attempts at adding security to DNS was largely an afterthought [rK97] that did not focus on adversarial surveillance but rather on tampering with answers by on-path or off-path adversaries as we describe in Section 5.6. The privacy of user queries to the DNS is almost never considered as we describe extensively in Section 5.4; we compare and contrast possible alternative solutions to the DNS in Section 5.1. We consider it an interesting development that web browsers have attempted to solve the failures of the DNS by resolving DNS names through large centralized caches that offer a secure communication channel to the user. The upstream queries from those caches are still entirely vulnerable to surveillance and manipulation, a stark contrast with other systems that provide end-to-end security such as DNSCurve [Ber08b] discussed in Secton 5.9. Similarly, some authoritative DNS root servers, notably the F-root servers, participate in the Passive DNS project [Int14] for collecting surveillance data about DNS queries and responses.

2.5 — Multicast Domain Name System (mDNS)

DNS is a delegated and distributed [Pos94] database, and to resolve public names, one requires access to the public databases located in various parts of the Internet, usually through connectivity to the Internet. Link-local Multicast Name Resolution (LLMNR) [ATE07] was an early proposal to allow for resolving names using only the local network segment. Later, Multicast DNS (mDNS) [CK13b] was devised to resolve names which are local to the current network segment using IP multicast [DC85,QA01,AAMS01, HT02]. The use of mDNS is usually combined with DNS-Based Service Discovery (DNS-SD) [CK13a, LCBM15] to allow for entirely local name resolution and service discovery, often under a specially reserved name [CK13c] such as `.local.` [CK13b] or another site specific configuration such as `home.arpa.` [PL18]. Resolved local names are not guaranteed to be globally unique or secure. The mDNS queries for a host are similar to normal DNS requests in structure but they are sent to a special IP address called a multicast address; there are different multicast addresses for IPv4 and IPv6 as we describe further in Section 7.4.2.

2.6 — Hypertext Transport Protocol (HTTP)

The Internet is said to be more than the World Wide Web (WWW) and yet, the World Wide Web is an important part of the Internet. The web is built on top of the Hypertext Transport Protocol (HTTP). HTTP is a protocol [BLFF96, FGM$^+$97, Mas98, BPT15] for transferring data between an HTTP client and an HTTP server. Similar to the original DNS protocol, security was not a concern during the design of the HTTP protocol. Early attempts were made to introduce some kind of encrypted transport and Public Key Infrastructure (PKI). The first encrypted transport Secure Socket Layer [EH95] (SSL) was broken nearly immediately, and replaced with a second version which is now effectively banned [TP11] for good reason by the protocol police [tOCS21]. The web needed cryptography for various reasons, primarily economic reasons, and so the Transport Layer Security (TLS) protocol was created [DA99] as an iteration on the previously broken SSL, and explicitly extended to the web [Res00].

2.7 — Transport Layer Security (TLS)

The Transport Layer Security (TLS) protocol is one of the most used encryption protocols on the Internet. It is used to transport and secure many [Hof99, New99, FH05] other protocols such as HTTP as discussed in Section 2.6. Over time, security issues have been found and the protocol has been adapted [DR06], growing organically [BG07, DR08] , like slime mold [Nak01] until version 1.3 [Res18], the current version of the TLS protocol. There are TLS specifications for running TLS over TCP and for TLS over UDP with Datagram Transport Layer Security (DTLS) [RM12].

While TLS is often paired with HTTP, it is also the basis for cryptographic protection for other protocols that are otherwise at different layers such as protecting IP packets. One noteworthy cryptographic protocol designed to work with TLS and DTLS is the Virtual Private Network (VPN) known as OpenVPN [Yon].

2.8 — Virtual Private Networks (VPN)

A Virtual Private Network (VPN) protocol allows a computer to establish a secure connection or connectionless tunnel between itself and a VPN endpoint such as a server. Usually a VPN protocol provides the usual cryptographic assurances with a variety of different authentication options. There are an abundance of Virtual Private Network

(VPN) protocols available with a variety of security properties. Some VPN protocols such as Point to Point Tunneling Protocol (PPTP) [HPV+99] are outdated and used for legacy reasons. Many VPN protocols are simply not serious cryptographic proposals and they are not interesting cryptographically. While there are probably hundreds, if not more, of VPN implementations, we focus on three primary protocols: OpenVPN [Yon], IPsec [Dun01], and WireGuard [Don17a].

2.8.1 – Sabotage. Of the three, OpenVPN is a protocol without a basis in formal specifications or peer review except where TLS or DTLS is concerned. IPsec is a protocol built by committee as part of the Internet Engineering Task Force (IETF) Request for Comments (RFC) process. Both are understood to have been weakened [AGG+14d, BBG13, PLS13, Lar13] [2] by the NSA intentionally. The techniques are not entirely understood but it appears that the NSA uses every option available when they deem it necessary. What they deem necessary is not always what is expected. This includes sending people to standardization meetings to sabotage the security standards as well as sending people into companies to perform so-called *cryptographic enabling*; this is how NSA euphemistically describes sabotaging cryptography or security. It is difficult to overstate the level of subterfuge understood to be attributed to the NSA, both by external investigations and by their own internal documents. The normally classified and thankfully leaked *black budget* [GM13a] shows hundreds of millions of dollars budgeted and specific successes against specific U.S. domestic and international companies. In a related document the NSA describes a normal situation where the NSA intercepts VPN traffic to decrypt the contents, modify the traffic if desired, and then *re-inject and re-encrypt the traffic* to send on to the original destination. The NSA estimated in 2011 that they performed around one thousand attacks against VPN sessions per *hour* and NSA projected it would soon be performing one hundred thousand such attacks in parallel per hour. It is reasonable to assume that this number is significantly higher after more than a decade.

In some cases, we see specific encryption products or devices which are said to have been *enabled*, which is to say, made insecure with regard to NSA surveillance capabilities. Many of the company names and device names, despite being newsworthy information, have been censored or otherwise redacted from published documents by journalists or their editors under immense international pressure from the NSA and the US Government at large. In Germany where journalists were relatively free to publish on these topics, it is understood [Spi15] that the US engaged in espionage against journalists for their reporting on these topics, including the author of this thesis.

Later, a new VPN, WireGuard [Don17b] was published in a peer-reviewed venue and quickly displaced some uses of IPsec and OpenVPN. WireGuard is based on the Noise IK pattern of the Noise Framework [Per18], and is essentially a very simple, very elegant tunnel design. The cryptographic implementation is minimally modified from the abstract cryptographic protocol to apply to IP networks. It has only one serious cryptographic shortcoming that is known at this time: like the Noise IK pattern it is based upon, it does not have forward-secret identity hiding. If it is possible to compromise one party, and if network traffic was recorded, an attacker can then confirm guesses with whom the compromised party was communicating, though the data exchanged in the past itself remains confidential. It is an unfortunate fact of the protocol that forward-secret identity hiding is an issue as this is one piece of information that nearly all adversaries want out of any protocol: a way to identify people.

---

[2] Each publication has different summaries and different documents as a result of different editorial policies and goals. These documents are some of the most important Snowden documents and the German publishers were much bolder than their US and UK counterparts. The Guardian office in the US was placed under heavy technical surveillance, which was intimidating to journalists working on these stories. Guardian-UK was also put under a secret official censorship order known as a D-Notice by the U.K. government. Later Guardian-UK was politely raided by Government Communications Headquarters (GCHQ) who then ordered Guardian-UK to destroy their copy of the Snowden archive while GCHQ operatives watched to confirm the destruction. Guardian-UK complied [Bor13a] and did not notify key journalists working on the Snowden archive before the destruction took place. Those publishing in Germany were free of such censorship orders and deeply dismayed by these events. The New York Times may have other structural issues as expressed by Bill Keller in Mediastan [Wah13, And21a].

CHAPTER 3

Background on cryptography

> *"We do not have national security concerns. We have concerns about human beings."*
>
> — Julian Assange *in a 2010 interview.*

This thesis uses cryptography to protect privacy. We discuss and design cryptographic protocols with the aim of thwarting surveillance of data and/or metadata. In approaching this difficult task, we evaluate in this chapter what cryptographic primitives are best suited to our needs. The goal is to make recovery of encrypted data by a surveillance adversary impossible.

In Chapter 5, 6, 7, and 8, we present network protocols that are protected by cryptography. We evaluate the security of those protocols in the standard symbolic model as implemented in verification tools such as Verifpal [Kob19]. We present manual analysis in Chapter 5, 6, 7, and 8, and we additionally verify Verifpal models of the protocols presented in Chapter 7 and Chapter 8.

Verifpal is a formal verification tool that works in the symbolic model. It presents a simple and intuitive language for writing models which is not common with formal verification tools. These models allow for straightforward human language to be used to describe participants in a protocol, their properties, and their actions as seen in Listing 8.1 or in Listing 8.3. After writing down the parties involved, and what it is that they're doing as a model, we are able to ask the verifier questions about this model. The verifier is considered to be sound but not complete. Soundness means that if the verifier returns an answer, it is correct. However a lack of completeness means that the verifier may not return an answer at all. All verifiers written as software are likely to have bugs or issues that lead to either of those properties being invalidated, so we consider such verification to be important but not perfectly certain without the possibility for errors.

We also consider properties of primitives such as key lengths and resistance to attacks by adversaries with quantum computers before allowing their use in any protocol verified in the symbolic model. Beyond the scope of this thesis, we would welcome the development of easy to use verification tools in more powerful computational models.

This chapter gives a preview of the cryptographic primitives used in upcoming chapters. We introduce each of the cryptographic techniques required for work introduced in subsequent chapters.

3.1 — Mathematics as informational self-defense

We consider people such as Laura and Julian, who want to exchange a message and we construct a number of scenarios to explain different important cryptographic concepts as functional primitives. We will use these primitives to construct some notion of privacy requirements such as confidentiality, integrity, and authenticity. Additionally, notions of identity may be built from these highly desirable properties.

3.2 — Notation

A note on notation for the following formal definitions:
The use of $x||y$ signifies that x is concatenated together with y. We use ℓ as variable

whenever we need to denote the length of an element of fixed length. If we need to deal with two elements of fixed but possibly different lengths we use ℓ and n to denote their lengths. $\{0,1\}$ is the set of bits. $\{0,1\}^*$ is the set of bit strings of an arbitrary length such as messages. $\{0,1\}^n$ is the set of bit strings of length n (e.g.: 128 bits, 196 bits, or 256 bits for a key). $m \mapsto d$ shows that input m is transformed into output d. DH, Sign, and Vf, defined later, use $\mathcal{K}$ to signify the space of secret keys, $\mathcal{P}$ to denote the space of public keys, and $\mathcal{S}$ to denote the space of signatures. We also group values into an n-tuple such as (x, y) or (x, y, z). $\oplus$ is used to signify exclusive or (XOR).

3.3 — Hashing

Reduction of arbitrarily long messages to a fixed bit width is necessary for a number of cryptographic proposals. To accomplish this, we introduce one of the most important functions in cryptography: the *hash* function. We define the hash function as $d = H(m)$ and formally $H : \{0,1\}^* \to \{0,1\}^\ell, m \mapsto d$. A hash function takes an arbitrary length bit string and compresses it down, usually but not always into a shorter fixed-length value, such as a 256 bit value. Hashing is useful to create a *digest* of any message such as any plaintext, ciphertext, or additional associated data. Hashing is generally designed to be extremely fast as well as to have some special properties such as preimage resistance, second preimage resistance, and collision resistance. Preimage resistance is a property that is achieved when there is no efficient method to invert the hash function to find an input for a given output. Second preimage resistance is a property whereby no second input can be efficiently found to match the output value of the first input. Any change to a hash function's input will create a change to the output in such a way that even finding two inputs that map to the same output is computationally infeasible. This property is called collision resistance.

For secure hash functions, finding collisions is easier than finding second preimages as an attacker may search for any collisions rather than search for inputs for only one matching output. Parameters have to be chosen such that both are computationally infeasible. In this thesis we will use the BLAKE2 [ANWW13] family of hash functions as well as others.

Password hashing is a variant of hashing that attempts to make confirming that an input matches an output more difficult. Consider an adversary that has a list of password hashes and wishes to use those hashes to their advantage. To use those password hashes they must find the original password input that maps to a password hash. Weak passwords, that is, known and in a list, will still be found by an adversary. Checking each item on the list takes a much longer time with cryptographic password hashing functions than with normal cryptographic hash functions. Normal hash functions may be able to process billions of items in a list per *second*, while a password hashing function may be tuned through the use of parameters to allow a single password hash output and check per second. Password hashing functions are generally designed to create a fixed bit width digest of data with the caveat that they are designed to be slow by being bound by some computational problem. Two common ways are to use large amounts of contiguous memory, as opposed to non-password hashing, or through the use of calculations that utilize the central processing unit or graphics processing unit in a way that creates an inefficiency purposefully where normal hashing functions are extremely efficient. Password hashing as seen in Section 8.3.1 is provided by argon2 [BDK16] which builds on BLAKE2.

3.4 — Symmetric Encryption: block cipher

Confidentiality of a message is a property where no one may read the contents of an encrypted message except the sender and the intended recipient. To encrypt a message, we need to agree on some way to encrypt messages, and the communicating parties must agree on a *symmetric key*. A symmetric key is a secret value shared by both the sender and receiver. One way to use a symmetric key is to use it with a *symmetric block cipher* which is a function that takes a fixed size block of data, and a symmetric key as inputs: $c = \mathsf{Enc}(k, m)$ or more specifically: $\mathsf{Enc} : \{0,1\}^{\ell} \times \{0,1\}^{n} \rightarrow \{0,1\}^{n}, (k, m) \mapsto c$. The output of a block cipher function is a ciphertext that conceals the original message, and its contents may be revealed only through a process called decryption. To decrypt, we use the same block cipher function in reverse, the ciphertext of our original message, and the correct symmetric key to transform the ciphertext into the original plaintext message: $m = \mathsf{Dec}(k, c)$ and formally: $\mathsf{Dec} : \{0,1\}^{\ell} \times \{0,1\}^{n} \rightarrow \{0,1\}^{n}, (k, c) \mapsto m$. Decryption and encryption with a symmetric block cipher primitive are the inverse of each other: $m == \mathsf{Dec}(k, \mathsf{Enc}(k, m))$.

In Section 8.3.1 we utilize Rijndael [DR02b] as a block cipher with a 256 bit block size, and 256 bit key size (i.e. $n = 256, \ell = 256$).

3.5 — Symmetric Encryption: stream cipher

Another method to encrypt data to achieve confidentiality between a sender and a receiver who share a symmetric key is to use a *stream cipher*. A stream cipher uses a symmetric key to produce outputs one bit or more at a time and is used to encrypt messages of arbitrary length.

As with block ciphers, specifics depend heavily on the design of the stream cipher's internals. The output of a stream cipher function is a *key stream* K and the i'th byte is represented as K_i. The key stream conceals the original message when each byte of the key stream is xor'ed, i.e., $c_i = K_i \oplus m_i$, with the corresponding plaintext byte, creating a ciphertext stream c of the same length as m with c_i for the i'th byte. Its contents may be revealed only through decryption.

To decrypt, we use the same stream cipher function, i.e., $m_i = K_i \oplus c_i$, the ciphertext c of our original message m, and the correct symmetric key k to transform the ciphertext into the original plaintext message.

A modern stream cipher produces a byte of keystream K_i from key k, nonce n, byte index i. The index i allows for finding an arbitrary position in the key stream. The nonce n is a number that is only used once [1], i.e. only for one message per key.

In Section 8.3.1 we utilize ChaCha20 [Ber08a, Ber08c] as a stream cipher as part of an Authenticated-Encryption with Associated-Data scheme. ChaCha20 uses a 256 bit key, 128 bits for the nonce and block counter together, and produces blocks of 512 bits as an output.

3.6 — Message Authentication Code

In addition to confidentiality, message integrity is an important security property. We want to ensure that a message has not been modified by an adversary. To add integrity and authenticity protection, we want to add an authentication tag to our message's ciphertext. The function producing this tag is called a Message Authentication Code (MAC). The authentication tag allows each party to verify that the message has not been tampered with during transit. To combine a MAC with encryption there are two basic options: MAC then encrypt, or encrypt then MAC. An advantage of encrypt then MAC is that modified messages may be quickly detected and are never decrypted.

---

[1] This is also sometimes called a message number. The nonce is typically a random number but may also be a monotonic counter of messages.

To produce an *authentication tag*, we use a MAC function that takes a symmetric key and a message as inputs: $t = \mathsf{MAC}(k, m)$, the output is the authentication tag. Formally: $\mathsf{MAC} : \{0,1\}^{\ell} \times \{0,1\}^{*} \to \{0,1\}^{n}, (k, m) \mapsto t$.

An example construction might result in generating a tag as $t = \mathsf{MAC}(k, m)$ but we want Laura and Julian to have confidentiality of the plaintext message, so we actually define the MAC over the ciphertext as $t = \mathsf{MAC}(k2, \mathsf{Enc}(k1, m))$. It is important that we generate a MAC tag *over the ciphertext* and not the original plaintext message to ensure that tampering with the ciphertext will result in a failure to verify the tag. If the MAC tag is over the plaintext rather than the ciphertext, an adversary can tweak the ciphertext and decryption will happen before the MAC can be verified – this often leads to highly practical attacks where the original message may be recovered by an adversary.

We define the MAC verification function as a boolean function that is either true or false: $r = \mathsf{verify\text{-}MAC}(k, c, t)$ and formally: $\mathsf{verify\text{-}MAC} : \{0,1\}^{\ell} \times \{0,1\}^{*} \times \{0,1\}^{n} \to r, (k, c, t) \mapsto r$ where $r = 1$ if verification fails, else $r = 0$ to signal success. When an authentication tag is needed, Poly1305 [Ber05] is used in Section 3.7 inside a standard Authenticated-Encryption with Associated-Data (AEAD) construction, and indirectly in Section 8.3.1.

3.7 — Authenticated-Encryption with Associated-Data (AEAD)

We use a standard authenticated-encryption with associated-data (AEAD) [Rog02, NL15] construction to encrypt messages to achieve confidentiality, integrity, and authenticity [2]. The AEAD also provides authenticity and integrity of an optional associated piece of data. Associated Data (AD) may provide important integrity protections while the AD is otherwise unencrypted. For example, the AD may cover an unencrypted portion of a message such as an address in a header field which may be used to reply to a message.

The AEAD encryption function takes several inputs: a symmetric key k, the plaintext message m, associated-data a and the nonce n, and outputs a ciphertext c and a MAC tag t over the ciphertext c and the associated data a, i.e. $(c, t) = \mathsf{aead\text{-}enc}(k, m, n, a)$. Formally: $\mathsf{aead\text{-}enc} : \{0,1\}^{\ell} \times \{0,1\}^{*} \times \{0,1\}^{b} \times \{0,1\}^{*} \to \{0,1\}^{*} \times \{0,1\}^{z}, (k, m, n, a) \mapsto (c, t)$.

To decrypt, we define $m = \mathsf{aead\text{-}dec}(k, c, t, n, a)$ with the understanding that either the plaintext of the original message is returned or an error is raised indicating that authentication failed (i.e. $\bot$). Formally: $\mathsf{aead\text{-}dec} : \{0,1\}^{\ell} \times \{0,1\}^{*} \times \{0,1\}^{z} \times \{0,1\}^{b} \times \{0,1\}^{*} \to \{0,1\}^{*} \cup \{\bot\}, (k, c, t, n, a) \mapsto m$. The $m == \mathsf{Dec}(k, \mathsf{Enc}(k, m))$ definition changes to $m == \mathsf{aead\text{-}dec}(k, \mathsf{aead\text{-}enc}(k, m, n, a), n, a)$ to account for the extra parameters [3]. The a parameter is usually a hash digest of whatever data needs to be authenticated.

---

[2] The good CIA.

[3] This includes an implicit tuple unpacking for the sake of simplicity.

Payloads for data are protected with the use of an Authenticated-Encryption with an Associated-Data (AEAD) construction utilizing ChaCha20 [Ber08a,Ber08c] for encryption and Poly1305 to create an authentication tag in Section 8.3.1.

3.8 — Non-interactive Key Exchange (NIKE)

In our encryption example, we assumed that Laura and Julian simply agreed on a symmetric key in person. This is an impractical assumption. It is much more likely that they will need to agree on a key without being in the same physical space [4]. To indirectly agree on a symmetric key for our encryption process, we require some kind of *key-agreement* process or protocol.

One way to agree on a symmetric key other than by agreeing on it directly is via some computation that is able to *derive* new symmetric keys with a system. We will use a *public key* system [DH76,Mer78]. Usually these systems are referred to as Diffie-Hellman (DH).

Each party generates a *secret key* sk from the space of secret keys $\mathcal{K}$ and a *public key* pk from the space of public keys $\mathcal{P}$. Any reasonable public key system should ensure that it is infeasible to recover a secret key from a public key.

Non-interactive Key Exchange (NIKE) is a type of key-agreement. Two parties must agree on the use of a NIKE system (e.g. "Let's use Curve25519!"), and then they may use that NIKE system to create a symmetric secret key as long as they know each other's respective public key. What makes NIKE special is the ability to derive a shared secret key without *any* additional communication beyond acquiring a public key. It should be computationally infeasible to correctly guess a shared key for a pair of public keys. Usually the agreement of the system is a matter of what users choose to publish without any explicit agreement or communication before using that system to securely communicate. To derive a symmetric key, each party publishes their public key on their own public website or on some public directory of keys. Two parties who have thus obtained each other's public keys are able to, without any direct communication, compute an identical symmetric shared secret k between their secret key and their peer's respective public key. This computation process is known as *deriving* a cryptographic key. This cryptographic key is a symmetric secret value shared between both parties.

Internally, the process of deriving a key using some NIKE scheme includes a step where an intermediate shared element is computed (e.g. k) which is then processed to ensure that the resulting symmetric key is a bit string indistinguishable from a uniformly random bit string. One method to derive a suitable symmetric key from a random element is to process it with an HMAC-based Extract-and-Expand Key Derivation Function (HKDF) [KE10a].

We consider an abstract NIKE function, DH(secret_key, public_key), where the parameters for the scheme are internally fixed with Laura and Julian as an intrepid pair of example users. This function takes a secret key and a public key to compute a shared secret symmetric key. Laura and Julian respectively each generate a keypair for encryption, $\mathsf{dhsk}_\mathsf{L}, \mathsf{dhpk}_\mathsf{L}$ and $\mathsf{dhsk}_\mathsf{J}, \mathsf{dhpk}_\mathsf{J}$. Using Laura's $\mathsf{dhsk}_\mathsf{L}, \mathsf{dhpk}_\mathsf{L}$, and Julian's $\mathsf{dhsk}_\mathsf{J}, \mathsf{dhpk}_\mathsf{J}$, the computation should always return the same secret value. That is to say that if $\mathsf{k} = \mathsf{DH}(\mathsf{dhsk}_\mathsf{J}, \mathsf{dhpk}_\mathsf{L})$ then $\mathsf{k} == \mathsf{DH}(\mathsf{dhsk}_\mathsf{L}, \mathsf{dhpk}_\mathsf{J})$ holds; i.e. both DH return the same shared secret k. Formally: $\mathsf{DH} : \mathcal{K} \times \mathcal{P} \rightarrow \{0,1\}^\ell, (\mathsf{sk}, \mathsf{pk}) \mapsto \mathsf{k}$.

---

[4] For example, Julian is in the high security Bellmarsh Prison and Laura is in her New York City apartment.

NIKE has an interesting and non-obvious property that is often called deniability or sometimes repudiation. Either party arrives at the same symmetric shared secret and thus either party alone could have derived the symmetric shared secret value. Finding a message on Julian's laptop that decrypts with the key shared between him and Laura is not definitive cryptographic proof of any communication between Julian and Laura as he could have computed this from his own secret value and Laura's public key.

We use two concrete NIKE constructions for our protocol designs in subsequent chapters: Curve25519 [Ber06] and CSIDH [CLM+18a] such as in Section 8.3.1. Curve25519 uses Elliptic Curve Cryptography (ECC) [Mil85, Kob87] to create a NIKE system. CSIDH is a NIKE and internally it is not using Diffie-Hellman in groups, but a more abstract way of using group actions. CSIDH is a candidate for resisting adversaries with quantum computers. It is one of very few post-quantum candidates for NIKE.

3.9 — Verification of public keys

To be secure users must *verify* that the public keys were properly obtained. Laura must carefully check Julian's public key through some out-of-band method to ensure it has not been substituted or modified during transmission, and Julian must do the same for Laura's public key. Checking means that they confirm that the bit string representation of the public key is identical from either perspective.

Public keys are safe to reveal by design unless a cryptographic system has fatal mathematical flaws relevant in some adversary model. Public key cryptography systems allow users to solve a basic cryptographic session establishment problem by transmitting a public key through an otherwise insecure channel. This same public key may then be used to establish a secure cryptographic session. However, absent public key verification, there may be no security as the public keys may be completely replaced by an attacker. Even when public keys are exchanged inside of a cryptographic session, one must ask about the root of trust. Mallory is seen in Figure 3.1 as a classic Machine-in-the-Middle (MITM) impersonating Laura to Julian, and Julian to Laura.

$$\mathsf{Laura} \xleftrightarrow{\mathsf{DH}(\mathsf{sk}_L,\mathsf{pk}_M)} \mathsf{Mallory} \xleftrightarrow{\mathsf{DH}(\mathsf{sk}_J,\mathsf{pk}_M)} \mathsf{Julian}$$

Figure 3.1: Example: Mallory in the middle. There is *no security* for either Laura or Julian.

The verification step is error prone, yet extremely important, and without it, we cannot be sure that the encryption system is not being used with an adversary who wishes to violate or break the properties we desire for secure communication. The adversary need only substitute their own public key for Laura's public key to Julian, and for Julian's public key to Laura; now each person encrypts to the adversary Mallory, but thinks they are talking to each other.

3.10 — Signatures

Digital signatures allow an author of a message to assert that they are the author of the message, and any change to the message will invalidate the signature. A digital signature is verifiable by any party, including third parties, as long as the verifier has the corresponding signature public key, the message, and the signature over the message. Signatures have an interesting and obvious property that is often called non-repudiation, which is the opposite of the deniability property as mentioned in Section 3.8. Any reasonable signature system should ensure that it is computationally infeasible to forge signatures for a given public key, and similarly it should be infeasible to recover a secret key from a public key or a signature.

Signing takes a message and the signer's secret key and produces a signature $s = \mathsf{Sign}(sk, m)$ and formally: $\mathsf{Sign} : \mathcal{K} \times \{0,1\}^* \to \mathcal{S}, (sk, m) \mapsto s$.

We define verification as a function that returns a boolean value that maps to true or false: $r = \mathsf{Vf}(pk, m, s)$ and formally: $\mathsf{Vf} : \mathcal{P} \times \{0,1\}^* \times \mathcal{S} \to \{0,1\}, (pk, m, s) \mapsto r$. We map 0 to indicate successful signature verification and 1 to mean failure to verify the signature.

A message may be constructed that starts with a user generating a *fresh* keypair for use in a NIKE, followed by a signature s_i using their long term, verified signing key over the new NIKE public key.

In the following protocol, Laura and Julian each have generated a keypair for signatures, sk_L, pk_L and sk_J, pk_J, and have verified their respective public keys. The first round of the i-th run of the protocol has Laura send an unencrypted message that consists of three parts $date_{i,L}, dhpk_{i,L}, s_{i,L}$; similar comments apply to Julian. The signature from Laura is composed as $s_{i,L} = \mathsf{Sign}(sk_L, date_i \| dhpk_{i,L})$ and thus the signature covers the time, as well as the new public key $dhpk_{i,L}$. Julian composes similar messages as Laura with his respective $dhpk_{i,J}$ public key. Laura receives Julian's first round message, parses it, and then verifies that the signature is valid, and Julian does the same for Laura's first round message. Once the first round messages have been processed successfully, they both move on to the second round.

The second round uses Laura's $dhsk_{i,L}$ secret key with Julian's $dhpk_{i,J}$ public key to compose a message using the AEAD construction to protect the message:

$$k_i = \mathsf{DH}(dhsk_{i,L}, dhpk_{i,J}); c_{i,L}, t_{i,L} = \text{aead-enc}(k_i, m_{i,L} \| s_{i,L}, n_{i,L}, a_{i,L}).$$

Julian can compute k_i by using his secret key and Laura's public key. At this point they can exchange messages using an AEAD. Encrypted messages could also include a new, fresh substitute for pk_L or pk_J respectively, such that the original signature keypair is only used initially.

If Laura and Julian naturally erase their respective secret keys, $dhsk_{i,L}$ and $dhsk_{i,J}$, as well as the session key, k_i, after the end of the session, the messages sent across the wire are no longer decryptable unless the overall public key system is broken. This concept is generally called *forward secrecy* which is useful if an adversary subsequently gains access to the user devices that are performing the encryption, the adversary will be unable to use the remaining keys to decrypt past session data. The term *key erasure* is a more expansive concept where keys have a limited, defined lifetime, such as a session key that is actively erased by the cryptosystem when the session or the lifetime has ended.

The example system looks like the following in detail where the i subscript indicates the round of the protocol:

0. Laura and Julian each generate keypairs for signatures: sk_L, pk_L and sk_J, pk_J.
1. They do not meet but rather they exchange signing public keys over the Internet.
2. They both verify out-of-band, manually, that they have obtained the correct signing public key for their peer: pk_L and pk_J.
3. a) Each generates a fresh Curve25519 keypair for encryption: $dhsk_{i,L}, dhpk_{i,L}$ and $dhsk_{i,J}, dhpk_{i,J}$.
   b) Laura signs the date and ephemeral public key: $s_{i,L} = \mathsf{Sign}(sk_L, date_{i,L} \| dhpk_{i,L})$.
   c) Laura transmits her first round message including the signature: $R1_{i,L} = date_{i,L}, dhpk_{i,L}, s_{i,L}$.
   d) Julian receives $R1_{i,L}$ and verifies using Laura's public key. He confirms that the signature verification is valid: $r_{i,L} = \mathsf{Vf}(pk_L, date_{i,L} \| dhpk_{i,L}, s_{i,L})$.
   e) Julian confirms that the date is reasonably recent, unique, and a higher value than ever seen previously to prevent replayed messages.
   f) Julian signs his date and ephemeral public key: $s_{i,J} = \mathsf{Sign}(sk_J, date_{i,J} \| dhpk_{i,J})$.
   g) Julian responds with his first round message: $R1_{i,J} = date_{i,J}, dhpk_{i,J}, s_{i,J}$.
   h) Laura receives $R1_{i,J}$ and verifies that it was correctly signed by Julian: $r_{i,J} = \mathsf{Vf}(pk_J, date_{i,J} \| dhpk_{i,J}, s_{i,J})$.
   i) Laura confirms that the date is reasonably recent, unique, and a higher value than ever seen previously to prevent replayed messages.
   j) Laura composes $m_{i,L}$ and then transmits her $R2_{i,L} = c_{i,L}, t_{i,L}$:
      $k_i = \mathsf{DH}(dhsk_{i,L}, dhpk_{i,J})$
      $(c_{i,L}, t_{i,L}) = \mathsf{aead\text{-}enc}(k_i, m_{i,L}, n_{i,L}, a_{i,L})$.
   k) Julian receives Laura's $R2_{i,L}$, he computes $k_i = \mathsf{DH}(dhsk_{i,J}, dhpk_{i,L})$ and verifies that $m_{i,L} = \mathsf{aead\text{-}dec}(k_i, c_{i,L}, t_{i,L}, n_{i,L}, a_{i,L})$ decrypts successfully.
   l) At this point Julian and Laura have established a secure communications channel. They are able to transmit any number of messages encrypted using AEAD. However they agree that after a fixed short time, e.g. three minutes, they will destroy all session keying material.
   m) Both parties destroy all data related to secret keying material: $k_i, dhsk_{i,L}, dhsk_{i,J}, dhpk_{i,L}, dhpk_{i,J}$.
4. Either party can begin again by generating new DH keypairs. They would need to re-run the protocol, sign a new R1 message that contains their new $dhpk_{j,L}$ or $dhpk_{j,J}$ public keys respectively, and then they would be able to generate a new R2 message with fresh keys.

Generally, long term key pairs are used to produce signatures with some notion of identity and shorter lived session key pairs are used to produce encryption keys. The above example uses only one level of keys for simplicity in understanding an example use of signatures. Many but not all deployed protocols forego using third party checkable signatures.

For compact and fast signatures, we use Ed25519 [BDL+11] in Section 7.3.4. However, verifying that a signature corresponds to a given public key is only part of the story – we stress that users must also verify that public keys belong to specific entities for the signature to have any practical notion of security. We also note that other constraints are visible in steps 3e and 3i.

3.11 — Protocols from building blocks

We think of each cryptographic primitive mentioned above as a mathematical building block with which we build our network protocols. Each selection is based on some mathematical problem considered to be *hard* [5].

In the GNU Name System described in Chapter 5 we describe the use of Curve25519, Ed25519, and various Internet protocols related to resolving hostnames on the Internet in an improved, privacy preserving manner.

The WireGuard protocol combines BLAKE2, ChaCha20, Curve25519, Poly1305, and SipHash [AB12]. The Tiny WireGuard Tweak in Chapter 6 introduces a single additional application of the BLAKE2 hash function to create resistance to attackers with quantum computers who record traffic and wish to attack it to recover and decrypt previously protected IP packets. We show that even full packet captures of entire WireGuard sessions are not sufficient when the Tiny WireGuard Tweak is applied.

We introduce the new Vula protocol and a Python implementation for GNU/Linux that automatically encrypts local-area IP network traffic. Vula, introduced in Chapter 7, combines CSIDH, Ed25519, and WireGuard as stated previously to build a fully automatic, peer to peer, local-area network traffic encryption system without the use of trusted third parties, and without any centralization, also known as a Single Point of Failure (SPOF). In Section 7.4.3, the parameters for WireGuard device and peer configuration are combined with Ed25519 signatures. CSIDH is used by Vula to derive a symmetric key which is used to augment WireGuard's normal cryptographic protections against an adversary with a quantum computer.

The new REUNION protocol is introduced in Chapter 8 and we release a Python implementation for use on a local-area IP network. REUNION is a protocol designed to help people rendezvous in various networks in a privacy preserving manner. REUNION may be used to perform peer discovery and/or peer verification using an easy to remember passphrase. REUNION combines a *user chosen*, potentially low-entropy, passphrase with a public random value to improve on previous rendezvous solutions such as PANDA [Lan12a]. It is designed to be safe with a passphrase that is easy to remember, and it is also safe to disclose this passphrase after use. One of the primary goals is to ensure that everyday *pocket litter* [Wik21n] [Nak08b] [RP14, Identity intelligence:

---

[5] It turns out that it is inadvisable to write "Thanks for the math! Factoring is hard, lets go shopping!" in the guestbook when visiting the mathematics exhibits of the Pyongyang Science and Technology Park of the Democratic People's Republic of Korea (DPRK). After all, one must remember: "Nothing is difficult for the Great Leader!" and of course: "In the face of adversity it is never correct to go shopping!"

Image Is Everything] is not used against people who use this protocol even if they write the passphrase down on a piece of paper. This is in strong contrast to common methods of rendezvous such as exchanging a simple business card. The passphrase used by any number of users at the same time may be used to securely rendezvous by sending a single payload message to any number of users who share the same passphrase. A third party observer or participant is unable to know who was able to successfully decrypt a message except when that third party guesses the passphrase correctly during the protocol run. Protocol runs may be recorded by an adversary who is delayed in using a passphrase guess during the run of the protocol. After the protocol run, even a perfect log of the full protocol run is not useful for breaking the confidentiality of the payload message. The passphrase is processed with argon2. The resulting key is used with Rijndael to conceal Curve25519 public keys encoded with Elligator 2 [BHKL13]. CSIDH is used to produce a secure, transitionally post-quantum cryptographic protocol. The payload is protected with a standard AEAD construction consisting of ChaCha20, and Poly1305. REUNION is a secure, transitionally post-quantum cryptographic protocol for private rendezvous.

CHAPTER 4

The Adversary

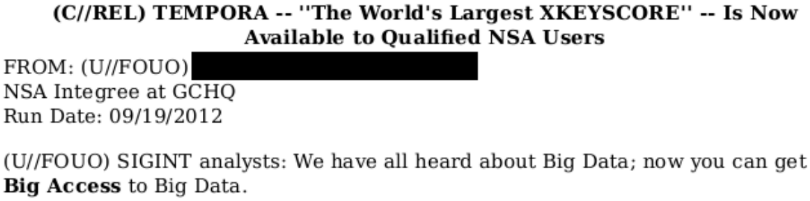
> *"What does a scanner see? he asked himself. I mean, really see? Into the head? Down into the heart? Does a passive infrared scanner like they used to use or a cube-type holo-scanner like they use these days, the latest thing, see into me–into us–clearly or darkly? I hope it does, he thought, see clearly, because I can't any longer these days see into myself. I see only murk. Murk outside; murk inside. I hope, for everyone's sake, the scanners do better. Because, he thought, if the scanner sees only darkly, the way I myself do, then we are cursed, cursed again and like we have been continually, and we'll wind up dead this way, knowing very little and getting that little fragment wrong too."*
>
> — Philip K. Dick, *in his 1977 book* ***A Scanner Darkly*** *[Dic77].*

This chapter documents real adversary capabilities by highlighting internal documents and reporting about various groups that carry out targeted and mass surveillance. It is not yet a comprehensive survey of every possible threat actor or capability but we hope that this chapter will contribute to an overall understanding at some point in the future. We start our focus with the United States of America's National Security Agency (NSA). We consider various internal organizations of the NSA such as the Office of Tailored Access Operations (TAO) and TAO's later incarnation as the Computer Network Operations (CNO) [Wik20b] by examining the Advanced Network Technologies Catalog (ANT catalog) [AHS13,APR+13b] in Section 4.6. We also consider substantial discussion about the United States of America's Central Intelligence Agency (CIA), and the United States of America's Federal Bureau of Investigation (FBI). Other government agencies such as the United States Drug Enforcement Agency (DEA), American domestic police forces, as well as others in and outside of the United States. Non-American agencies are also considered.

The capabilities discussed in this chapter span several different groups who often work together in various capacities. The Adversary is shorthand for specific *sets of capabilities*, as these capabilities are often fundamentally in conflict with the basic law in most countries, even when the deploying agency or party does not think so. We consider certain capabilities that are in conflict with basic constitutional liberties to be a core problem. These capabilities create conditions for what has been called by former NSA SIGINT expert Bill Binney as *turnkey tyranny*. Thus while we may be able to reason about the state of the world today as not tyrannical, these systems create the conditions by which policies could change under an anti-democratic leader and thus extinguish liberty. The lessons of Europe's early 20th century democratic experiments tell us that this can happen nearly overnight [Sny17]. Does it make sense to create systems to protect democracy that are so powerful that they could be used to destroy it? Must technology enable such a policy change, or might we use technology, such as non-escrowed cryptography, to ensure liberty? There are many possible answers and this thesis proposes the use of cryptography, in concert with other tactics and strategies, to enhance, expand, and maintain liberty.

While many people are culpable in building, deploying, and using these systems of targeted and mass surveillance, we consider the capability itself to be the most important problem at the moment. In the future we should almost certainly create a process similar to the South African Truth and Reconciliation Commission (TRC) to deal with people who built systems to enable war crimes, and to recognize as well the people harmed by those systems. Yet at the current point in history, this probably cannot be safely accomplished; too many people with access to these capabilities still wield the power to crush their political opponents and perhaps to destroy democracy itself.

Humanity no longer lives in a unipolar world where the United States of America sets rules and nearly every country on the planet follows. We can expect that the surveillance capabilities that we tolerate in this new world, and the surveillance capabilities that we do not technically thwart, will eventually be deployed by adversaries who lack our respect for the liberal democratic values. Within our own countries, in the framework of natural rights, we grant our governments wide-ranging authorities. A government has no right to exercise powers not granted to it, because government authority is predicated on the consent of the governed. Absent public knowledge and consent, any authority exercised violates the pact between citizens and the state. It follows that many of these secret programs break that pact, and therefore in this thesis we encourage non-violent direct action, in the form of research and development of cryptographic protocols, to thwart surveillance, and to build secure systems to protect basic human rights. Thus it is not the NSA, FBI, or CIA that constitute the Adversary; it is their political and technical ability to break the basic agreements that form the basis of a free society, a capability, indeed power, too great for any group to possess. We must ensure that, at a later date, no person can reasonably say that they did not know what their government was doing.

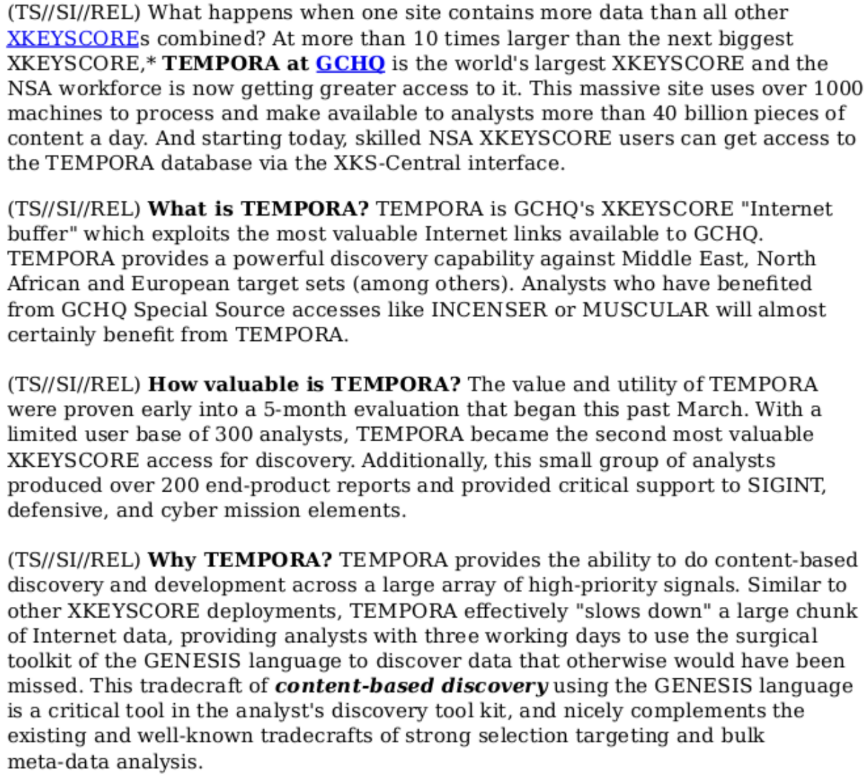


(C//REL) TEMPORA -- "The World's Largest XKEYSCORE" -- Is Now Available to Qualified NSA Users

FROM: (U//FOUO) [redacted]
NSA Integree at GCHQ
Run Date: 09/19/2012

(U//FOUO) SIGINT analysts: We have all heard about Big Data; now you can get **Big Access** to Big Data.

Figure 4.1: TEMPORA: Big data ... Big access
Courtesy of GCHQ and NSA [Uni13b].



(TS//SI//REL) What happens when one site contains more data than all other XKEYSCOREs combined? At more than 10 times larger than the next biggest XKEYSCORE,* **TEMPORA at GCHQ** is the world's largest XKEYSCORE and the NSA workforce is now getting greater access to it. This massive site uses over 1000 machines to process and make available to analysts more than 40 billion pieces of content a day. And starting today, skilled NSA XKEYSCORE users can get access to the TEMPORA database via the XKS-Central interface.

(TS//SI//REL) **What is TEMPORA?** TEMPORA is GCHQ's XKEYSCORE "Internet buffer" which exploits the most valuable Internet links available to GCHQ. TEMPORA provides a powerful discovery capability against Middle East, North African and European target sets (among others). Analysts who have benefited from GCHQ Special Source accesses like INCENSER or MUSCULAR will almost certainly benefit from TEMPORA.

(TS//SI//REL) **How valuable is TEMPORA?** The value and utility of TEMPORA were proven early into a 5-month evaluation that began this past March. With a limited user base of 300 analysts, TEMPORA became the second most valuable XKEYSCORE access for discovery. Additionally, this small group of analysts produced over 200 end-product reports and provided critical support to SIGINT, defensive, and cyber mission elements.

(TS//SI//REL) **Why TEMPORA?** TEMPORA provides the ability to do content-based discovery and development across a large array of high-priority signals. Similar to other XKEYSCORE deployments, TEMPORA effectively "slows down" a large chunk of Internet data, providing analysts with three working days to use the surgical toolkit of the GENESIS language to discover data that otherwise would have been missed. This tradecraft of ***content-based discovery*** using the GENESIS language is a critical tool in the analyst's discovery tool kit, and nicely complements the existing and well-known tradecrafts of strong selection targeting and bulk meta-data analysis.

Figure 4.2: What is TEMPORA?
Courtesy of GCHQ and NSA [Uni13b].

The global span of surveillance systems reveals a larger conspiracy of the kind seen in Figure 4.1 and Figure 4.2, where intelligence agencies make their own international agreements [Mai20] among themselves, and partner with private-sector corporations that maintain infrastructure for the largely privatized Internet. The Government Communications Headquarters (GCHQ [Cam82]) of the United Kingdom is an example of such a partner with their XKeyscore 4.5 deployment that is codenamed TEMPORA [BHG13, BBG13] [Uni13b, "Big Access to Big Data"] [And20, Chapter 3: Who is the Opponent?]. As revealed by the Guardian [BHG13, BBG13] the Corporate Surveillance Partners for TEMPORA were given code names. This incomplete list of partners includes DACRON for Verizon Business, GERONTIC for Vodafone Cable, LITTLE for Level 3, PINNAGE for Global Crossing, REMEDY for British Telecom, STREETCAR for Interoute, and VITREOUS for Viatel. GCHQ's partnership with the NSA goes far beyond XKeyscore. Documented operations and training manuals describe explicit psychological operations [Gre14a]. GCHQ's Joint Threat Research Interest Group (JTRIG) regards the Internet as the primary space of their operational activities, and stating that their interference should ideally have offline impact on individual human beings.

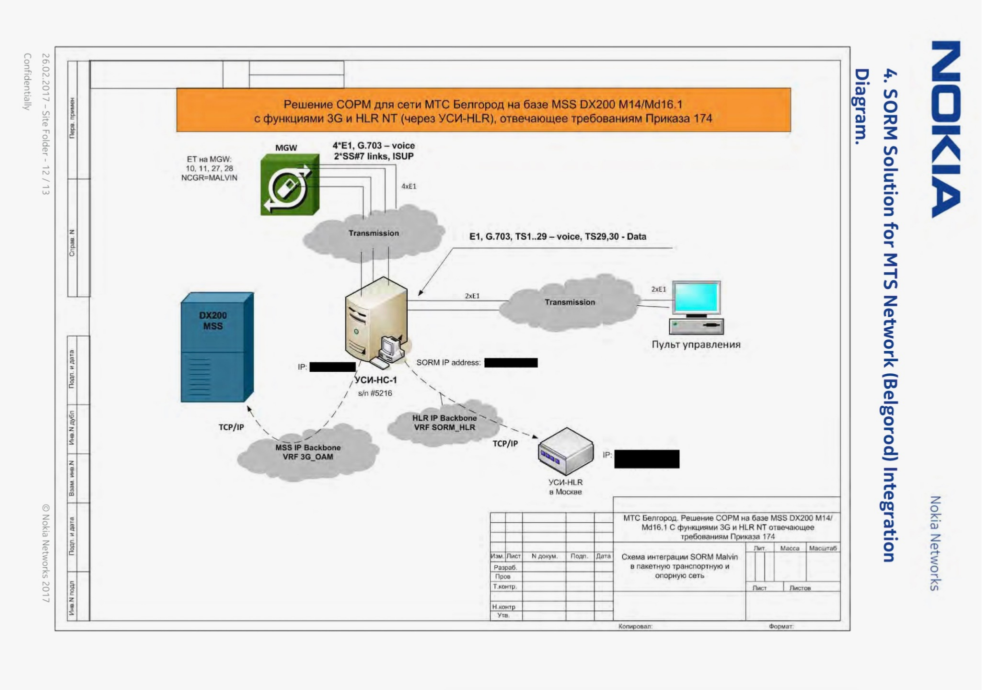

NOKIA
4. SORM Solution for MTS Network (Belgorod) Integration Diagram.
Nokia Networks
26.02.2017 – Site Folder - 12 / 13
Confidentially
© Nokia Networks 2017
Решение СОРМ для сети МТС Белгород на базе MSS DX200 M14/Md16.1
с функциями 3G и HLR NT (через УСИ-HLR), отвечающее требованиям Приказа 174
MGW
ET на MGW:
10, 11, 27, 28
NCGR=MALVIN
4*E1, G.703 – voice
2*SS#7 links, ISUP
4xE1
Transmission
E1, G.703, TS1..29 – voice, TS29,30 - Data
2xE1
Transmission
2xE1
Пульт управления
DX200
MSS
IP: [redacted]
SORM IP address: [redacted]
УСИ-НС-1
s/n #5216
TCP/IP
MSS IP Backbone
VRF 3G_OAM
HLR IP Backbone
VRF SORM_HLR
TCP/IP
IP: [redacted]
УСИ-HLR
в Москве
МТС Белгород. Решение СОРМ на базе MSS DX200 M14/Md16.1 С функциями 3G и HLR NT отвечающее требованиям Приказа 174
Схема интеграции SORM Malvin в пакетную транспортную и опорную сеть
Изм.
Лист
N докум.
Подп.
Дата
Разраб.
Пров
Т.контр.
Н.контр
Утв.
Лит.
Масса
Масштаб
Лист
Листов
Копировал:
Формат:
Перв. примен.
Справ. N
Подп. и дата
Инв N дубл.
Взам. инв N
Подп. и дата
Инв N подл.

Figure 4.3: SORM network floor map
Courtesy of Nokia Networks and TechCrunch [Whi19].

In other cases, public/private surveillance takes the form of telecom backdoor requirements, such as the SORM family of systems [SB13, Wik17b, Whi19] [Kri18] in Russia. Corporations in Russia are required to grant access as seen in Figure 4.3 to agencies such as the Russian Federation's Federal Security Services (FSB) for deploying and using the SORM family of interception systems. In Egypt, we see that the Technical Research Department [Int16] plays a key intelligence role in surveillance and control of the Egyptian Internet with the help of Nokia, FinFisher, HackingTeam, and Narus. In China, the Cyberspace Administration of China (CAC) is primarily responsible for the surveillance and censorship operations on Chinese networks under the Golden Shield Project. The CAC is under the State Internet Information Office (SIIO) and the Office of the Central Cyberspace Affairs Commission. Companies from the United States have helped sell equipment to support the Golden Shield Project, and indeed a confidential sales-pitch deck [Sti08] from Cisco Systems leaked to the public. The slides explicitly repeat a technological policy goal of targeting a religious minority [Sys02, see slide 57 for details]. The Great Firewall of China is principally operated by the CAC. It has similar capabilities to systems such as XKeyscore as mentioned in Section 1.2 and with source code examples in Section 4.5 while being deployed to primarily monitor and interfere with Chinese networks. Certain aspects of the Chinese active censorship systems, such as the Great Cannon, are used to attack public systems outside of China [1].

---

[1] Unlike the authors of systems such as XKeyscore or PRISM, the primary architects of the Great Firewall and the Golden Shield project are known and promoted in public. Fang Binxing is considered to be the father

While the corporate collaboration of Cisco with the Chinese authorities is grim, it is far from unique even in nominally democratic countries. For Operation Rubikon, also known as Operation Thesaurus, the German Bundesnachrichtendienstes (BND) and the US CIA worked together (with the knowledge of the Swiss intelligence service Schweizer Nachrichtendienste (SND)) to install backdoors in encryption devices sold by Swiss firm CryptoAG. CryptoAG's market targets were largely governments without expertise to develop their own cryptographic hardware and software. The BND and the CIA held secret co-ownership of CryptoAG until 1993, and then the CIA held sole ownership until 2018. The devices were vulnerable by design, which allowed unaffiliated intelligence services, such as the former USSR's KGB, and the East German Ministry for State Security, to independently exploit CryptoAG's intentional flaws. [2] A Dutch example is provided by Bart Jacobs on the European signals intelligence alliance known as the Maximator [Jac20] partnership. Maximator is essentially a tiny European signals intelligence agreement that, while fascinating and interesting historically, is not deeply relevant to planetary mass surveillance activities. The Maximator agreement can be thought of as a few European countries who share physical borders also sharing intelligence data, and generally keeping their lines of communication open. It does not appear that those party to the Maximator alliance are using their agreement and relative positions to spy on the entire planet – in stark contrast to the Five-Eyes agreement.

Nearly all so-called lawful interception systems serve an unintended dual function: they allow current, lawful authorities to surveil their own telecommunications infrastructure, but, perversely, they also weaken that infrastructure by giving foreign spies a tempting target. As part of our research, we uncovered evidence that the telecommunications infrastructure in many countries has been compromised by intelligence services. The Snowden archive includes largely unpublished internal NSA documents and presentations that discuss targeting and exploiting not only deployed, live interception infrastructure, but also the vendors of the hardware and software used to build the infrastructure. Primarily these documents remain unpublished because the journalists who hold them fear they will be considered disloyal or even that they will be legally punished. Only a few are available to read in public today. Targeting lawful interception (LI) equipment is a known goal of the NSA [3]. Unpublished NSA documents specifically list their compromise of the Russian SORM LI infrastructure as an NSA success story of compromising civilian telecommunications infrastructure to spy on targets within reach of the Russian SORM system. The NSA slides have "*you talk, we listen*" written in Cyrillic on the jackets of two Russian officers. [4]. It is not unreasonable to assume that parts of, if not the entire

---

of the Great Firewall of China, and Shen Changxiang is considered to be the primary technical mind behind the Golden Shield Project.

[2] There have been other important contemporary and historical government spying issues in Switzerland, including the Onyx interception system [Wik20a], essentially the Swiss version of ECHELON, and the Secret Files Scandal [Wik21a]. Both demonstrated mass surveillance against the domestic population of Switzerland by the Swiss government authorities based on the targeted persons' political affiliation. An older Swiss student of the thesis author shared that his father was targeted by this program, which led his deep fascination with surveillance in democratic countries (Personal communication). This mirrors the thesis author's personal experience with their own family's Internet being subject to mass surveillance in San Francisco, California, which created a personal drive to explore the topic that has lasted for nearly two decades.

[3] See *(S//SI//REL) Exploiting Foreign Lawful Intercept (LI) Roundtable* [Unk15b] and *The Greek wiretapping scandal and the false promise of intelligence cooperation in the information era* [Pap18]

[4] Review of unpublished Snowden documents about NSA's activities compromising *deployed*, lawful inter-

American CALEA infrastructure have been compromised by similarly skilled adversaries and we merely lack the confirmation in public. Key European LI systems have been compromised by NSA and/or GCHQ. In Athens, such electronic intrusion carried physical consequences: when a telecom employee was found dead [Bam16], the Vodafone lawful interception capabilities were found to have been abused, with the CIA as the likely culprit. Another example, the Belgacom intrusion [Gal14a, Gal18] was codenamed *Operation Socialist* by GCHQ and NSA. This operation was against an EU member state's publicly owned telecommunications company, is another success of targeted hacking.

This chapter is intended to illustrate the industrial nature of spying infrastructure and its tooling. This infrastructure and these tools do not exist in a vacuum. The purpose of having access to such infrastructure and tools is to get data. The data is to support a government's operational capabilities, some legitimate, some illegitimate, some overtly illegal nationally and/or internationally. The example documents in this chapter show real adversary capabilities, real adversary goals, and each discussed program shows the *desired direction of travel* of electronic surveillance activities for *any* competent adversary, especially those unburdened by rules or accountability. Surveillance, once considered a largely passive process, must be considered as an active process. Passive collection of data from data transit lines such as the TAT-14 cable interception by GCHQ [PRS13a] as well as satellite communications interception [PRS13a, BCRT15] (FORNSAT) is still a cornerstone of mass surveillance activities, but the overall space of surveillance is much larger than simple passive collection. It includes things such as human infiltration and various kinds of sabotage, as well as using surveillance data for specific political aims, including reducing legal restraints on the spies collecting the data. While we understand the view of electronic surveillance as a useful tool of twenty-first century statecraft, we consider that these are merely temporarily useful tools. These tools have additional unintended side-effects, especially when exposed to the general public. Building secure systems is extremely difficult; building human communities to support building secure systems is perhaps just as difficult. The effort documented and discussed in this chapter to break real-world systems, and to impact real people, reflects how much *more* effort is required to secure similar systems from such an adversary. Finding physical evidence of surveillance is common [Ear21, see Section 9.1.3 of [Ear21] for an actual physical device found by the thesis author on an activist's car. [5]].

---

ception systems and as well as additional success against the vendors of such hardware or software. Needless to say, a compromised interception system is anything but lawful in the hands of an adversary.

[5] Personal communications with this activist confirmed that they had previously been targeted by a now exposed but previously undercover British police officer named Mark Kennedy; a literal case [McK19, Woo18, SG21] of State rape according to the activist.

4.1 — Zersetzung or Dirty Tricks?

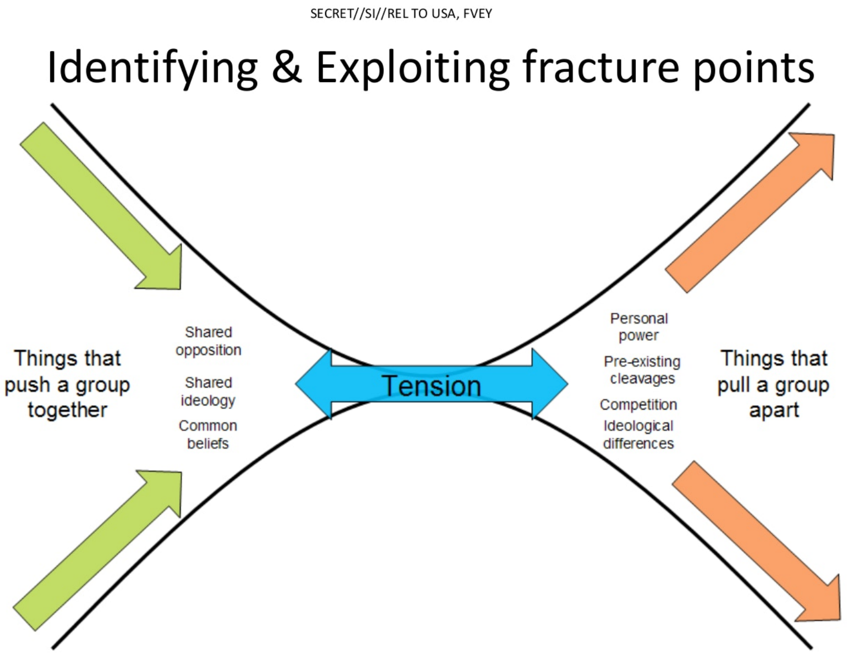
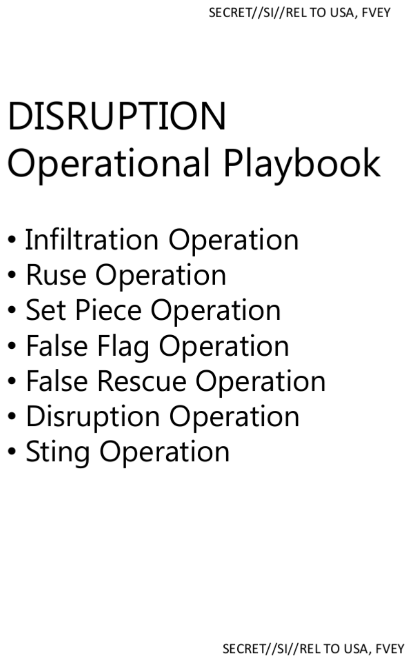

SECRET//SI//REL TO USA, FVEY
Identifying & Exploiting fracture points
Things that push a group together
Shared opposition
Shared ideology
Common beliefs
Tension
Personal power
Pre-existing cleavages
Competition
Ideological differences
Things that pull a group apart

Figure 4.4: JTRIG: identifying & exploiting fracture points
Courtesy of Glenn Greenwald [Gre14a].

It is now well understood that the political aims of surveillance activities may include things such as assassination [MH11, Ram11] by drone strike, but what is not well understood is something equally chilling: psychological operations (PSYOPS). The general scope of PSYOPS, a broad area of active research and development, is beyond the scope of this thesis. Internal documents from the GCHQ's JTRIG describe their mission as "destroy, deny, degrade, disrupt" as seen in Figure 4.7. They additionally discuss the use of *honey traps* as seen in Figure 4.8 to achieve their goals. A honey trap is best understood as a human intelligence (HUMINT) technique that includes a range of activities. An important historical example to set the tone is the Operation Midnight Climax [LS92], as part of Project MKUltra. The CIA paid sex workers in San Francisco, California, to give unsuspecting non-consenting American citizens lysergic acid diethylamide (LSD) as part of Sidney Gottlieb's experiments into mind control. The CIA tried to induce their targets to reveal information while under the influence of various substances, and to commit themselves to performing criminal activities such as assassination. Related but largely unknown is that Theodore Kaczynski [Kac95, Dam03] was also swept up in MKUltra. Without the CIA and LSD, would we ever have had The Unabomber? In the same time frame, the East German Ministry for State Security (Stasi) carried out they termed Zersetzung [Min76, ua, Wik21v] operations against their supposed internal enemies. Zersetzung, a psychological warfare technique literally meaning *decomposition* or *decay*, can be understood as the *operational psychology* used by the Stasi against their real or imagined enemies.



SECRET//SI//REL TO USA, FVEY

DISRUPTION
Operational Playbook

- Infiltration Operation
- Ruse Operation
- Set Piece Operation
- False Flag Operation
- False Rescue Operation
- Disruption Operation
- Sting Operation

SECRET//SI//REL TO USA, FVEY

Figure 4.5: JTRIG: DISRUPTION Operational Playbook
Courtesy of Glenn Greenwald [Gre14a].

JTRIG's methods, as revealed by Glenn Greenwald's reporting [Gre14a], are a modern version of Zersetzung that encompasses the Internet, ensuring that such operations are no longer confined to a specific location. In Figure 4.9 we see that JTRIG is explicitly planning operations not only against individual people but also against companies. JTRIG references *Understanding scam victims: seven principles for systems security* by Stajano and Wilson as UCAM-CL-TR-754 [SW09], a paper later republished by the ACM [SW11]. Grothoff remarked in 2018 [Gro18] that this paper was an important JTRIG reference point. It is remarkable that JTRIG's interest in how people fall for scams is geared toward improving their success rate at targeting people for JTRIG scams. It has an internal logic that is consistent: JTRIG studies scammers to become better scammers. The essence of their job is to trick and harm people for their own political ends.

Lest the reader imagine these tactics are only directed at terrorists with serious plans to commit acts of violence, a recent US Army PSYOPS training manual set includes an example operation to enhance domestic support for policies in the war on (some) drugs. Practices that seem specific to intelligence services and the military can spill over into other institutions. National policing organizations have access to massive amounts of data and the ability to ask the FBI for more information. In Operation Whistle Pig [Win21], exposed by journalist Jana Winter, FBI and United States Customs and Border Protection (CBP) agents sought to recruit journalists as confidential intelligence sources, with the aim of exploiting the journalists' access to feed a beneficial-to-the-CBP narrative to the public. The CBP agents involved used their access to huge repositories of data to investigate the journalists that they planned to target. In the end, the agent involved

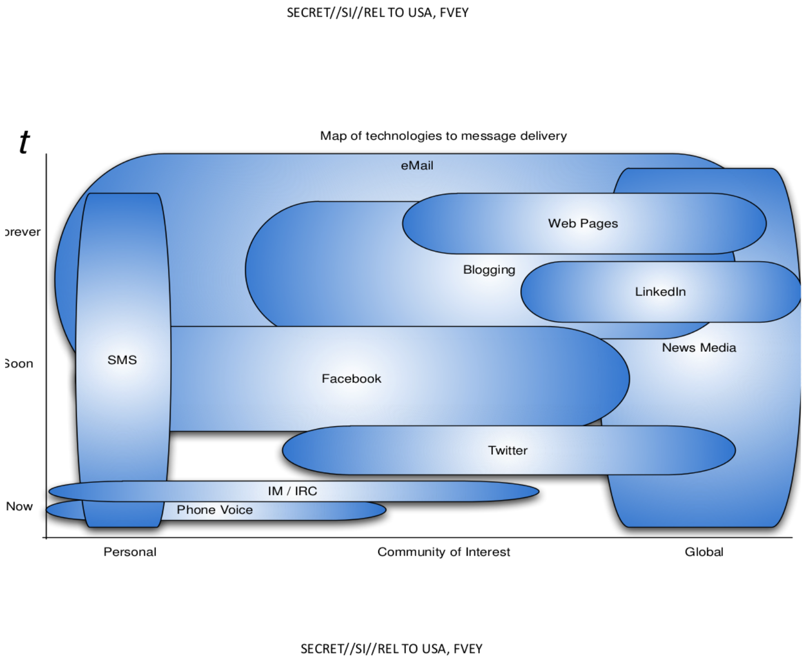
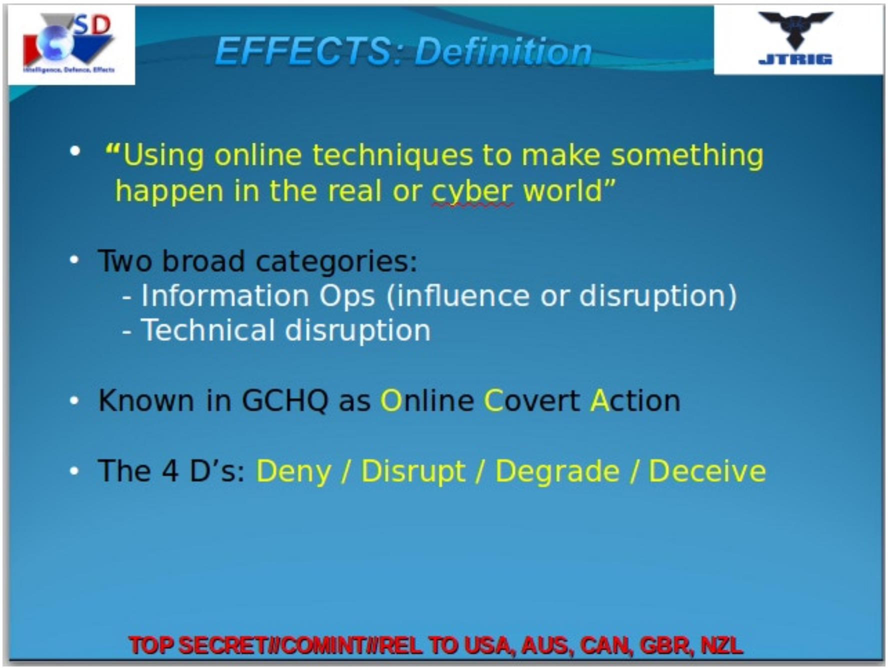

SECRET//SI//REL TO USA, FVEY
Map of technologies to message delivery
t
orever
Soon
Now
eMail
Web Pages
Blogging
LinkedIn
News Media
SMS
Facebook
Twitter
IM / IRC
Phone Voice
Personal
Community of Interest
Global
SECRET//SI//REL TO USA, FVEY

Figure 4.6: JTRIG: Map of technologies to message delivery
Courtesy of Glenn Greenwald [Gre14a].

were cleared of wrongdoing because there was no material difference between Operation Whistle Pig and the day-to-day activities of any CBP agent. The internal CBP resolution of this particular case is stunning: there basically were no rules for the agents to follow, therefore they did not break any rules, and under the principle of *nulla poena sine lege* [6] should not be punished. Consequently their activity was considered legal and ultimately constitutional by CBP. This result, while internally consistent, seems to indicate a serious structural failing in the United States regarding accountability for surveillance abuses.

---

[6] Latin meaning *No penalty without law* [Wik21l]



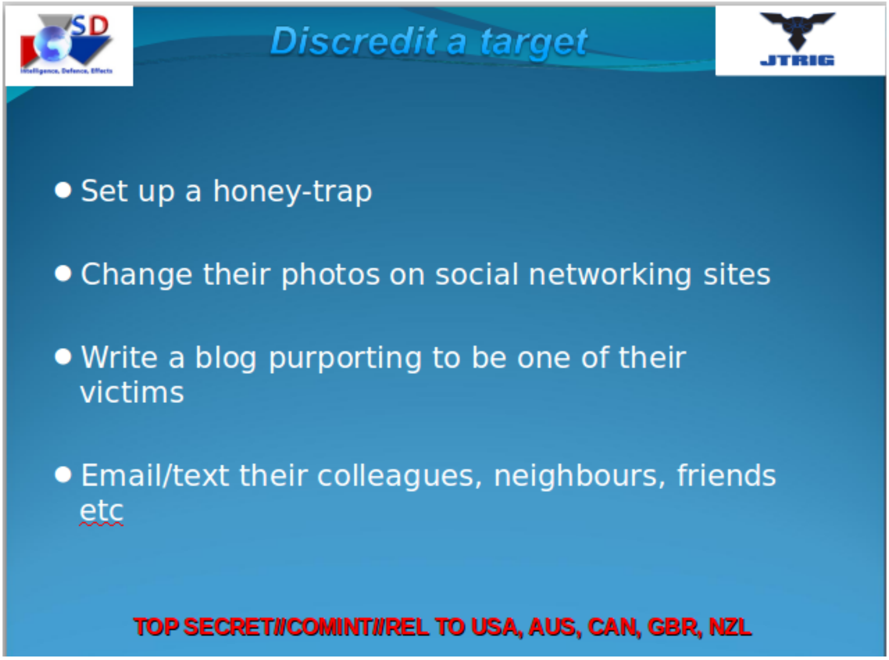
EFFECTS: Definition

- "Using online techniques to make something happen in the real or cyber world"
- Two broad categories:
  - Information Ops (influence or disruption)
  - Technical disruption
- Known in GCHQ as Online Covert Action
- The 4 D's: Deny / Disrupt / Degrade / Deceive

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

Figure 4.7: JTRIG: EFFECTS: Definition
Courtesy of Glenn Greenwald [Gre14a].

The history of explicit psychological operations is fraught with peril when considered from the human rights domain. Damningly, the UK and the United States appear to continue these practices to this day [Gre14a]. Do others also do this? Without a doubt – from Cambridge Analytica [The19] to Russian troll farms (Russian IRA) – but none realistically compares to the investment, the capabilities, and the supposed moral *values* of the UK and the USA. Psychological operations with *negative human outcomes for the targeted* are an intentional policy goal of the GCHQ JTRIG unit by their own document's admission, as seen in Figure 4.5, Figure 4.4, Figure 4.7, Figure 4.8, and Figure 4.9. Remarkably, this is considered acceptable behavior by UK authorities and those who partner with the GCHQ. Rather than condemnation, they offer casual dismissals, claiming that everything that these agencies do is legal or in the best interests of their respective countries. Semi-serious commentary [Mic14] such as "spies are gonna spy" has become a catch phrase for apologists and civil-liberty nihilists. Every country has intelligence capabilities, and indeed, spies will spy; what matters is the basis on which they operate, their economic and political limits, their legal limits, and of course, the practical capabilities of their equipment. In the case of domestic surveillance of the American population, it is clearly the case that many people, including United States federal judges, consider this kind of spying to be carried out on an invalid basis, a rejection of the basic constitutional values that grant intelligence services their authority in the first place. It follows that those without representation in American elections, such as domestic civilian surveillance targets in Europe, will find the basis to be even less valid than those who merely feel that their own government is not accountable to them.



Discredit a target

- Set up a honey-trap
- Change their photos on social networking sites
- Write a blog purporting to be one of their victims
- Email/text their colleagues, neighbours, friends etc

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

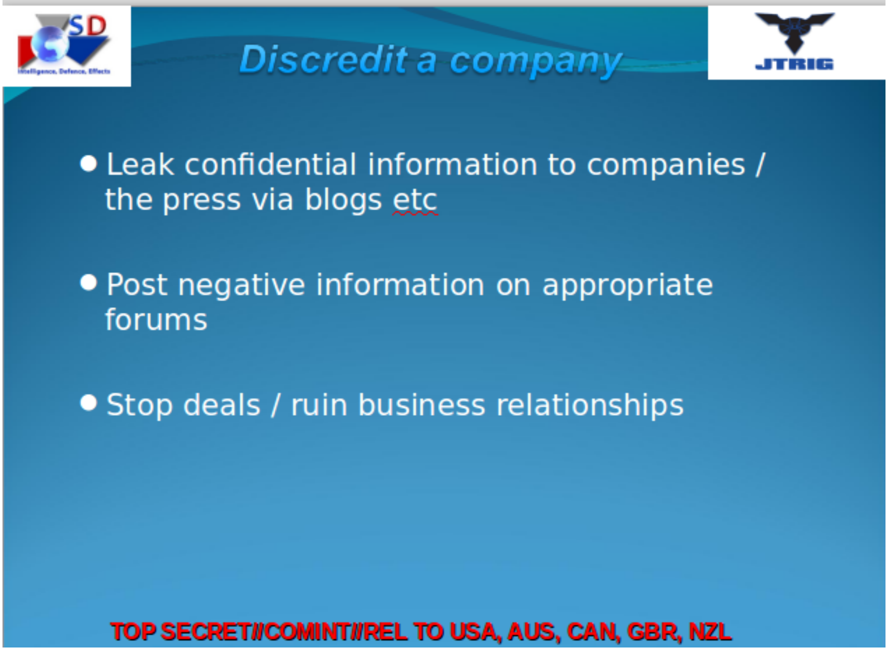
Figure 4.8: JTRIG: Discredit a target
Courtesy of Glenn Greenwald [Gre14a].

Chilling effects of such surveillance systems are plainly obvious and well documented [Pen16]. The same is true for *active measures* [Tay88] [7] such as those used by JTRIG or the famous East German Stasi. These two groups could not be more different in some ways, and it is exactly for this reason that their commonalities are so worrisome. What we know about mass surveillance today is largely the result of non-academic discussions led by whistleblowers, artists, journalists, and other independent researchers. Julian Assange [Ass07] regards it as an open secret that academics who are funded by government agencies may actively avoid critical topics that could put their funding at risk. It is interesting to note that a very *small* number of well-known professors have made their name not only by their research but also by taking a stance on the issue at all. The academics, scientists, and engineers [8] who built most of the surveillance systems discussed in this thesis are not known publicly for their work.

---

[7] Active measures include covert infiltration of groups by undercover law enforcement and/or intelligence agents. The agents use covert, often illegal actions to socially, economically, or physically disrupt political groups.

[8] By chance the thesis author had the opportunity to meet a founder of the surveillance company Narus. Narus mass surveillance and analysis systems were deployed by the NSA inside AT&T facilities to intercept all traffic flowing through their large capacity network cables as documented [KB09] by whistleblower Mark Klein. The founder, as a person who helped design, build, deploy, and operate the Narus equipment for the NSA to spy on the domestic American population, had a simple message for the thesis author: "I'm sorry." Narus is now owned by the American corporation Boeing.



Discredit a company

- Leak confidential information to companies / the press via blogs etc
- Post negative information on appropriate forums
- Stop deals / ruin business relationships

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

SD
Intelligence, Defence, Effects
JTRIG

Figure 4.9: JTRIG: Discredit a company
Courtesy of Glenn Greenwald [Gre14a].

4.2 — Foundational events and disclosures in surveillance

The early history [Seg14,Fit20] of telegraph cable tapping and initial efforts to gather large datasets foreshadow similar situations that have played out until the present day. This section surveys the events of recent surveillance history that have shaped our current understanding of adversary capability and motivation. Initially secret sources and methods inevitably proved vulnerable to public disclosure through insider leaks, whistleblowing, espionage, hacking, political direct action, legislation, journalism, FOIA requests, institutional bungling, war, or academic research. The resulting trove of information serves as the basis for countermeasures contemplated in this thesis.

1918-1945: Abuse of census data. Beyond the Nazi use [Mil97] of census data to round up supposedly undesirable citizens and non-citizens, as described in Section 1.2, Dutch and French census data were equally abused by the Nazis. Other European nations suffered similar outcomes nearly everywhere the Nazis went with their military. The original intentions of data collection in the Netherlands and in France were not meant to benefit the Nazis in their genocidal march across Europe [9]. Sadly, and predictably, the Nazi force occupying the Netherlands and France used whatever data they found as they pleased. Edwin Black extensively discusses some of the first so-called *Big Data* projects of

---

[9] The thesis author has had the opportunity to speak with members of Dutch law enforcement and intelligence about this topic on several occasions. One officer maintained that gathering data on the Jewish population was not wrong per se, that the collectors of the data had done nothing wrong, and that no one can stop such an invading force from misusing the data. When the thesis author inquired if he collected the same kinds of data about Muslims today, he confirmed that they did.

Europe in his incredible book *IBM and the Holocaust: The Strategic Alliance Between Nazi Germany and America's Most Powerful Corporation* [Bla12]. The cover of his book perfectly illustrates the *comfortable* relationship between IBM [10] and the Nazi regime as the Nazi swastika is clearly printed on the punch cards as a kind of branding customization. These are the same punch cards that were used by the Nazis to assist with their execution of the Holocaust. These crimes by the Nazi regime regularly enabled by IBM defy comparison. IBM has failed to make meaningful amends for their conscious *material support* of the Nazis regime and indeed appears to have run a strong public relations effort against Black's award winning book. The nearly limitless crimes by the Nazis should serve as a very expensive lesson to everyone about the real suffered human costs and about the differences between legality, intentionality, and capability. An invading force such as the Nazi regime cared primarily about the latter, and used the capabilities to further their own agendas. It is also a lesson to would-be corporate partners that in the long term, the politics of the day will be subsumed by the historical impact of collaboration [11] and its human costs.

Mid-20th century Europeans were not alone in their abuse of census data. We note that during the Second World War, the United States government used concentration camps [How09] to detain its own citizens, as well as non-citizens. This internment project is a pre-digital example of the American government using census data and other data for political repression and racial persecution. American authorities used census data [Pix09] to identify undesirable people on any basis, including race [Roo42]. The United States government dispossessed entire communities [Spi86] of ethnically Japanese immigrants, their children, and their families through marriage.

The immense amount of data being collected about people by the NSA is illustrated in Figure 4.10.

1940: Project SHAMROCK (disclosed 1976). Project SHAMROCK [Owe12] was a surveillance program that was originally a project of the Armed Forces Security Agency (AFSA) and then later an NSA program. The primary target was telegraph communications data delivered on microfilm. The program lasted from 1940 until 1975. Its data was sourced from the RCA Corporation (RCA), Western Union, and the International Telephone & Telegraph (ITT). The data included domestic and foreign communication, and was performed without legal authority. No accountability by those acting without authority is a common theme.

1947, 1952: Founding of CIA and NSA (disclosed 1975). After the Second World War, the CIA was founded on September 18, 1947 and the NSA was founded on November 4, 1952. They were not the first intelligence agencies of the United States, and yet they are the most contemporaneously relevant. They were both highly secretive, and extremely powerful political agencies which naturally absorbed former soldiers and spies from the Second World War. They shaped domestic and international political situations from the very beginning. Entire histories have been written about both agencies, their work together, and their work separately. Both agencies cooperate with domestic and

---

[10] Henry Ford of Ford Motor company was awarded the Grand Cross of the German Eagle in 1938 by Karl Kapp and Fritz Heller of the German government in Detroit, Michigan. Thomas J. Watson of IBM was awarded the merit of the German Eagle in 1937 by Adolf Hitler himself in Germany.

[11] The expression "*Quisling!*" is used as an insult by students of history in Europe because of the negative example set by Vidkun Quisling, the Former Prime Minister of Norway during the Nazi occupation.

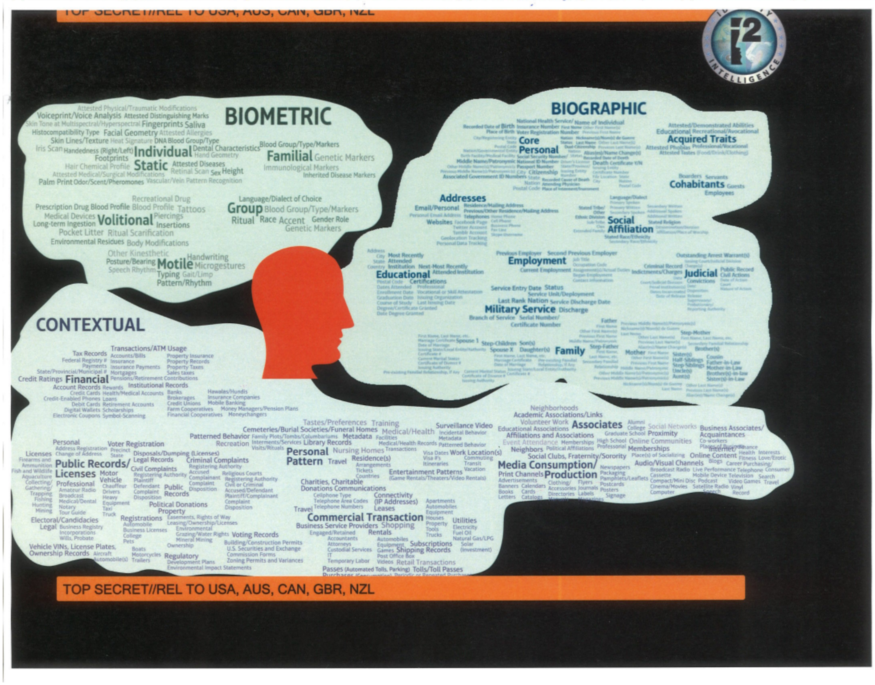

TOP SECRET//REL TO USA, AUS, CAN, GBR, NZL
i2
INTELLIGENCE
BIOMETRIC
Individual
Familial
Static
Group
Volitional
Motile
BIOGRAPHIC
Core
Personal
Acquired Traits
Addresses
Social Affiliation
Cohabitants
Educational
Employment
Judicial
Military Service
Family
CONTEXTUAL
Financial
Public Records/
Licenses
Legal
Personal
Pattern
Associates
Media Consumption/
Production
Commercial Transaction
TOP SECRET//REL TO USA, AUS, CAN, GBR, NZL

Figure 4.10: Identity intelligence
Courtesy of New York Times [RP14].

foreign agencies, and so their capabilities are shared, and extend to agencies that would not normally seem to be related even in mission statements. The CIA precursor was the Office of Strategic Services (OSS), while the NSA precursor was a mix of organizations including the US Army Military Intelligence's cryptographic section (MI-8), and the Black Chamber in 1919 to 1929, Signal Intelligence Service (SIS) during the Second world war, and the Armed Forces Security Agency (AFSA) after the Second World War. The FBI as an organization is older than both the CIA and the NSA having been established in 1908 under the name Bureau of Investigation (BOI). In 1935, the BOI changed name to the Federal Bureau of Investigation (FBI). The history of the Russian FSB is generally known as the KGB and their related allies such as the East German Stasi. Each of these agencies appears to be subject to Pournelle's Iron Law of Bureaucracy [Pou10] with a slight flair for local characteristics.

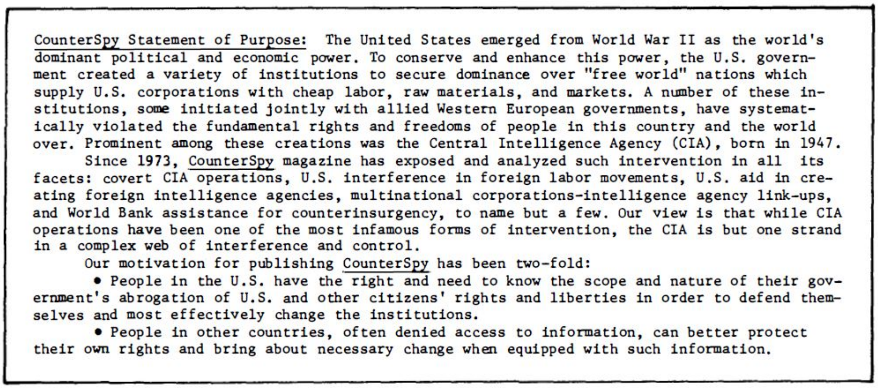
1956: Project COINTELPRO (disclosed 1976). This was also the era of COINTELPRO [Dav92], a period of time where the United States Federal Bureau of Investigation was targeting American citizens domestically for active measures. The exposure of COINTELPRO by a small group of individuals known as the Citizens' Commission to Investigate the FBI [Med14] sparked two major reviews by the US House and Senate: the Pike Committee [KB77, SC81] and the Church Committee [Com75, SC81], respectively. Both committees faced major resistance from the FBI, the CIA, the NSA, and even the White House. The Pike Committee made one important observation about the CIA which likely applied to all agencies: the CIA was often **following the orders of the President of the United States** even when the President understood that the CIA objected to a course of action for legal or moral reasons, they followed orders. Sweeping overhauls were made which included creating a special new kind of court: a secret, non-adversarial court for review of surveillance operations against American citizens or US persons. US persons has a definition in law that generally is understood to mean any American citizen, or a visitor physically on US soil for non-diplomatic reasons. Critically the Church committee and the Pike Report, discussed how surveillance was used to aid in crushing dissent, including, but not limited to, political assassination.

1962: Project MINARET (disclosed 1976). Project MINARET [Owe12] was considered as a sister program to SHAMROCK, and was originally known under other names from 1962 until 1969; the program continued until at least 1978 under various code names. MINARET was an NSA program to spy on the domestic population of the United States for political reasons. US citizens were put on watch lists and *tasked* for surveillance. The Government Communications Headquarters (GCHQ [Cam82]) of the United Kingdom contributed data on targeted American activists. The program was disclosed by the Church Committee, with US Senator Frank Church also discovered as a target.

1967: David Kahn's The Codebreakers. The definitive book on codebreakers was written by Kahn [Kah96]. The NSA wanted to suppress his original work and subjected it to a ban according to Kahn [Kah02]:

> "***Because when my book, The Codebreakers, was published in 1967, just 35 years and one month ago, it became the subject of a ban on the part of the National Security Agency. A notice was circulated here at Fort Meade and was sent to all NSA outposts worldwide. The book was never to be mentioned. It was never to be acknowledged when the media – or anybody else – asked about it, as at cocktail parties. Its author was anathema at the NSA. He revealed that America was breaking codes! Hated less only than Martin and Mitchell. And now here he is, speaking at its 50th anniversary. I sometimes feel as if I should hold up that notice the way Harry Truman, after he won in 1948, triumphantly held up that Chicago Tribune with a banner headline shouting: Dewey Beats Truman. Well, Kahn beat NSA.***"

It is not unreasonable to wonder what must or could not be shared, and through omission what impact this has on any written history. Kahn's book is a remarkable history that helps understand the internal and acceptable view on codebreaking and surveillance by the NSA itself. If it had been wholly objectionable it could have been censored on national security grounds under a secrecy order [oGOSoGIR81, pages 416-418]. Early surveillance issues were primarily centered around a small number of cables and a small number of government and commercial physical locations. Western Union illegally shared the contents of telegraph cables in bulk with the Black Chamber (MI-8), a United States Army military intelligence precursor to the NSA during the 1920s. It was understood that spying, especially during peacetime, was completely illegal and it was deeply shocking to the public when revealed by Herbert O. Yardley in 1931 [Yar31].



CounterSpy Statement of Purpose: The United States emerged from World War II as the world's dominant political and economic power. To conserve and enhance this power, the U.S. government created a variety of institutions to secure dominance over "free world" nations which supply U.S. corporations with cheap labor, raw materials, and markets. A number of these institutions, some initiated jointly with allied Western European governments, have systematically violated the fundamental rights and freedoms of people in this country and the world over. Prominent among these creations was the Central Intelligence Agency (CIA), born in 1947.

Since 1973, CounterSpy magazine has exposed and analyzed such intervention in all its facets: covert CIA operations, U.S. interference in foreign labor movements, U.S. aid in creating foreign intelligence agencies, multinational corporations-intelligence agency link-ups, and World Bank assistance for counterinsurgency, to name but a few. Our view is that while CIA operations have been one of the most infamous forms of intervention, the CIA is but one strand in a complex web of interference and control.

Our motivation for publishing CounterSpy has been two-fold:

- People in the U.S. have the right and need to know the scope and nature of their government's abrogation of U.S. and other citizens' rights and liberties in order to defend themselves and most effectively change the institutions.
- People in other countries, often denied access to information, can better protect their own rights and bring about necessary change when equipped with such information.

Figure 4.11: CounterSpy magazine statement of motivation
Courtesy of altgov2.org [Kic].

1960s-?: Project ECHELON (disclosed 1971). In 1971, Ramparts magazine published an interview [Hor72] with an NSA insider named Winslow Peck. This interview marks the start of serious uncontrolled reporting on NSA's worldwide activities. This interview revealed the existence of the NSA's internals and mission to the public at large, as well as the scale of their funding in a critical manner. More importantly it also revealed a global spying network that matched a well understood political alignment of major English speaking post-war powers. Critically this article exposed the fundamentals of what would later be known as the ECHELON mass surveillance spying network run by the United States. The other members of the *Five Eyes* (FVEY) were revealed to be: The United Kingdom, Australia, New Zealand, and Canada.

Later, it was discovered that Winslow Peck was the pseudonym of the first public NSA whistleblower: Perry Fellwock. CounterSpy magazine [Com73] was started by Fellwock and carried on for the rest of the 1970s. Printed issues of CounterSpy are now collector's items, with some early issues published online [Kic], for their extremely detailed analysis, including names of involved parties. The motivation for publishing CounterSpy is shown in an excerpt in Figure 4.11.

1978: Establishment of the FISA court. After the Church Committee, the introduction of a new type of court under the Foreign Intelligence Surveillance Act (FISA) was supposed to end domestic *unilateral* surveillance by the NSA, and their partners. In theory, after the Church Committee, this special, non-adversarial secret surveillance court was created by the FISA legislative changes to ensure American's rights would be protected. The Foreign Intelligence Surveillance Court (FISC) is largely considered to rubber stamp [12] requests from the FBI. The FBI has routinely misled the FISC, and from the little that is known, the FISC has neither the technical knowledge, nor the general temperament to actually act as a safeguard. The FISA court has signed off on general warrants that acted as a dragnet for the domestic American population's Internet traffic. The secret FISA court was created to stop this exact kind of blanket general warrant, from violating the constitutional liberties of the American people. The FISA court's absolute, abysmal failure to protect regular American people was a key motivation for the whistleblowing activities of Snowden. One of the early Snowden documents published by Greenwald was about Verizon's dragnet surveillance partnership [Gre13b] with the NSA and the FISA court's approval that was signed by Judge Roger Vinson. His court order [Gre13a] specifically makes a distinction between foreign and domestic communications records, and approves collecting it all:

---

[12] Among insiders familiar with the FISC, they joke behind the judges backs: rubber stamps do not appreciate the comparison.

> "***IT IS HEREBY ORDERED that, the Custodian of Records shall produce to the National Security Agency (NSA) upon service of this Order, and continue production on an ongoing daily basis thereafter for the duration of this Order, unless otherwise ordered by the Court, an electronic copy of the following tangible things: all call detail records or "telephony metadata" created by Verizon for communications (i) between the United States and abroad; or (ii) wholly within the United States, including local telephone calls.***"

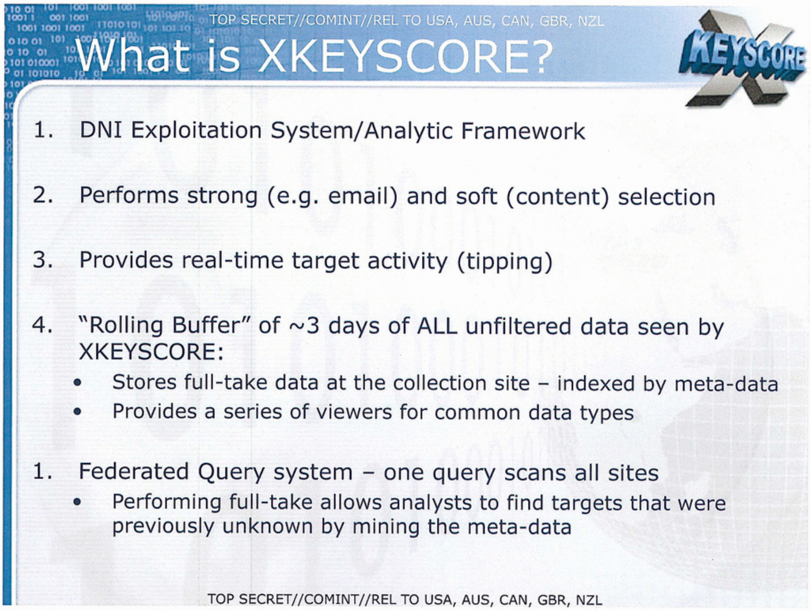
This FISC order is outrageous when we consider the original purposes of the FISA law. It is additionally quite perplexing on a number of levels, and ironically did not ask Verizon to produce data about *purely foreign telephony metadata*. The Church and Pike Committees appear to have not gone far enough and did not actually succeed in protecting the domestic American population from the collect-it-all intelligence community. A secret court without an adversarial process is not an effective safeguard of individual liberties. When we discuss ways to defeat surveillance, especially unlawful surveillance, we need to reflect not only on what may work, but what absolutely did not work. The FISC seems to be an example of both a policy and a judicial failure. The FISC relies on the FBI to be honest, and it is known that the FBI has submitted incorrect information including about people that the FBI considers to be their political enemies. The FBI has strong incentives to be wittingly dishonest and indeed appears to have been dishonest on many occasions with the FISC. Absent an adversarial process, it is unclear how the process itself could even uncover such subterfuge from the FBI. Indeed it was an Inspector General who discovered the FBI malfeasance. Meanwhile the FISC continues to toil away in obscurity, protecting the rights of absolutely no one, secretly. Regular courts are not faring any better with contemporary cases such as Wikimedia vs. NSA [GC16] and Jewel v. NSA [Bau21]. In recent cases where a person or group or company has had standing to challenge the US government's domestic surveillance practices the US government routinely claims State secrets privilege. Usually this claim is accompanied by an in-camera review by the presiding Judge who usually sides with the US government. The law alone is clearly and obviously not sufficient to protect people from mass surveillance. Consider that some prominent members of the Judicial branch are completely technically inept, and then consider that judges often rely on trusting the US government's statements absent their own independent technical understanding. The CIA meanwhile, operates their own surveillance capabilities including capabilities that are entirely outside of the purview of the FISC, even now [U.S22].

1971-1973: Daniel Ellsberg and the Pentagon Papers. During the 1970s Daniel Ellsberg became a household name for his epic folk hero actions. He is known primarily in public for the Ellsberg Paradox [Ell61], and for whistleblowing about the American war in Vietnam which is now commonly known as the Pentagon Papers. For his courageous action to expose serious fraud, waste, and abuse by various responsible parties in the US Government, his psychologist's office was burglarized and he was wiretapped. Then US President Nixon ordered Ellsberg to be "**permanently incapacitated**" by a CIA team on the steps of the Capitol building. The chilling effects of these blatantly illegal actions were so strong that the presiding court dismissed the legal charges against Ellsberg. He walked free, having gone underground for a short time to stay alive. Ellsberg later said [Ell72] of the American war in Vietnam that it "*needs not only be resisted; it remains to be understood.*"; this observation perfectly applies to surveillance, and especially the technical aspects of mass surveillance.

1970s-1980s: Duncan Campbell's GCHQ reporting. British investigative journalist Duncan Campbell has remarked in public that the 1972 Ramparts interview [Hor72] with Winslow Peck was a driving force behind his investigations into the GCHQ. Winslow Peck visited Campbell in London in 1976 which was the serious start of investigations into the GCHQ. In the 1970s and 1980s, Campbell and his coauthors worked tirelessly [Cam79, Cam82, Cam98] to inform the British public about intelligence activities carried out by their state. This line of reporting was even more dangerous than contemporary times and unlike the United States, the United Kingdom has an Official Secrets Act without the protections of the First Amendment in the Bill of Rights. Within the first year of reporting on GCHQ, Campbell was arrested, jailed, and was facing around thirty years in prison for revealing the existence of the (GCHQ). Campbell's coauthors were also treated harshly; Mark Hosenball, an American journalist was deported and banned from the United Kingdom as an alleged threat to national security. The UK government could not make the charges stick due to the open nature of the facts gathered that were the basis of their reporting. Their work opened a new era of investigative journalism into the world of electronic intelligence (ELINT), radar intelligence (RADINT), communications intelligence (COMINT) and signals intelligence (SIGINT). Campbell has remarked that the original Time Out story, The Eavesdroppers [Cam76] was for the UK what the Rampart's interview with Winslow Peck was for the United States of America. Their reporting took electronic and signals intelligence work out of the shadows. They wrote about the technical details of various programs. They revealed locations, and activities, as well as naming involved parties. The level of commitment required for such reporting is extremely high: Campbell faced serious suppression including a period of time incarcerated, as did his coauthors.

1982-present: James Bamford's NSA reporting. Following the work of Campbell, American investigative journalist James Bamford wrote a series of books [Bam82,Bam02, Bam05, Bam09] critically examining the NSA, and he continues to regularly release new books. Similarly to Kahn's Codebreakers, some of what Bamford writes comes from Freedom Of Information Act requests (FOIA) where the NSA carefully reviews, censors, blocks, and/or releases information to the requester. NSA has long FOIA time delays and it sometimes seems purposeful as a part of their general strategy for controlling information. FOIA documents from the NSA represent what the NSA is willing to acknowledge in public, in a sense, it is the opposite of Assange's observation about how internal documents intended for other insiders tell the real story. Bamford's works, like Kahn, are remarkable; often they are based on other internal sourcing such as interviews and not only documents approved by the NSA, and his work should not be diminished because of the

NSA's trickery around FOIA requests. His work documents the foundation of the NSA organization in the post-war era, and covers important contemporary activity of the agency as well.



TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

What is XKEYSCORE?

1. DNI Exploitation System/Analytic Framework
2. Performs strong (e.g. email) and soft (content) selection
3. Provides real-time target activity (tipping)
4. "Rolling Buffer" of ~3 days of ALL unfiltered data seen by XKEYSCORE:
    - Stores full-take data at the collection site – indexed by meta-data
    - Provides a series of viewers for common data types

1. Federated Query system – one query scans all sites
    - Performing full-take allows analysts to find targets that were previously unknown by mining the meta-data

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

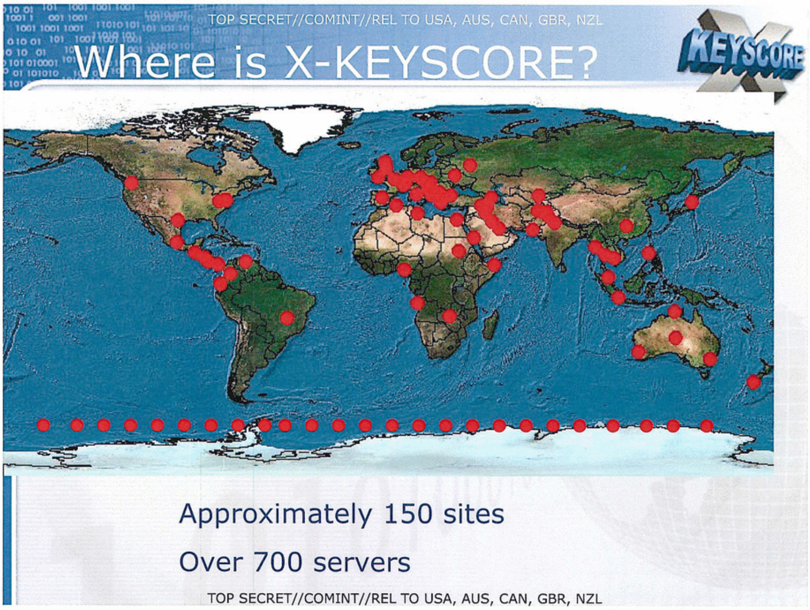
Figure 4.12: What is XKEYSCORE?
Courtesy of Glenn Greenwald [Unk13].

Consider as an example the conclusions that would be reasonably reached if we considered *only FOIA documents* [Cim21] in isolation as a way to learn about capabilities such as those found by Cimpanu. In his article we see the FBI's claim presented as "*(U//FOUO)* [13] *FBI's Ability to Legally Access Secure Messaging Content and Metadata*". The FBI document covers what content can be recovered ***legally*** from popular messaging software and explicitly names iMessage, Line, Signal, Telegram, Threema, Viber, WeChat, WhatsApp, and Wickr.

What we know from the Snowden archive is that the FBI's ability to *extra-legally* access metadata and content is far broader than the so-called legal access shown in the [Cim21] FOIA document. It is also noteworthy that not all messengers are equal in terms of the data provided under a legal pretense, yet they ***all appear to cooperate with the FBI in some way*** to disclose data to the FBI according to the FBI document. Some of the data returned would clearly be of assistance to hack a target's device, and any technical details of active hacking would be kept secret as part of the FOIA process. In the Snowden archive, we see lots of hacking and hacking related programs run by NSA such as the TURBULENCE [Wik21u] program which is made up of modular sub programs [Amb13]. Those programs include TURMOIL [Gal14b], TUTELAGE [AGG$^+$15a], TURBINE [GG14, Wik20d], TRAFFICTHIEF [Wik20c], and XKeyscore [Gre13d, Unk13, AGG$^+$14b, Unk15a] as shown in Figure 4.12 and Figure 4.13, as well as data that was pilfered during those break-ins. We discuss these programs, related programs, and furthermore present XKeyscore surveillance engine source code in Listing 4.1. While it is obviously useful to have FOIA documents, considered alone they may be extremely misleading, it may be that some documents are even released exactly to mislead the public.

---

[13] "*Unclassified//For Official Use Only*"



TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

Where is X-KEYSCORE?

Approximately 150 sites

Over 700 servers

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

Figure 4.13: Where is X-KEYSCORE
Courtesy of Glenn Greenwald [Unk13].

1992-2003: First Crypto War. The 1990s saw many discussions about surveillance and the place of cryptography in society. Early [Gol18] Internet Freedom policies were developed that would continue for the next two decades, playing a much more complicated role in the 2010s than previously understood [Mac21,Ope13]. This period included a number of pivotal historical events: the publication of PGP by Philip R. Zimmermann, the founding of the Electronic Frontier Foundation (EFF) by John Gilmore, John Perry Barlow, Mitch Kapor, the cypherpunks mailing list, the Clipper Chip introduced by the Clinton administration and subsequently broken by Matt Blaze, and perhaps most importantly the Bernstein v. United States legal case [Ber03] carried on in the background from 1992 until 2003. This period is sometimes referenced as the First Crypto War [Lan18a] and is the subject of the well regarded Steven Levy book Crypto [Lev01]. Levy largely concluded that the crypto rebels won, though it seems in hindsight and twenty years after his book that much of the total struggle was never done in public. Many related legal challenges have largely been at a stalemate, and technological countermeasures have been regularly clandestinely sabotaged. If one considers theories of change, we note that there are many competing theories from policy changes to technological development to stop spying. By combining the technical and the legal, it seems that Bernstein expanded people's practical legal rights. His win was a strategic win for everyone with lasting results. Much of the technical success of this era cannot be said to have the same lasting impact without also attributing some of the credit to Bernstein's win.

2000: Duncan Campbell's report to the European Parliament. At the end of the 20th century, Campbell was commissioned by the European Parliament [14] to produce a report about interception capabilities. He decided to expose how intelligence activities were interrelated, even among States. His report, Interception Capabilities 2000 [Cam99] was extremely detailed. He continues to report on similar issues [BCRT15, MA21]. His report to the European Parliament exposed the physical infrastructure of mass surveillance, and tied it together with programs such as ECHELON [Cam00]. Previous to this report, much of what was known about ECHELON was nearly unverifiable rumors beyond the original Rampart's interview with Winslow Peck. Asking about the topic of ECHELON in polite company that did not specialize in journalistic research on the topic of surveillance was on a par with discussing UFOs or UAPs [And21b, Loe] over dinner with strangers.

Campbell's report is the context in which we should consider the capabilities before the events of September 11th, 2001. While the Second Crypto Wars [Lan18a] are generally understood to start in roughly the year 2000, implementation did not start in earnest until 2001. It was only a few days after September 11th, 2001, that the United States started upgrading their collection and processing facilities, beginning an era of unprecedented, extralegal activity that in practice [Thi18] upended much of the post-war era international law about torture, assassination, and of course, surveillance. Through the last twenty years of this activity, the result is that what was once illegal became regular secret practice, until it was normalized [Koh13], made legal, and made permanent. US President George Bush famously used the legal arguments of John Yoo [Sch09] at UC Berkeley to support his policy goals, and Barack Obama similarly relied on Harold Koh [Koh13] at Harvard for the same reasons. It is not clear that US President Trump even bothered with a legal fig leaf. In all three cases, completely immoral, unconstitutional, and illegal activities such as torture, mass surveillance, and assassination have been normalized as acceptable activities of the government agencies of the United States of America.

While it is widely understood that NSA and other intelligence agencies like the CIA have performed significant domestic surveillance operations, the era after 2001 was a turning point both nationally and internationally. This was the start of the *collect-it-all era* [Cra15], though the public was not yet aware and those who spoke of this as a confirmed reality were often ridiculed. While political assassinations had been understood to be banned since the Church and Pike Committee era, the Iran-Contra scandal [Koh90] of the 1980s was an example of the size of the loopholes possible for misbehavior. Discussion of this kind of behavior was dismissed out of hand by those who assumed the political reforms led by the Church and Pike committees had somehow been successful.

---

[14] Produced according to Campbell for the "*Director General for Research of the European Parliament (Scientific and Technical Options Assessment programme office)*"

2001: Attacks and aftermath. Late 2001 was a turning point for surveillance, as it was a turning point for government interference in the lives of many ordinary Americans and many people all over the world regardless of nationality because of their religious faith and their religious expression of that faith. A huge effort was made to infiltrate religious communities of interest, and some of those who refused to become assets or informants left the United States. Accurate numbers of people who became informants and/or assets are still generally unknown. There was a quaint, almost overtly naive notion that to leave the United States would somehow stop law enforcement and intelligence related harassment, when in fact it only changed the severity of what was allowed. Rumored for years before it was confirmed in public, the United States government began to assassinate people with drones outside of the usual, expected, geographical boundaries of the war zones where wars were being openly fought. This included targeting American-born American citizens for assassination without trial as a result of their political and religious speech. The US government made claims about terrorism but absent a trial, these claims are simply not credible. Their standards for such assassinations are similarly non-credible. The new standard for this kind of assassination policy by the US government is now called the Disposition Matrix [Wik21g], an obviously unconstitutional process with unjust results. In a spectacular case [MH11,Ram11,Sca13b,SG14], a famous American in self-imposed exile, Anwar al-Awlaki, was targeted for assassination and anyone in his vicinity was considered acceptable collateral damage. Several attempts over time failed and in each attempt innocent bystanders were reportedly killed. Eventually the prolific American YouTube star was killed by a Hell-Fire missile fired by his own government. Subsequently his two American children were killed. His sixteen-year-old son Abdulrahman Anwar al-Awlaki was killed by American drone strike, and his eight-year-old daughter Nawar al-Awlaki was shot in the neck by SEAL Team 6 and left to bleed to death in January 2017 [Gre17]. All three victims were American citizens born in the United States. One of the children was only eight years old, and the other merely sixteen years old. None of them were given a trial before their remote controlled assassination by flying an armed robot drone over Yemen. In an unrelated but relevant attempt at recruitment, the Fifth Circuit Courts of the United States ruled [Cus21] that the American citizen in question cannot sue the Federal government for targeting him after he has refused to become an informant. These crimes were carried out with the help of surveillance data. The United States is a country that not only has a death penalty, it is a country that unilaterally brings that death penalty to other countries' soil. This is part of the legacy of the last twenty years, modern American assassination programs have evolved in lockstep with improvements in surveillance. Unfortunately for the innocents targeted, surveillance need not be perfect to be militarily and politically actionable. Many unquestionably innocent people have been killed [B1̈7] with near zero accountability while mistakes by the US government accumulate [KHAS21, New21] in public [15]. America has routinely prioritized legally prosecuting whistleblowers such as Daniel Hale [Dev21] or extra-legally persecuting those who have finished their legal sentences like Manning [Bar21] as they have helped to expose US government crimes to the public, rather than prosecuting the people who commit war crimes.

---

[15] Are we the baddies? With regards to mass surveillance, and assassination by drones, unquestionably. Compare the principles of America to the practices: there was a time recently when the US President did not have the legal power to have you murdered without negative legal or political consequences for themselves. That time has now gone. Should you find yourself in this situation – attempts by your family members to challenge your slated assassination will be thrown out of court for a lack of standing as happened to Anwar's

2003: Mark Klein and AT&T. In 2003, an AT&T employee named Mark Klein noticed some changes in his work facility. As a whistleblower [KB09] he exposed the complicity of AT&T. Specifically he revealed mass surveillance by the NSA at an AT&T facility in San Francisco, California. The location at 2nd and Folsom Street was even understood to be operated out of a specific room in the AT&T facility, 641A as shown in Figure 4.14. This was revealed to be a key West coast interception point for the NSA and other intelligence services including law enforcement partners such as the FBI. A variety of legal cases were brought forward by public interest law firms, and nearly all of them were quashed under the guise of the state secret privilege.

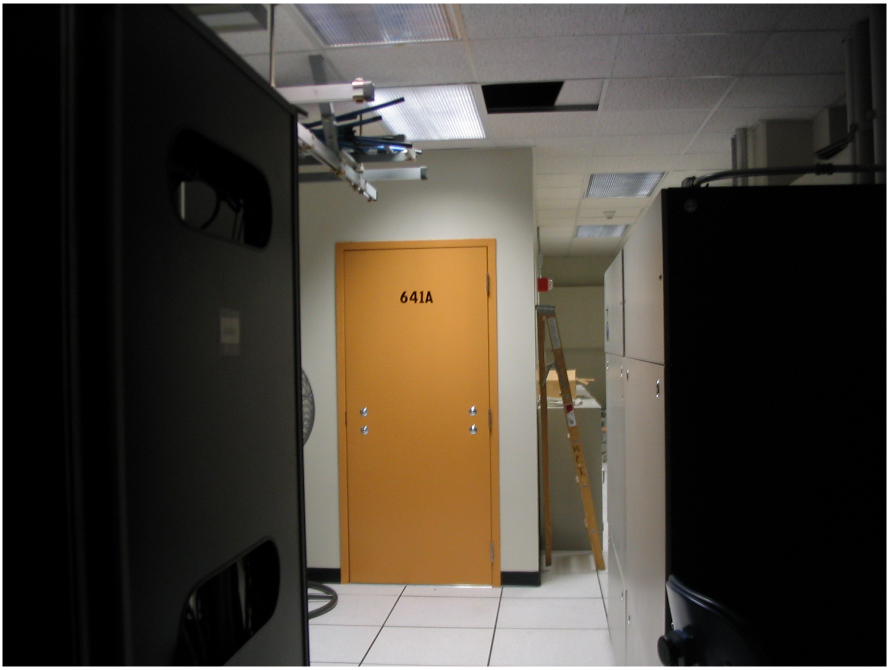


Figure 4.14: NSA domestic surveillance: AT&T room 641A
Courtesy of whistleblower Mark Klein by private correspondence.

2001-2007: Joseph Nacchio case. While AT&T cooperated willingly with NSA subversion of their customers' trust, a phone company in the American Pacific Northwest region, QUEST, did not initially agree. The CEO of QUEST, Joseph Nacchio, understood the profound legal and constitutional implications of mass surveillance, and he reportedly refused to allow NSA to deploy interception equipment. A few months later, he was charged with insider trading and eventually imprisoned. This series of

---

father in an American court shortly before his son was assassinated.

events [D'A07, Coh10, Ric18, Pet13a] almost certainly did not go unnoticed by other telecom executives or technologists who were subject to approach by various agencies, including the NSA. Interestingly, Nacchio was apparently willing to submit to the NSA surveillance if the NSA was willing to follow public law by having the FISC submit an order to him. The NSA reportedly declined to approach the FISC as they apparently did not think the FISC would agree with their activities. Nacchio was likely subject to arbitrary punishment for refusal to comply with what was clearly an illegal order at the time by the NSA, far beyond their authorities. Those such as James Clapper and General Keith Alexander who blatantly lie [Rya13] to the US Congress are generally held to a different set of rules than those who object to, or who reveal, mass surveillance. This is in stark contrast to whistleblowers who are forced into exile, where they may become fugitives, like Snowden, or who faced cruel, unusual, and degrading treatment in prison, like Manning.

2000s: A flood of whistleblowers. A continuing series of whistleblowers from various American law enforcement and intelligence agencies has stepped forward. Each told largely the same story from their respective point of view, and each of them suffered various kinds of retribution for their bravery. The first [Sil07, Mad13] of the early 21st century to speak out about mass surveillance issues was Thomas Tamm [HS16, Isi08] from the United States Department of Justice (DOJ). Thomas Tamm worked in the Office of Intelligence Policy and Review, and he chose to talk to the American news media as an anonymous source. His anonymity did not last. Shortly after Russel D. Tice [Tic06] of the NSA and the Defense Intelligence Agency (DIA) spoke to the America news media about unlawful and unconstitutional wiretapping practices by various elements of the US government. Later, Bill Binney [ARAHS19], Thomas Drake [Ell10], J. Kirk Wiebe [Sho13], and Edward Loomis [Sho13] stepped forward with various claims including serious constitutional claims about mass surveillance performed by NSA on the domestic American population. They reported their findings about wide ranging domestic spying [Wik21o], as well as concerns about fraud, waste, and abuse in the Trailblazer [Wik21t] program to Diane Roark [Wik21t], who in turn was also persecuted. John Kiriakou [Sho13] from the CIA, and as well as others, stepped forward for a variety of personal, legal, political, and/or societal reasons. One reason shared by many whistleblowers was that they rejected the use of torture by the US government. Torture had just been given the new name *enhanced interrogation techniques*, and many media outlets refused to even use the word torture to describe what was clearly internationally prohibited torture. Over time, the mass surveillance descriptions from US government whistleblowers largely matched the private sector surveillance infrastructure, such as what Mark Klein had exposed [KB09] as deployed in AT&T: indiscriminate mass surveillance of high-capacity network cables. In some cases it was claimed that the extracted data was being used, or could be used, for specific political purposes. Fiber optic or copper, whatever flowed into and through the AT&T facility went directly to the NSA and indirectly to their corporate partners, like the FBI, who consume their intelligence products. Still, there was a concerted effort by various agencies such as the DOJ, the FBI, and even the White House to conceal and to deny any mass surveillance by American authorities. Certainly there was additionally an effort to conceal the use of data pilfered from spying. After the passage of the FISA Amendments Act, and a series [Wik21h] of additional modifications to the FISA law, US government efforts to conceal turned into efforts to normalize what had just been made retroactively legal, even though major constitutional issues remain.

Was the data in fact being used? We know about the *parallel construction* policy [SC13] where the Drug Enforcement Agency (DEA) of the United States used illegally obtained data to target people for legal prosecution. Is it so hard to believe that this is happening more broadly than the few places where it has been concretely exposed? One thing that the various whistleblowers usually did not bring with them was classified information that could prove their claims beyond a doubt. The reason for the lack of physical or informational evidence is a matter of simple legal analysis: revealing classified information, even if it includes evidence of someone else more politically powerful breaking the law, is seen as a crime in itself when one has signed a non-disclosure agreement, and especially if the person carries a security clearance. Even without iron clad evidence the whistleblowers were and continue to be generally targeted with outrageous lies about their personal and professional lives, and their political views.

2009-2011: Early WikiLeaks and Chelsea Manning. A turning point in this discourse happened when the alleged source of a number of WikiLeaks publications was arrested. The history of the legal and political trials and tribulations of Chelsea Manning is now largely understood by the public, and has been amply documented in newspapers, books, and even films. However, the impact of the leaked documents is a story in itself – the documents are used in court cases, in films, and in assisting in research in many fields of endeavor. Thus while Manning has been subject to arrest, physical and mental abuse amounting to torture [16], and myriad moments of suffering, her impact is undeniable and continues to this very day. This change recalls the basis of scientific journalism [Jul06, Ass06, Bru11], as described by WikiLeaks' founder Julian Assange. His analysis of how insiders speak to each other and his claim that internal documents tell the internal truths of organizations is not entirely novel, but his application of theories of pressure to break secrecy was novel in terms of their practical application. At the time that Manning was arrested, nearly no news gathering platform had a reasonably secure cryptographic system for submitting documents, usually there was no practical method of anonymizing a submission as a source, and generally a source was at the mercy not only of networks under surveillance but of the operational security competence of journalists. A flood of documents began to flow when news producers everywhere began to launch instances of software projects such as DeadDrop, later renamed to SecureDrop on their public news gathering websites.

People in English-speaking countries first became aware of WikiLeaks' publications [Wik08] related to human rights violations in Kenya. In 2009 Amnesty International's United Kingdom office gave WikiLeaks an award for their work in Kenya. John Paul Oulo and Oscar Kamau Kingara, who worked with WikiLeaks, were assassinated [Gue09] in 2009 in the streets of Nairobi, Kenya. WikiLeaks became a household name in the United States of America for publishing Collateral Murder [Wik10d], a video of American forces in Iraq killing civilians, including two Reuters journalists. It was widely considered to be a journalistic scoop. Shortly after the publication of Collateral Murder publications of the Afghan War logs [Wik10e, Ell10], and the Iraq war logs [DSL10, Ell10] was only topped by the subsequent release of approximately 250,000 secret diplomatic cables [Wik10c, Jul15, Wik10a], minus those temporarily withheld for harm reduction

---

[16] See Manning v. Clapper (1:16-CV-02307) [Man16] for further details of her treatment in her own voice.

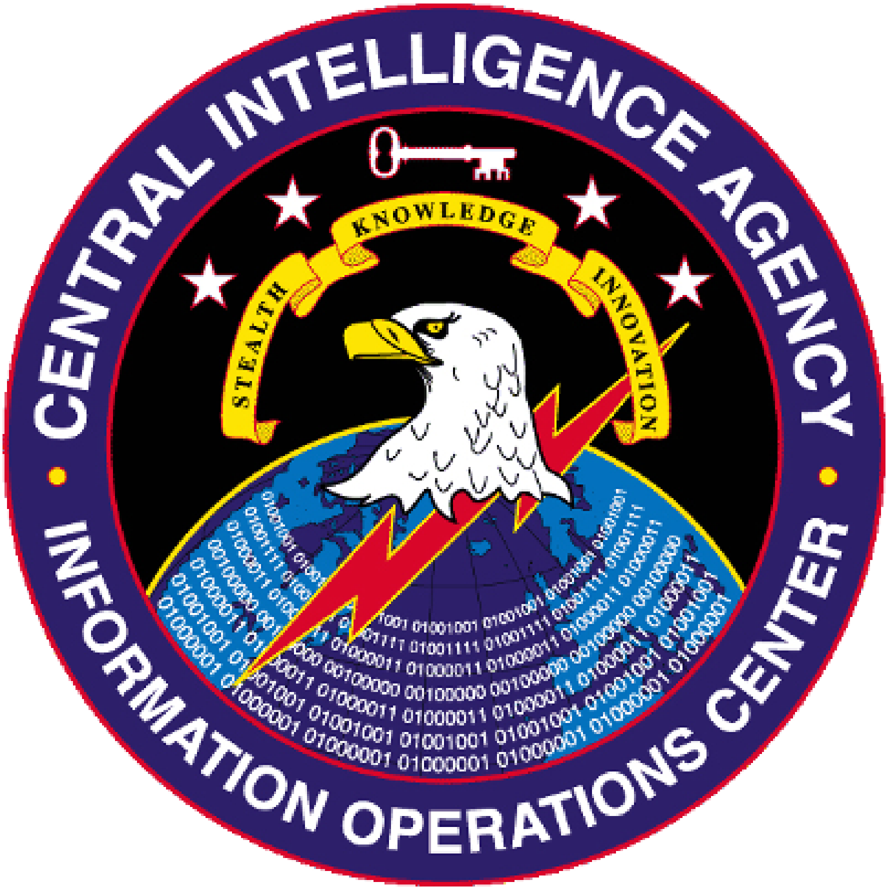


Figure 4.15: Central Intelligence Agency Information Operations Center logo
Courtesy of WikiLeaks [Wik17k].

reasons, from the US State department. It would be impossible to mention WikiLeaks, without mentioning their famous alleged source Manning, and waves of repression that followed these publications as we explain later. WikiLeaks founder, Assange was unique in the world of journalism, with a background studying mathematics and physics, and a reputation as a skilled computer hacker [DA12] in every sense of the word. Thus it is no surprise that WikiLeaks has often focused on technical details and on publishing full documents whenever possible, embodying the concept that Assange calls scientific journalism.

The WikiLeaks Spy Files publication series [Wik11a, Wik11b, Wik13, Wik14] focused on exposing the role of corporations in targeted and mass surveillance by publishing their product brochures, copies of the malware used by police and intelligence, generally contextualizing products around targeted and mass surveillance.

2017-2018: Reality Winner case. Reality Winner received a harsh five year three month prison sentence for leaking a small number of documents that were clearly in the public interest. Effectively all of these whistleblowers suffered a ban on work in a given field of endeavor by losing their jobs and their clearances, a harsh punishment that lasts for their natural lifetime. An open question remains of what role publishers have to play in source protection – the Intercept as a publication has had multiple sources arrested through what appear to be a mixture of infiltration, technical, and legal mistakes, to say nothing of their political cowardice in the aftermath. The Intercept is sometimes accused of being a kind of document-leaking honeytrap [Mag90] because of this record [17]. Former sources do not seem pleased with their encounters with the Intercept, their journalists, or the technical and non-technical security measures that failed to shield them. The publisher and the editor of the Intercept have largely avoided any serious accountability for these mistakes, no one has been fired or disciplined, and founding editors Laura Poitras and Glenn Greenwald left The Intercept for related reasons.

2016-2018: Terry Albury case. Former FBI Special Agent Terry J. Albury received four years in prison while also losing his pension, as well as clearances which might otherwise have allowed him to find work in the future. Albury's leaks [The17] are in some sense the most important for understanding the real world human impact of targeted and mass surveillance. His leaks expose how targeted and mass surveillance data eventually will be used by agencies as they see fit, regardless of the reasons for collection in the first place. In his own words reflecting on his time in the FBI, Albury summarized it as follows: "*I helped destroy people*" *[Rei21].* Targeted and mass surveillance data made that job easier.

2014: Andrew Clement paper on IP routing. Thanks largely to whistleblowers who exposed documents as proof of domestic surveillance operations, a large number of non-intelligence oriented people began to take the issue of mass surveillance seriously. One of the first academics to approach the topic with regard to network packet flows and the geography of collection was Andrew Clement at the University of Toronto. With the IXMaps [CPP10] project, Clement attempted to reveal additional interception points by analyzing the paths that IP packets take through the Internet. One of his classic examples show that a Canadian visiting the Canadian Hockey Hall of Fame website, would be routed through an American Internet Exchange Point (IXP) that was likely a point for NSA surveillance.

2007-present: Trevor Paglen. Understanding surveillance operations also requires understanding the culture of people performing this surveillance as well. Artist Trevor Paglen worked to document what he called Blank Spots on the Map [Pag09]. He has published extensively in various mediums. His research and photography of forbidden areas, or of unfamiliar objects, in a familiar sky, confront viewers. In doing so, the geography of much of the surveillance collection became objects of discussion, objects of art available for purchase in a gallery context, and it became slightly more permissible to speak about the topic. His works are nationally and internationally recognized, and with that recognition comes normalization of discussion about the topics. His books include collections of military and intelligence themed patches [Pag10] to be affixed to uniforms or other clothing. This kind of open source intelligence (OSINT) helps us to understand the social aspects of programs, the culture of the people inside the group, and sometimes even the scale of funding.

---

[17] It does not help that they have closed their Snowden archive and reportedly it has been destroyed. What can one expect from a business partnership with a billionaire? Not much in the source protection department apparently.

2651 Olive St **St Louis, MO 63103**	AT&T
420 South Grand Ave **Los Angeles, CA 90071**	AT&T
611 Folsom St **San Francisco, CA 94107**	AT&T
51 Peachtree Center Ave NE **Atlanta, GA 30303**	AT&T
10 South Canal St **Chicago, IL 60606**	AT&T
30 E Street SW **Washington, DC 20003**	Verizon
811 10th Ave **New York, NY 10019**	AT&T
12976 Hollenberg Dr **Bridgeton, MO 63044**	AT&T

Table 4.1: Contents of pamphlet handed out at the Whitney Museum in 2012.

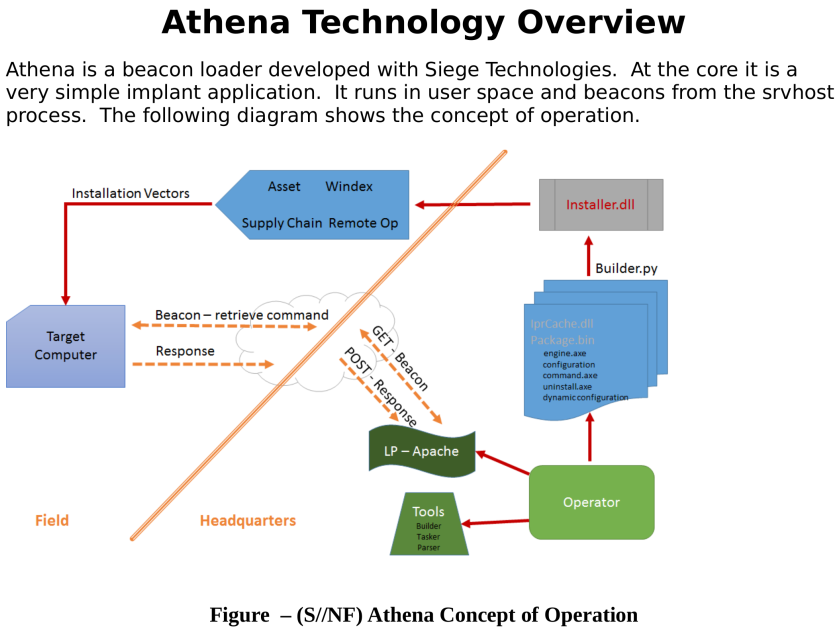
2012: Binney, Poitras, Appelbaum. In a different sort of artistic intervention, documentary film-maker Laura Poitras hosted a talk [Lya16] with Bill Binney and the author of this thesis at the Whitney Museum in 2012 in New York City. In this talk, Bill Binney stated a number of facts about domestic and international mass surveillance that were largely dismissed by the press in the audience that evening. During this talk, a masked person walked through the audience and handed out pamphlets [Pub12] that directed visitors to the addresses of suspected domestic interception locations. Many years later through analysis of the Snowden archive [GM18], several of these addresses were indeed confirmed to be domestic interception locations of the NSA. The addresses disclosed in 2012 are shown in Table 4.1.

2017: WikiLeaks Vault 7 disclosures. In the spring of 2017, WikiLeaks began publishing a series about the CIA: Vault 7 [Wik17k]. This series enumerated technical details of the CIA targeted hacking operations [Wik17k] and nearly all of the programs revealed relate to surveillance, as well as uses of surveillance data. Many of the programs revealed were developed by the CIA's Embedded Development Branch (EDB).

Highlights from the Vault 7 publications include `Dark Matter`, malware developed by EDB for targeting the firmware of Apple devices and computers [Wik17m].

An anti-forensics framework [Wik17u] called `Marble Framework` which includes techniques to frustrate and mislead reverse engineers on the origin of the software. Grasshopper [Wik17q] is used to build customized malware payloads that target the Microsoft Windows operating system.

The EBD developed `Hive` [Wik17s] for assisting in exfiltration of data stolen by CIA malware. `Weeping` Angel [Wik17z] is an implant for Samsung televisions televisions that appears to be designed by CIA and British intelligence to spy on people using their



Athena Technology Overview

Athena is a beacon loader developed with Siege Technologies. At the core it is a very simple implant application. It runs in user space and beacons from the srvhost process. The following diagram shows the concept of operation.

Installation Vectors
Asset
Windex
Supply Chain
Remote Op
Installer.dll
Builder.py
Target Computer
Beacon – retrieve command
Response
GET - Beacon
POST - Response
IprCache.dll
Package.bin
engine.axe
configuration
command.axe
uninstall.axe
dynamicconfiguration
LP – Apache
Operator
Tools
Builder
Tasker
Parser
Field
Headquarters

Figure – (S//NF) Athena Concept of Operation

Figure 4.16: Athena Technology Overview
Courtesy of WikiLeaks [Wik17g].

own television. Document watermarking is performed with `Scribbles` [Wik17x] to find leaks of documents. The `Archimedes` tool [Wik17f] attacks computer systems on the same local-area network. We explore defenses to this kind of attack tool in Chapter 7.

`AfterMidnight` [Wik17d] is a framework for creating malware that targets Microsoft Windows systems.

The `Athena` [Wik17g] project is a user space malware implant produced together with corporate contractor Siege Technologies (later acquired by Nehemiah Security) for the CIA with intended use such as supply chain interdiction [Wik21c] as shown in Figure 4.16. `Pandemic` [Wik17c] is a malware implant for Microsoft Windows that infects shard files in an attempt to spread.

`Cherry Blossom` is malware that is used to compromise routers and/or Wi-Fi access points made by DLink, Belkin, Linksys and other brands [Wik17j]. `Brutal Kangaroo` is a CIA project for a suite of tools (`Drifting Deadline`, `Shattered Assurance`, `Broken Promise`, `Shadow`) useful when attacking air-gapped networks of Microsoft Windows computers using USB and other portable storage devices [Wik17i]. `Elsa` is geolocation focused malware that targets Microsoft Windows computer systems which have Wi-Fi (802.11) hardware [Wik17o].

There are many programs that are part of the Vault 7 WikiLeaks publications that it is difficult to summarize each publication without understating the importance of the series as a whole. `OutlawCountry` [Wik17v] is a GNU/Linux kernel module that allows for covert firewall sabotage. `BothanSpy` [Wik17h] is used to steal secure shell (SSH) credentials and SSH sessions from Microsoft Windows users, and a `Gyrfalcon` is the GNU/Linux variant. `Highrise` [Wik17r] provides short message service (SMS) proxying capabilities on Android devices. Raytheon submitted ideas for future malware [Wik17y] contracting work. The `Imperial` [Wik17t] suite of projects to backdoor various systems including `Achilles`, software for implanting malware in MacOS disk image files, `Aeris` for GNU/Linux or BSD systems, and `SeaPea` which is a MacOS kernel implant. For physical intrusions, `Dumbo` [Wik17n] is a project designed to be run from a USB disk to compromise the targeted machine running Microsoft Windows. `CouchPotato` [Wik17l] is a video and still photo interception tool.

Surreptitious biometrics collection with `ExpressLane` [Wik17p] is done by compromising those who have legitimate access to biometrics.

`Angelfire` [Wik17e] is a suite of other projects ("*Solartime, Wolfcreek, Keystone (previously MagicWand), BadMFS, and the Windows Transitory File system*" [Wik17e]), primarily aimed at compromising Microsoft Windows systems. Targeting regular computers is not the only goal of the CIA; the `Protego` [Wik17w] project is a missile control system built with Raytheon that appears to allow geographic control of the weapons system.

The Vault 7 series started with a publication release named *Year Zero* [Wik17k] with thousands of documents from the CIA's Center for Cyber Intelligence (CCI) as seen in Figure 4.17. The CCI is a part of the Directorate for Digital Innovation (DDI). The CCI is physically located in Langley, Virginia and it is made up of sub-groups such as the Engineering Development Group (EDG), the Embedded Devices Branch (EDB). WikiLeaks explicitly linked the Year Zero Vault 7 publication to the CIA hacking [Wik17a] of the 2012 French presidential election. Normally attribution of adversarial hacking activity is extremely challenging, and it often takes extensive technical research or insider information to link one piece of software to another, or one program to a wider campaign. By publishing the samples of the malware from the CIA, along with the technical documents that explain how the malware functions, WikiLeaks linked wider activities with the specific tools that were used in a given operation.

WikiLeaks also published a partial analysis of the CIA organizational structure as seen in Figure 4.17, revealing sections of the CIA which are responsible for developing the Vault 7 malware or focusing on specific platforms. The technical details in the Vault 7 series stands apart from nearly every other major publisher, except Der Spiegel, who published a sample of the QWERTY [AGG+15b] malware attributed to the NSA. After publication of the NSA malware, security researchers were able to conclusively link the NSA to a series of attacks. Der Spiegel confirmed that the Regin malware is an NSA tool [RSS15]. The information about Regin has since been integrated into common security software, and thanks to the publications, we also learned not only about the intentions of the malware authors but exactly who is responsible. In response to various publications [JAS13], Der Spiegel was targeted [Spi15] for espionage by the NSA. While the espionage by the United States against a publisher such as Der Spiegel is serious, we cannot overstate the openly hostile CIA response directed at a newer publisher, WikiLeaks. The Director of the CIA (DIRCIA), Mike Pompeo, held a talk in 2017 in Canada at the Center For Strategic International Studies. In his talk, Pompeo said "*WikiLeaks walks like a hostile intelligence service*". On several other occasions he has repeatedly characterized WikiLeaks as a "*non-state hostile intelligence service*" in an attempt to dismiss the award winning publisher's constitutional protections as a journalistic endeavor. This classification is an attempt by the DIRCIA to attack members of the press that it cannot legally control. The CIA has no legitimate authority to regulate the free press or even specific publishers; rather Pompeo attempts a cheap propaganda trick that would make any fascist blush: the CIA invented a new category and stated that WikiLeaks is in that category. Furthermore, in his 2017 speech, Pompeo links the Taliban, WikiLeaks, and Al-Qaeda together as top concerns of the CIA. According to Dorfman, Naylor, and Isikoff [DNI21], CIA spies have also tried to reclassify journalists Greenwald, and Poitras as so-called *information brokers* [DNI21] in an attempt to prosecute them, and those who have worked [18] with them [19].

Furthermore, Pompeo claims in his 2017 speech that WikiLeaks is "*Often abetted by State actors like Russia*." This comment by Pompeo is a classic intelligence sleight-of-hand trick commonly known as *Deny, Counter Accuse*. Pompeo does not mention that Manning, who was convicted of leaking to WikiLeaks in a military court, is an American citizen. Nor does Pompeo mention that the largest leak of technically focused CIA documents appears to be sourced from someone who had insider access at the CIA. While it is possible that the CIA was simply hacked, it is also likely the case that WikiLeaks was abetted by an American state actor, namely a CIA insider. In fact, the CIA does not appear to think they were hacked by an outsider. A CIA employee named Joshua Schulte was arrested and tried over the Vault 7 leaks. His first trial ended without a conviction. Schulte's remarks about his prison conditions are particularly remarkable in that he speaks about the conditions of American prisons, and even highlights the plight of innocent people suffering immensely in those conditions. Schulte also drew attention to conditions in pre-trial detention, asserting that the accused are forced into punitive conditions, and that pre-trial detention is identical to post-conviction jails. Schulte specifically raised Kalief Browder's case [Fle19] as an example of the common injustices faced by those merely *accused* of being criminals and who are punished for daring to exercise their right to a trial.

Browder was accused of simple theft – stealing a backpack. He claimed he was innocent, and that he wanted to exercise his constitutional right to a speedy trial. Unfortunately for Browder, the American promise of a right to speedy trial is not equally available to all. Browder, a black teenager, was placed in the notorious Rikers Island prison for three years before his trial, two years spent in solitary confinement. He attempted suicide in prison and reported that the correctional officers in Rikers encouraged him to kill himself. Browder maintained his innocence and refused plea deal offers from the prosecution's office. The prosecutor attempted to use his pre-trial conditions as leverage to avoid a trial, but Browder continued to demand a trial. The prosecutors, after holding him for years in Rikers, were unable to bring a witness. The prosecution's inability to bring a witness was known while he was incarcerated, and in the end, they had no ability to bring a case. Two years later reportedly due to his traumatic experiences in prison, as a still innocent but now free person, Browder succeeded in committing suicide. Browder's treatment is not atypical of American pre-trial detention.

American prison conditions are indicative of systemic failures in providing basic Constitutional protections. The prison conditions reported by Schulte are no better and indeed may be worse because of increased attention from the CIA. The US government has also attempted to use these conditions to force a plea deal from Schulte, who, like Browder, demands his right to a trial. Schulte's second trial is scheduled for 2022. Is Schulte the Ellsberg or the Manning of the CIA? He maintains his innocence even while the US government engages the media in a campaign to smear his character. His case reveals CIA's near apoplectic fear at a single employee having such power to discredit even the DIRCIA himself.

---

[18] That includes the author of this thesis, an American citizen.

[19] Pompeo is the first director of the CIA in recent memory to overtly offer the too often awarded CIA prize for excellence in journalism: assassination by the CIA!

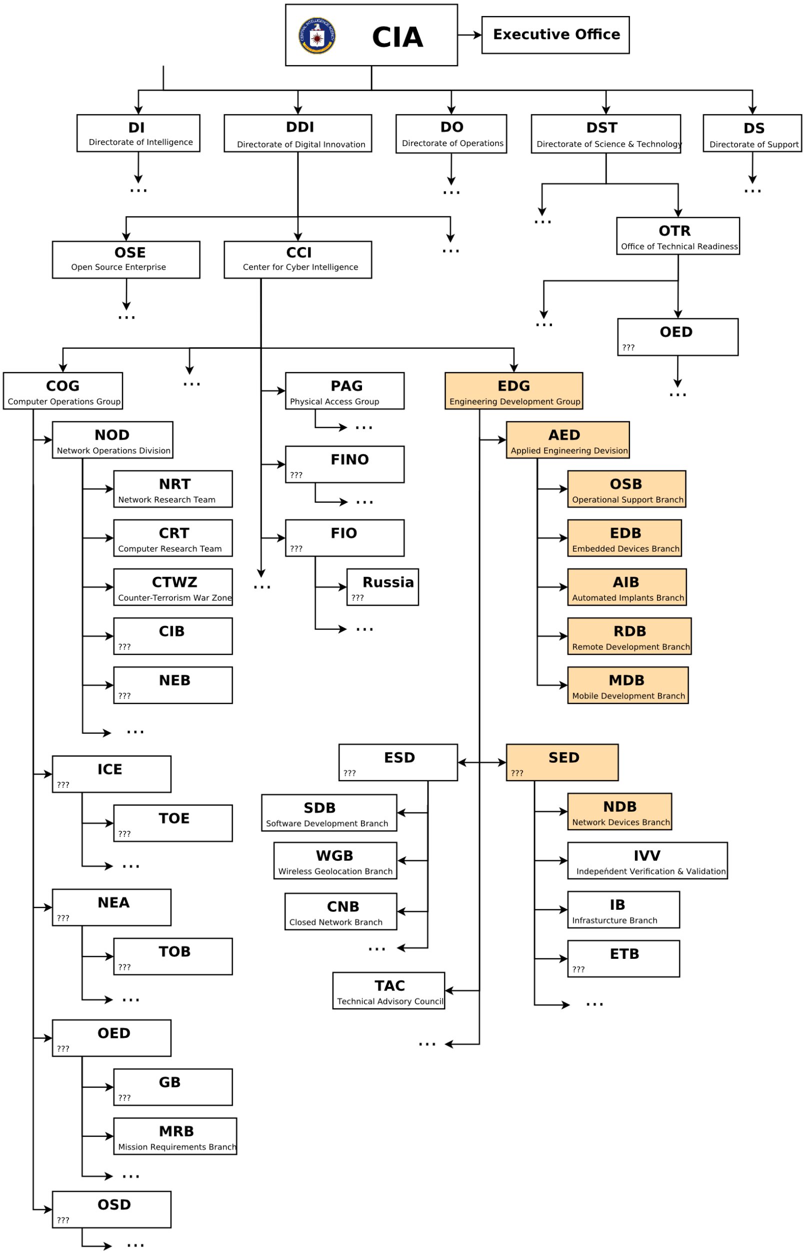

CIA
Executive Office
DI
Directorate of Intelligence
DDI
Directorate of Digital Innovation
DO
Directorate of Operations
DST
Directorate of Science & Technology
DS
Directorate of Support
OTR
Office of Technical Readiness
OSE
Open Source Enterprise
CCI
Center for Cyber Intelligence
OED
???
COG
Computer Operations Group
PAG
Physical Access Group
EDG
Engineering Development Group
NOD
Network Operations Division
AED
Applied Engineering Devision
NRT
Network Research Team
FINO
???
OSB
Operational Support Branch
CRT
Computer Research Team
FIO
???
EDB
Embedded Devices Branch
CTWZ
Counter-Terrorism War Zone
Russia
???
AIB
Automated Implants Branch
CIB
???
RDB
Remote Development Branch
NEB
???
MDB
Mobile Development Branch
ICE
???
ESD
???
SED
???
TOE
???
SDB
Software Development Branch
NDB
Network Devices Branch
WGB
Wireless Geolocation Branch
IVV
Independent Verification & Validation
NEA
???
CNB
Closed Network Branch
IB
Infrastructure Branch
TOB
???
ETB
???
TAC
Technical Advisory Council
OED
???
GB
???
MRB
Mission Requirements Branch
OSD
???
...

Figure 4.17: Central Intelligence Agency partial organizational chart
Courtesy of WikiLeaks [Wik17k].

In response to WikiLeaks' publication activities, and in contrast to the way Der Spiegel was targeted by US intelligence, the CIA demonstrated its commitment to liberal democratic values by plotting various attacks on Assange, including kidnapping, poisoning, and shooting him. Once these plans were revealed [DNI21] to the world by Dorfman, Naylor, and Isikoff, Pompeo said that the sources for the article should be arrested for disclosing classified information, ineptly confirming the main points of the article as valid. The repression against WikiLeaks continues to the current day. The CIA has approached the British government to discuss assassinating Assange in London *for his publication activities*. Though Assange was under the protection of the Ecuadorian government as a recipient of political asylum, the British reportedly [DNI21] offered to carry out the shooting for the CIA. This is not so dissimilar from the situation where the United Kingdom, France, United States, and Belgium worked together to assassinate [DW02] Patrice Lumumba [20]. Ultimately the Belgian government was responsible, and in 2002 Belgium apologized [Age02] for their contribution to the assassination. A key difference with WikiLeaks is that the British government, working with the CIA, has yet to succeed in killing Assange. One way that academics may assist in thwarting the goals of the British government and the CIA is by simply citing and discussing the publications of WikiLeaks. Another way is by naming the people involved, though there are severe American legal sanctions on a per-name basis for revealing undercover operatives. However, there are legal conflict between the duty to stop crimes versus the crime of revealing the names of the spies who carry out assassination. There is precedent for this in Italy, where public prosecutors have tried to prosecute individual CIA agents who allegedly participated in American kidnapping and torture programs after 2001. Ensuring that the CIA may not carry out assassinations on European soil without consequences seems of higher importance than protecting the identities of people who plot to murder perceived journalistic or political enemies.

4.3 — Summer of Snowden and the post-Snowden Era

A turning point in the discussions of mass surveillance happened in early 2013 thanks to the whistleblowing of Edward Snowden. Until Snowden, those who were closely paying attention understood that previous whistleblowers were not making up their claims but it was still possible to dismiss many of the claims. The publications surrounding Snowden's whistleblowing changed that dynamic entirely. Evidence of mass and targeted surveillance has been extensively documented in newspapers [GS13b, Gel13], films [Poi14,Sco14,Par15,Boo15,GH15,Poi16,RA16,Sto16,Poi17,Bro17], books [Gre14b, RS14, TW17, Gel21], magazines [ABG+13, AGK+14, KGE+14, GWE+15, AGG+15a], and many other forums [Rob14, Pre17, Pre16].

Any summary of the post-2013 landscape regarding surveillance, be it targeted or mass surveillance, will fall short of fully capturing the totality of the events relating to the disclosures sparked by Snowden. With a nod towards the then recent Arab Spring, the summer of 2013 was lovingly referred as the *Summer of Snowden* by many of the journalists involved in publishing Snowden-related documents. To fully appreciate the breadth of the publishing, one needs to review the periodical newspapers, magazines, films, public policy discussions, videos of debate panels, and other related forums. It

---

[20] Lumumba was slandered constantly as part of a neo-colonialist struggle between European powers and national liberation movements in Africa. The struggle was painted in Western media as a battle between Western Capitalists and Eastern Communists, though Lumumba was emphatically not a Communist. The smears shaped the responses to his assassination for roughly forty years before the truth was uncovered [DW02].

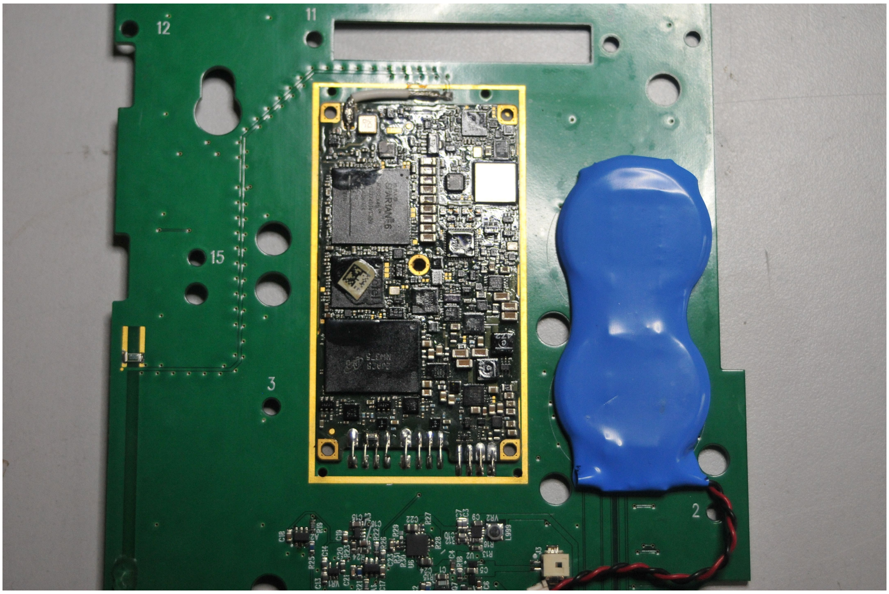


Figure 4.18: Hardware implant added to Andy Müller-Maguhn's IP19 Cryptophone Courtesy of Andy Müller-Maguhn [MM18].

is also important to look at publications from the decade previous to Snowden's leaks to witness the magnitude of the changes. Legal indictments were written and served, awards were given out, and many politicians untainted by mass surveillance expressed their gratitude. Others who were caught red handed loudly denounced Snowden and the people who worked to expose what he felt was so criminally unconstitutional as to warrant a life in exile rather than to remain silent. Snowden is not alone in his exile from America; he is part of a set of people who worked in national and international contexts to ensure that certain truths, backed by evidence, could be known far and wide, far beyond any single country. In this regard, Assange and Snowden should be seen as another nexus of activity, so when their paths crossed in 2013, the result became a part of the fabric of history of helping to publish highly contentious true facts. Were it not for Assange's commitment to source protection, and Section Editor Sarah Harrison's proactive actions in support of that commitment, Snowden would almost certainly be rotting away in a jail cell as Manning was at that very moment in history. What is it that Edward Snowden risked his life to reveal?

The full scope of the Snowden archive spans many different areas and so it is insufficient to simply read a handful of articles or a single book on the topic. In this thesis we focus on aspects of the Snowden archive that concern surveillance, censorship, betrayal of the public trust, and with a keen eye towards human rights considerations in the processing and use of the pilfered data.

One of the most critical document collections published as a result of Snowden's whistleblowing is the confirmation of sabotage by the NSA and their accomplices under the program name BULLRUN [AGG$^+$14d, BBG13, PLS13, Lar13]. The BULLRUN program was active at the time of publication and is presumably still active, perhaps under a different codename. It includes attacks on the security of hardware, software, firmware; corruption of national and international standards; the insertion of key intelligence personnel at specific American or international corporations; and other serious matters that defy easy categorization. Many of the original BULLRUN stories lacked serious technical detail, and of course the specific undercover assets who performed sabotage against commercial, government, and even hobbyist cryptographic software were not exposed. Many of the companies that went along willingly had their developer names, product names and company names scrubbed from documents before release. The NSA describes devices as *enabled* when NSA or a partner has sabotaged it. CPU manufacturers [21] who have had their products "*enabled*" remain largely unknown, and unconfirmed in public. Many journalists who have worked on the Snowden archive know significantly more than they have revealed in public. It is in this sense that the Snowden archive has almost completely failed to create change: many of the backdoors and sabotage unknown to us before 2013 is still unknown to us today. Sometimes the backdoor added by an Adversary is **in** software, hardware, or standards documents; other times, the backdoor **is the threat model** [22].

4.3.1 – Mass surveillance of the Internet. Understanding mass surveillance is possible from many positions – legal, technical, political, and operationally are all lenses reflected in the leaked Snowden documents as well as other documents. Seeing through the eyes of a surveillance adversary often requires considering their internal training materials as more than simply training in technical matters. It is a reflection of the actual legal and political situation on a day-to-day basis. It is a reflection of technical ability, and of operational political values and legal conclusions. A large amount of the legal analysis regarding systems and how they are used appears untested in *open* court, and has not yet achieved a victory over constitutional challenges. Some cases, such as the Presidential Surveillance Program (PSP) [Wik22b], also known as its codename STELLARWIND [Wik21q], was tacitly acknowledged when years later involved parties such as corporate and government executives were given retroactive immunity, and old practices were then brought into law, if only vaguely. On its face, a legal analysis of the Fourth Amendment [23] of the Bill of Rights should lead to a rejection of *general warrants*, considered a tyranny of the British at the time of the framers work on the Bill of Rights. A rejection of general warrants seems to plainly outlaw bulk wiretapping, even if later that bulk data is searched for particular patterns which are later supported by probable cause and an oath or affirmation. It is clear that non-technically inclined judges often disagree with this assessment, and often side with the surveillance adversaries. Nevertheless, the NSA and other intelligence agencies or government agencies do try their best to *collect it all* [Gre13c] and *exploit it all* regardless of judicial review, and let the analysts sort it out. It is not just the NSA: consider the United States Naval Intelligence motto: "*In God*

---

[21] While working on documents in the Snowden archive the thesis author learned that an American fabless semiconductor CPU vendor named Cavium is listed as a successful SIGINT "*enabled*" CPU vendor. By chance this was the same CPU present in the thesis author's Internet router (UniFi USG3). The entire Snowden archive should be open for academic researchers to better understand more of the history of such behavior.

[22] Consider the threat models of Onion routing and mixnets. Only mixnets even claim to stand up to worldwide coordinated mass surveillance, the only relevant threat model for an anonymity system.

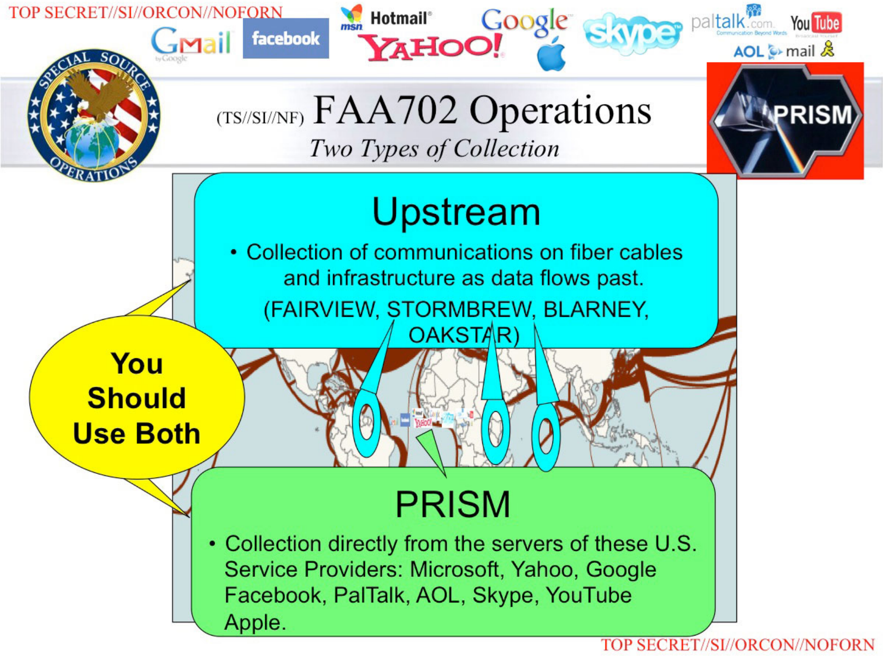
[23] "*The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.*" [Wik22a]

we trust; all others we monitor!" [Tur14]. This theme is common among spies of most organizations and nationalities.

Internet data is collected by NSA and their partners using myriad of different technical programs that in turn have program names for collections of programs. An example of this is the phrase "*FAA702 Operations*" which is used as an umbrella term as seen in Figure 4.19. This figure shows that FAA702 Operations includes data captured from two general types of surveillance. Commonly seen in documents similar to Figure 4.19, the NSA uses the euphemism collection rather than surveillance, code names for programs are generally in uppercase such as PRISM, and Special Source Operations is an NSA division – not to be confused with Special Collection Service (SCS – codename F6) which is a joint CIA and NSA program. It is helpful to learn the informal language of spy-agency bureaucrats who write slide decks as their writing is sometimes the only documentation available to the public other than oral histories. Documents leaked from Five Eyes countries often include a SIGINT Activity Designator [Wik21p, NSA05, Ele21] (SIGAD) to identify a specific surveillance collection program numerically. US-984XN is the classified SIGAD while the program name PRISM is unclassified. Additionally, SIGADs may be followed by a colon and then a Producer Designator Digraph [Ele21, NSA05] (PDDG). The PDDG denotes a code for the producer or the group performing the actual surveillance. When possible, we list programs by codename and SIGAD to aid researchers in breaking compartmentalization to encourage holistic understanding of mass surveillance rather than simply considering each program in isolation. The NSA also uses the term *partner* when a corporation or government agency willingly works with them, and they use the term *unilateral* when parties are unwilling to work with NSA or if they are unknowingly assisting the NSA; the NSA is successfully performing surveillance in both cases.

The first program featured in the main body of Figure 4.19 is called *Upstream* which in turn is made up of a collection of programs named *FAIRVIEW* (AT&T), *STORMBREW* (Verizon), *BLARNEY* (AT&T) for domestic mass surveillance and *OAKSTAR* (Unknown) for international surveillance of Internet data. OAKSTAR is a collection of at least eight programs for targeting international mass surveillance: BLUEZEPHYR (US-3277), COBALTFALCON (US-3354), MONKEYROCKET (US-3206:6T)), ORANGEBLOSSOM (US-3251), ORANGECRUSH (US-3230:0B), SHIFTINGSHADOW (US-3217:MU), SILVERZEPHYR (US-3273:SK), and YACHTSHOP (US-3247:PJ). Like American nesting dolls, ORANGECRUSH is a cover term that includes the program PRIMECANE, a program for access involving unknown third parties. YACHTSHOP is a cover term that includes the program partner code named BLUEANCHOR with a SIGAD of US-3247 and a PDDG of PJ. SILVERZEPHYR is a cover term that includes the program's unknown partner code named STEELKNIGHT. Interestingly, SILVERZEPHYR's legal justification includes reference to the Transit Authority (TA) and FISA Amendment Act (FAA) which strongly implies a partner with facilities on US soil. The various Tier-1 Internet transit carriers such as Level 3 are obvious suspects, and indeed the MUSCULAR program showed that to target Google *unilaterally* the NSA worked with Google's fiber optic network provider to gain access to the internal Google network from another angle. A map with a partial view of FAIRVIEW equipment is shown in Figure 4.21. The map bears the marks of the NSA Special Source Operations (SSO). The map also underscores that the domestic surveillance program is in close partnership with the FBI.

The MUSCULAR (DS-200B) [GS13a] program shows how US intelligence teams up with British intelligence to spy on the internal networks of companies such as Yahoo!



TOP SECRET//SI//ORCON//NOFORN

(TS//SI//NF) FAA702 Operations

Two Types of Collection

Upstream

- Collection of communications on fiber cables and infrastructure as data flows past. (FAIRVIEW, STORMBREW, BLARNEY, OAKSTAR)

You Should Use Both

PRISM

- Collection directly from the servers of these U.S. Service Providers: Microsoft, Yahoo, Google Facebook, PalTalk, AOL, Skype, YouTube Apple.

TOP SECRET//SI//ORCON//NOFORN

Figure 4.19: PRISM: FAA702 Operations: Two Types of Collection
Courtesy of Glenn Greenwald [GM13b].

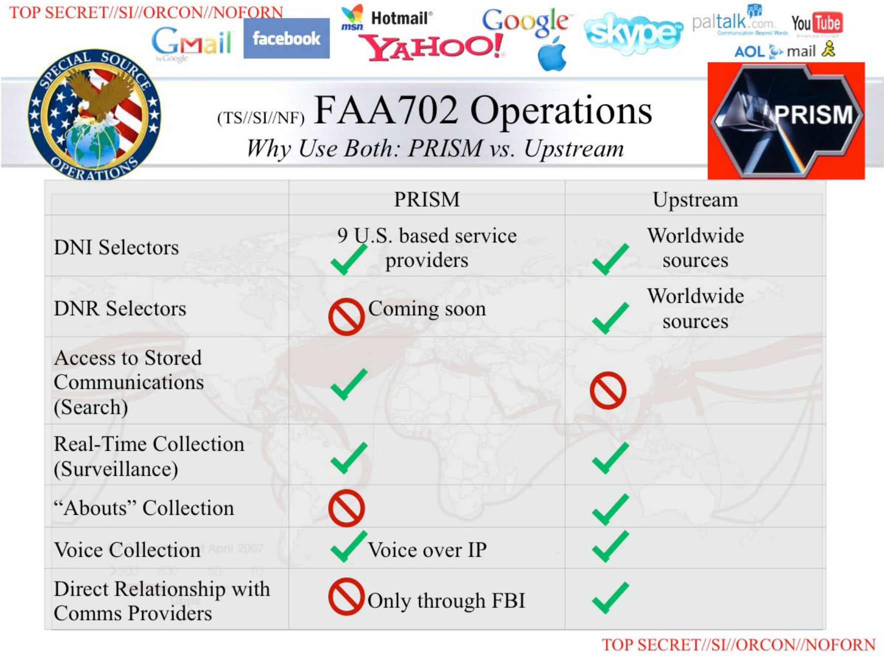
and Google. The program began operation in 2009, ingesting around 20 Gb/s of traffic daily. While the traffic for both is nominally protected by encryption such as HTTPS, their internal networks were not secured from the perspective of a surveillance adversary. NSA and GCHQ worked with the ISPs of both firms, gained unilateral access to surveil internal and external network traffic, and used their access to extract interesting data from the flows of data such as address books, web cam traffic, email, and much more.

Data stored with providers is also subject to search and seizure without any notification to the targeted accounts or the people associated with those accounts. The PRISM [GM13b] (SIGAD: US-984XN) program is defined as "*Collection directly from the servers of these U.S. Service Providers: Microsoft, Yahoo, Google, Facebook, PalTalk, YouTube, Skype, AOL, and Apple.*" Dropbox is mentioned as a future PRISM partner as shown in Figure 4.19 and Figure 4.25. An architectural diagram as shown in Figure 4.23 and Figure 4.24 shows how the NSA has access to these American corporations: through a partnership with the United States Federal Bureau of Investigation (FBI).

Weaknesses intentionally added to systems such as Google's email service are obvious targets for attack by other intelligence services or simply curious hackers. Completely unsurprisingly, Google's interception system was compromised [KN11] by adversaries affiliated with Chinese intelligence and used to spy on American officials, journalists, Chinese dissidents, and other sensitive targets [Nak13a] of interest to the attackers. Similarly unsurprisingly is that Yahoo! [24] was *further* compromised [Pet16] by US intelligence, and

---

[24] There is a kind of logic at play which is not completely obvious without personal experience being harassed


GMail
facebook
msn Hotmail
YAHOO!
Google
skype
paltalk.com
YouTube
AOL mail
SPECIAL SOURCE OPERATIONS
PRISM

TOP SECRET//SI//ORCON//NOFORN

(TS//SI//NF) FAA702 Operations

Why Use Both: PRISM vs. Upstream

	PRISM	Upstream
DNI Selectors	✓ 9 U.S. based service providers	✓ Worldwide sources
DNR Selectors	⊘ Coming soon	✓ Worldwide sources
Access to Stored Communications (Search)	✓	⊘
Real-Time Collection (Surveillance)	✓	✓
"Abouts" Collection	⊘	✓
Voice Collection	✓ Voice over IP	✓
Direct Relationship with Comms Providers	⊘ Only through FBI	✓

TOP SECRET//SI//ORCON//NOFORN

Figure 4.20: PRISM: FAA702 Operations: Why Use Both: PRISM vs. Upstream
Courtesy of Glenn Greenwald [GM13b].

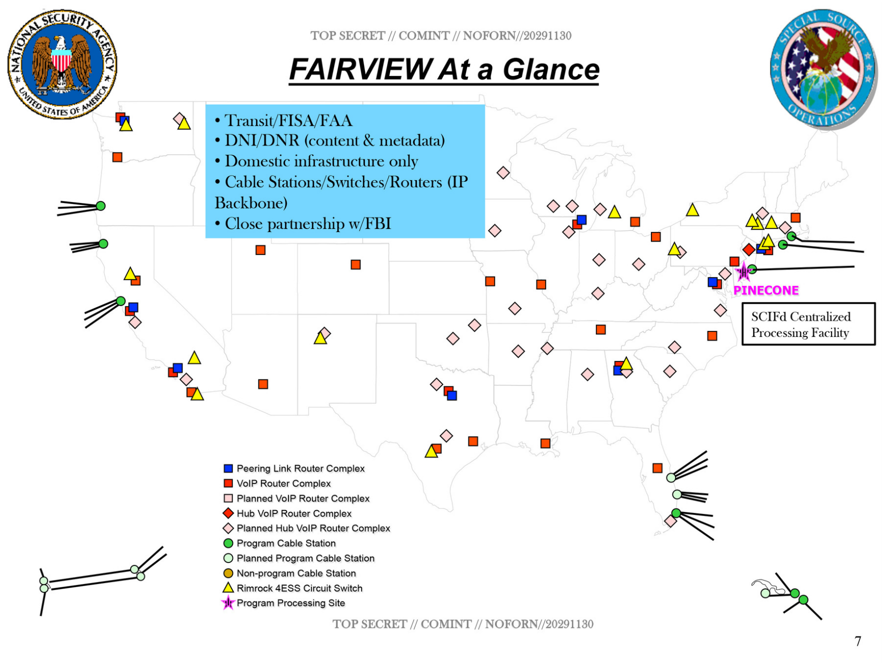
others [TT16] who remain unidentified. Adding backdoors to an already insecure system is not a reasonable path towards building a secure system. The technical debate on this matter is settled. It is not just obvious that they will be hacked, it is a matter of record that companies and indeed officials setting public policy *have been compromised* as a result of these shortsighted, externally-enforced architectural decisions.

The FBI Data Intercept Technology Unit (DITU) interfaces with the corporate networks and acts as a proxy for the NSA's surveillance queries. The DITU is part of the FBI Data Intercept Technology Program (DITP). The Operational Technology Division (OTD) of the FBI is the umbrella division that is responsible for the development, deployment, and use of the DITP. The specific internal FBI hierarchy is OTD/TCB/CIS/DITU. [25]

---

by the FBI: Yahoo email addresses are the preferred email given out by FBI employees who are attempting to recruit confidential informants. Part of the reason is that using Yahoo! is a way to ensure that any communications will not be available on FBI systems tied to the specific person at FBI. Accountability and transparency are often out of reach with the FBI, it is no different when they use Yahoo! services as part of their tradecraft. This activity continues until today and has never been properly investigated in the open.

[25] Equipment for the DITU has been delivered to FBI building 27958A in Quantico, Virgina.



TOP SECRET // COMINT // NOFORN//20291130

FAIRVIEW At a Glance

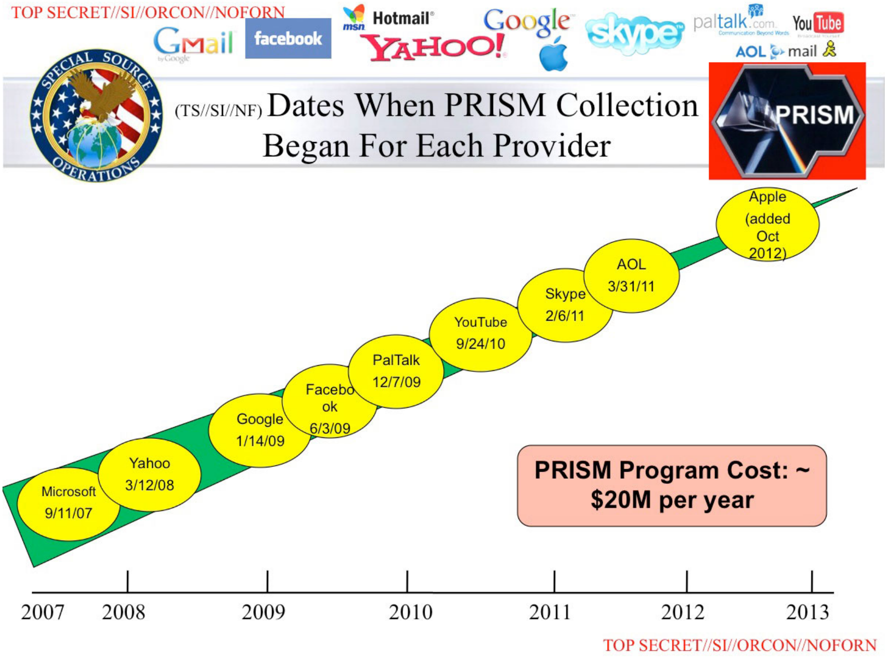
- Transit/FISA/FAA
- DNI/DNR (content & metadata)
- Domestic infrastructure only
- Cable Stations/Switches/Routers (IP Backbone)
- Close partnership w/FBI

🟦 Peering Link Router Complex
🟥 VoIP Router Complex
◻ Planned VoIP Router Complex
◆ Hub VoIP Router Complex
◇ Planned Hub VoIP Router Complex
● Program Cable Station
○ Planned Program Cable Station
● Non-program Cable Station
△ Rimrock 4ESS Circuit Switch
☆ Program Processing Site

TOP SECRET // COMINT // NOFORN//20291130

7

Figure 4.21: Geography of FAIRVIEW
Courtesy of Glenn Greenwald [GM13b].



TOP SECRET//SI//ORCON//NOFORN

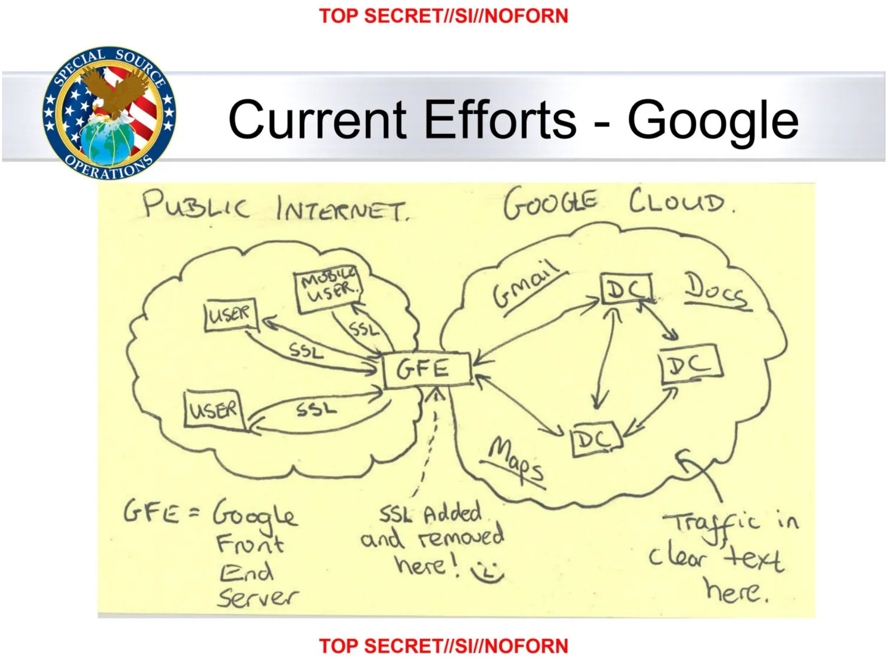
(TS//SI//NF) Dates When PRISM Collection Began For Each Provider

Microsoft 9/11/07
Yahoo 3/12/08
Google 1/14/09
Facebook 6/3/09
PalTalk 12/7/09
YouTube 9/24/10
Skype 2/6/11
AOL 3/31/11
Apple (added Oct 2012)

PRISM Program Cost: ~ $20M per year

TOP SECRET//SI//ORCON//NOFORN

Figure 4.25: Dates When PRISM Collection Began For Each Provider
Courtesy of Glenn Greenwald [GM13b].



TOP SECRET//SI//NOFORN

Current Efforts - Google

PUBLIC INTERNET.
GOOGLE CLOUD.
MOBILE USER.
USER
USER
SSL
SSL
SSL
GFE
Gmail
DC
Docs
DC
DC
Maps
GFE = Google Front End Server
SSL Added and removed here! :-)
Traffic in clear text here.

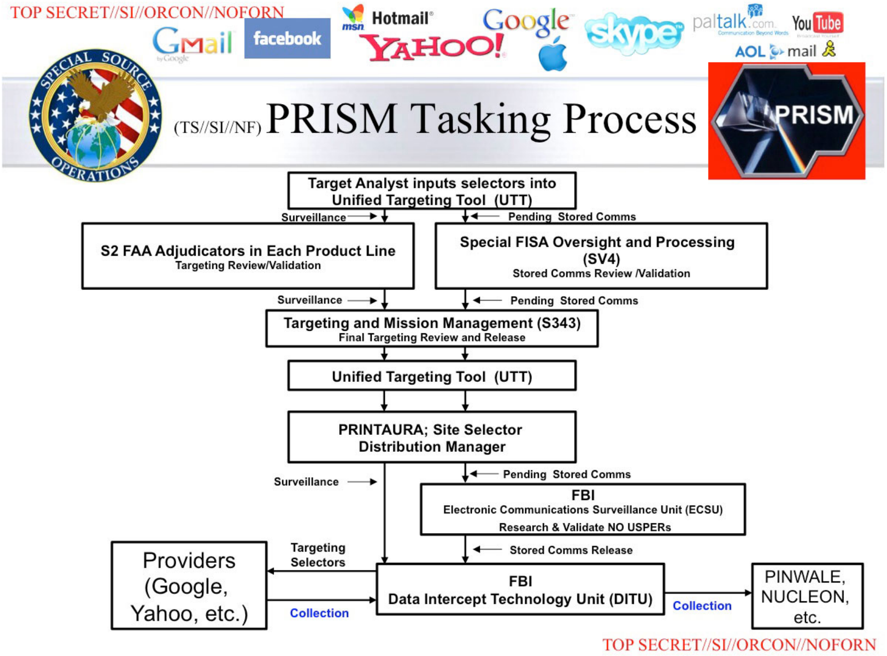
TOP SECRET//SI//NOFORN

Figure 4.22: "SSL added and removed here! :-)"
Courtesy of Washington Post [GS13a].

The FBI DITU allows CIA, NSA, and FBI personnel and their systems access to the PRISM partner data. In this way, it can and is often dishonestly said that *the NSA does not have direct access* to Apple's or any PRISM partner's network. Through the use of the Unified Targeting Tool (UTT) as seen in Figure 4.23, an analyst does not need to know who ultimately runs the equipment, they can simply pick selectors and otherwise private user data is returned to them. The PRISM slide deck was not published in full, and the public does not fully understand aspects of the program such as the retrieval of voice content data as seen in Figure 4.24. Domains hosted by PRISM partners are also subject to selector based surveillance.

Several pages of the PRISM slides list targets and related surveillance data, and a majority of them appear to be a matter of political surveillance rather than defense against terrorism. One example that is not well-known except among the journalists who had access to the full PRISM slide deck is the explicit naming of targets. An example shows a suggestion for targeting of the Tibetan Government in Exile through their primary domain name. The `tibet.net` domain is named as an unconventional example that analysts should be aware of as also falling under the purview of PRISM. The email domain was hosted by Google Mail, a PRISM partner, at the time of the slide deck creation and it is still currently hosted by Google Mail as of early 2022.

The example of `tibet.net` underscores a non-obvious political reality of accepting aid from the United States of America. The Tibetan system administrators wanted to have security in the form of protection from Chinese hacking and surveillance operations. Rather than self-hosting, the technical team behind `tibet.net` decided to host



TOP SECRET//SI//ORCON//NOFORN

(TS//SI//NF) PRISM Tasking Process

Target Analyst inputs selectors into Unified Targeting Tool (UTT)

Surveillance

Pending Stored Comms

S2 FAA Adjudicators in Each Product Line
Targeting Review/Validation

Special FISA Oversight and Processing (SV4)
Stored Comms Review /Validation

Surveillance

Pending Stored Comms

Targeting and Mission Management (S343)
Final Targeting Review and Release

Unified Targeting Tool (UTT)

PRINTAURA; Site Selector Distribution Manager

Surveillance

Pending Stored Comms

FBI
Electronic Communications Surveillance Unit (ECSU)
Research & Validate NO USPERs

Stored Comms Release

Providers (Google, Yahoo, etc.)

Targeting Selectors

Collection

FBI
Data Intercept Technology Unit (DITU)

Collection

PINWALE, NUCLEON, etc.

TOP SECRET//SI//ORCON//NOFORN

Figure 4.23: PRISM Tasking Process
Courtesy of Glenn Greenwald [GM13b].

with Google for email, and with Cloudflare for web hosting. Part of the reason behind this choice was that Google had an excellent reputation at the time for having a talented security team. What was unknown at the time of this decision was that Google would, willingly or unwillingly, give up the data to the US government in secret. Thus in seeking to prevent surveillance by the Chinese government *some of the time* when the Chinese government successfully hacks their servers, they unknowingly accepted aid [26] that ensures their data will be under surveillance *all of the time*. Meanwhile, Google is able to promote a sense of good will by offering their white hat security paladin services to the Tibetans for free. In this case, Google knowingly or unknowingly has sold them out. Those who know the details are likely legally gagged from speaking about their complete betrayal of their security promises to their extremely vulnerable Tibetan customers. Needless to say that the development of hosting their web sites with Cloudflare raises similar concerns as Cloudflare has been subjected to at least one National Security Letter (NSL) that included a legal provision gagging them from speaking about the NSL.

4.3.2 – Telephone surveillance. MAINWAY [Wik21k] is an NSA program that consists of telephone metadata of the largest telephone providers in the United States of America. It was a key part of the 1990s ThinThread [Wik21s] program. The NSA claims that it only retains data about domestic telephone calls in the MAINWAY program for five years. There is little reason to believe what the NSA or any official related to the NSA has

---

[26] This included State Department grants to rebuild infrastructure in Dharamsala, India and the author of this thesis was directly involved in these matters.

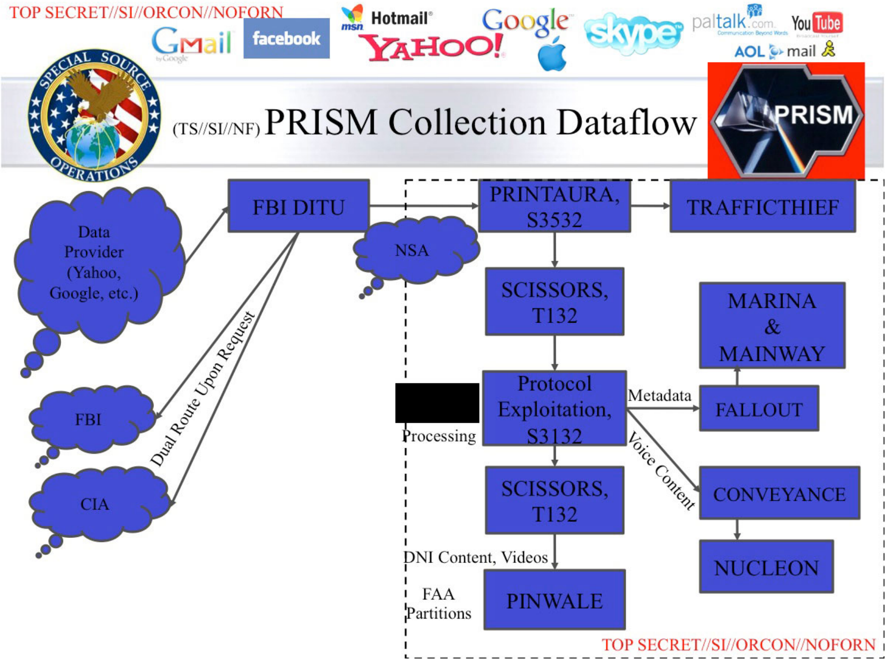

TOP SECRET//SI//ORCON//NOFORN
GMail
facebook
msn Hotmail
YAHOO!
Google
skype
paltalk.com
YouTube
AOL mail
SPECIAL SOURCE OPERATIONS
(TS//SI//NF) PRISM Collection Dataflow
PRISM
Data Provider (Yahoo, Google, etc.)
FBI DITU
NSA
PRINTAURA, S3532
TRAFFICTHIEF
SCISSORS, T132
MARINA & MAINWAY
FBI
CIA
Dual Route Upon Request
[redacted]
Processing
Protocol Exploitation, S3132
Metadata
FALLOUT
Voice Content
SCISSORS, T132
CONVEYANCE
DNI Content, Videos
NUCLEON
FAA Partitions
PINWALE
TOP SECRET//SI//ORCON//NOFORN

Figure 4.24: PRISM Collection Dataflow
Courtesy of Glenn Greenwald [GM13b].

to say on matters of data retention.

MYSTIC is a wide-ranging telephone surveillance program. Capabilities that were previously considered to be a technological ability only in science fiction were realized when MYSTIC was revealed [RS14, GS14, DGP14] to the world. For countries targeted with the MYSTIC program, a subset is further targeted by SOMALGET. The MYSTIC program is a metadata telecommunications intercept system. SOMALGET is the full content collection program. The entire telephone network of entire countries is monitored and data is put into an indexed, searchable dataset. For countries targeted by SOMALGET, a *full-take audio* surveillance program, the original voice data is not only captured but it is additionally analyzed. MYSTIC was revealed to impact a number of countries by name at the time of publication: the Bahamas, Mexico, the Philippines, Kenya and one mystery country: country X. The Bahamas, and country X are subject to SOMALGET full take data and voice collection. The publisher WikiLeaks observed that the monitoring of an entire country of people is a crime when done by outside parties, essentially an act of war by the surveillance adversary. WikiLeaks then revealed that the country in question, Country X, was Afghanistan [CYp14]. Through independent review of the Snowden archive, we confirm that this is the identity of Country X, and that WikiLeaks was correct in their claim. Now that the American war in Afghanistan has ended it seems noteworthy that even with reportedly full-take audio telephone surveillance as provided by the SOMALGET program, America still left Afghanistan as losers after a twenty year war.

Why did total telephone network surveillance not ensure that America and its embarrassing *coalition of the willing* would win the war? Perhaps the local resistance to the occupation knew that all the local telephone companies were compromised, and avoided using the telephone for anything sensitive. Perhaps it was a problem of analysis. Perhaps it was a success but the goals of the program were simply more modest than the goals of the overall war. Either way, it does suggest that total surveillance of telephones does not ensure that the surveillance adversary will win the larger conflict. Full-take telephone surveillance of entire countries is not compatible with the rule of law, nor is is compatible with a notion of proportionality. During wartime it appears to have failed to ensure winning the war, while during peacetime it seems to be obviously unconstitutional, and depending on jurisdiction outright illegal.

In historic context telephone exchanges have usually been targeted for their innate power even without full-take audio, or any systematic, real time metadata analysis. In the Spanish civil war, battles were fought over the Barcelona telephone exchange; the building still has scars of bullets from the conflicts fought in the civil war. Control of the exchange ensured that the winner could not only safely communicate but that they could do traffic analysis, and surveillance, on any telephone communication that routed through the exchange. Eventually, Barcelona fell to the Franco Fascists and the control of the telephone exchange has long been discussed as a major factor. Similarly, the Nazi use of telephone exchange data and metadata during their Second World War fundamentally changed some post-war billing systems. Certain countries seem to have understood the risks of infrastructure capture after their infrastructure and the data produced by that infrastructure had fallen into the hands of the Nazis.

A modern example of exploiting telephone metadata can be found in the NSA program CO-TRAVELER as revealed [GS13b] by Barton Gellman and Ashkan Soltani in the Washington Post. The purpose of CO-TRAVELER is to find pairs or more of people which appear to be traveling together from cell phone tower logs and other sources of data. The CO-TRAVELER program highlights the power of metadata to uncover things that are traditionally considered content analysis problems. Problematic logging practices, especially with regard to billing, is likely to be a source of data for the NSA. Among surveillance experts, it is well-known that some countries ban the outsourcing of telephone metadata on national security grounds (e.g. Switzerland), while other countries appear to not object with such billing services being done by the lowest bidders by obvious intelligence fronts. Why even bother to occupy or compromise the exchange? The telephone companies simply give the data away to the lowest bidder, paying for the service of turning the metadata into paper or electronic billing products. Naturally nothing stops those billing companies from reselling the data, little stops someone else from stealing it from those companies, and when the company is an intelligence front – they may do a good job of securing the data from everyone else but their own analysts. It is reasonable to assume that many other countries have developed programs similar to CO-TRAVELER, and there is no reason to believe that the NSA has stopped collecting the data which powered CO-TRAVELER, or that their use of this data has diminished in any way. Indeed, CO-TRAVELER is a surveillance tool that needs very little data when compared with MYSTIC and SOMALGET. The set of these three programs may be seen as incremental steps toward reaching the goals behind the *Collect-it-all* strategy. Each step increases the different kinds of surveillance analysis that is possible.

4.4 — Standardization of cryptographic sabotage

Unsurprisingly, the NSA continues to try to influence public cryptographic standards. Of particular interest after the BULLRUN program was exposed, and after the NSA role in paying bribes to include DUAL_EC in production RSA Security corporation systems (BSAFE toolkit) was exposed [Men13, Men14], is their attempt to push a questionable family of block ciphers. The NSA and their role in the ISO/IEC standardization process of Simon and Speck Block Cipher was well documented [AL21] by involved parties, cryptographers Tomer Ashur and Atul Luykx. Their summary report of the standardization process shows a continued hostility by the NSA to the international cryptographic community. Ashur was confronted by prominent members of the computer security industry who for some reason avoid total condemnation of the NSA, even after NSA was caught red-handed with only **some** of their backdoor efforts. The resulting response, *How to Backdoor a Cipher* [PA21] was an instant classic.

In cryptography, there is oft-repeated folklore about NSA being helpful with cryptography: "*NSA made DES stronger*." Specifically the folklore is that NSA knowledge of differential cryptanalysis resulted in a concrete security improvement that went beyond the public understanding. It is unclear where this claim originates. Primary source documents show [Joh98] that IBM, true to their other historical activities, collaborated with NSA, and agreed to reduce the key size to 56 bits (NSA wanted 48 bits, IBM wanted 64 bits). In 1979, Hellman declared: "*DES will be totally insecure within ten years*" [Hel79]. He stated that the key length should be doubled.

In the 1990s, a team of freely associated experts led by John Gilmore were able to document [Fou98] and build a specialized machine to attack DES. Gilmore's team was able to break DES in a very practical manner, and they advanced the cause of free speech by publishing a book with policy, software, and hardware schematics for anyone in the world to repeat. Consider NSA's goals in their own words:

> "*Narrowing the encryption problem to a single, influential algorithm might drive out competitors, and that would reduce the field that NSA had to be concerned about. Could a public encryption standard be made secure enough to protect against everything but a massive brute force attack, but weak enough to still permit an attack of some nature using very sophisticated (and expensive) techniques?*" [Joh98]

A lesson from these events is that NSA's cryptographic concerns are not the cryptographic concerns of cryptographers who work in public. For further discussion of standardization, and NSA, see Bernstein's paper [Ber20] about cryptographic competitions.

The NSA's behavior is understood perfectly by the goals of project BULLRUN: sabotage public cryptography to their own benefit, ideally with NOBUS style backdoors. NOBUS is a way to characterize sabotage of a system such that only NSA is able to exploit it, it is generally understood to be *Nobody but US*(A). This behavior is not limited to DES or DUAL_EC_DRBG [BLN16] or by suggesting weak key sizes. The idea of a NOBUS style backdoor is in contrast with a bugdoor style backdoor where a subtle or not so subtle software or hardware bug is left in place or inserted purposefully. In cases like the introduction of DUAL_EC_DRBG in Juniper devices [CMG$^+$16] is a mix of both. In the case of Juniper, a different group appears to have changed the Q parameter in the DUAL_EC_DRBG used in Juniper devices. The newly replaced Q parameter does not correspond to the NSA's secret key but to a completely unknown adversary. "*Nobody but US(A)*" seems to not hold in the DUAL_EC_DRBG case in practice unless one holds that once the Q parameter is changed it is not DUAL_EC_DRBG anymore. Strictly speaking this is probably true, but it further makes NOBUS seem like a point of rhetoric rather than a tight definition of a valid security strategy for backdoors. If NSA adds a backdoor to an otherwise secure system, and *then another unknown adversary* changes that backdoor to their advantage, what does it matter that the original was NOBUS? Building secure systems is hard enough – we are better off without backdooors or sabotage.

One question that is raised here is what exactly is the definition of NOBUS? If DUAL_EC_DRBG's Q parameter is recoverable by a hypothetical cryptographically relevant quantum computer, NOBUS cannot be a claim of *forever* being NOBUS. It seems that more likely a NOBUS backdoor merely means at the time of deployment, NSA had some unique advantage, and it is not actually intended to be a unique advantage for all time. An additional reason that NOBUS is raised as a security option as a strategy is to encourage the design, deployment, and use of sabotaged systems rather than secure systems. The overall strategy is paternalistic in nature: NSA knows best. NSA may very well know best, though their interests are not the national interests, and certainly not anyone's personal best interest. NSA does not have an obligation to release their cryptographic breaks even when they apply directly to American cryptographic national standards.

Juniper introduced the NOBUS style DUAL_EC_DRBG backdoor in 2008 to ScreenOS, and later another attacker, speculated by security researchers to be a Chinese state-sponsored hacking group ("APT 5"), changed the Q parameter for DUAL_EC_DRBG to no longer use the NSA controlled Q parameter but rather to use a Q parameter that was useful to the attackers who controlled the corresponding private key. This new DUAL_EC_DRBG enabled software was then deployed unknowingly by Juniper to customers. This backdoor allowed *someone* to break the cryptography in the Juniper devices, and it allows for full decryption of encrypted traffic to and from the Juniper security device. These devices are usually used for providing IPsec VPN services. Juniper commented on the use of DUAL_EC_DRBG in 2013 in relation to the Snowden affair, and they claimed to include counter measures that would make anything problematic about DUAL_EC_DRBG practically unexploitable. It is not clear if Juniper was intentionally being dishonest or not, and later their counter measures were found to be completely ineffective. This and *other* backdoors in Juniper's ScreenOS were not investigated in public until late 2016 when it was shown to be practically exploitable [CMG+16]. The research in 2016 by many high profile cryptographers [CMG+16] conclusively shows that NSA, by controlling the secret key for the corresponding Q parameter, was able to *passively* decrypt IPsec traffic. Evidence of NSA's exploitation of this type of sabotage is apparent. Many documents released in public from the Snowden archive and additional documents which are still not public make clear that this type of bug is being exploited at scale with help from NSA's surveillance infrastructure. It is still unclear who authored the changes at Juniper and if bribery from the NSA was involved as with RSA's deployment of DUAL_EC_DRBG to their customers as is discussed in Section 4.4. Furthermore, that IPsec can be sabotaged with a broken random number generator seems to be just the kind of standardization sabotage that NSA pushes for in IETF and ISO meetings. Usually NSA does this while undercover, and sometimes it is done openly.

DUAL_EC_DRBG is not really a NOBUS style backdoor that is permanent. The NOBUS assumption by NSA relies on the backdoor not being reconfigured by another adversary, and by selecting this construction, the NSA helped at least one foreign intelligence service to make a trivial change to a target's system, and then the backdoor could be exploited by a completely different party, and likely not the NSA after that point. It is speculated that the United States Office of Personnel Management (OPM) was hacked with assistance from this series of Juniper changes. It may be the most spectacular example of a NOBUS backdoor failing but it is not the only one. When we consider the ideas of those who advocate for NOBUS backdoors in public and in private, we must consider that every NOBUS backdoor may include using bugdoor strategies to implement the backdoor, and it may not actually be exclusively exploitable by the designers at the NSA. The Juniper example raises many questions – who made those changes in the Juniper source code control process? Who was responsible for reviewing those change? How was it that these changes were deployed to customers including some of the most sensitive US government networks? At least two answers have become clear from studying BULLRUN and related documents: the NSA and other agencies send spies to work undercover for targeted companies, and high profile companies hire former spies who continue to have lifetime obligations related to their clearances. In some cases the latter happens because the spies retain their knowledge and are able to make differently informed decisions as long as they do not have to explain why they made these choices, they are not liable to be prosecuted under the law. This is a strong contrast with Snowden whose crime again appears to be that he informed the only party who was not aware of these games: the American public.

Consider another more recent example: the NSA was long rumored to have placed a backdoor into the Philips PX-1000 in the early 1980s. The PX-1000 was an early handheld messaging device. The backdoor has now been confirmed by research performed in the open as documented [Cry22] by the Crypto Museum in Eindhoven. The original Philips PX-1000 deployed DES, and later the original model was replaced by the PX-1000Cr what is now known as the PX-1000Cr algorithm. The suspect PX-1000Cr cipher uses a linear-feedback shift register (LFSR) design and it is unquestionably weaker than DES. It has been long suspected that this backdoor may have been inserted into this specific device in an attempt to spy on anti-apartheid activists and other supporters of Nelson Mandela. One of the key leaders of the Dutch anti-apartheid movement at the time, Conny Braam, describes [Bra92,Bra04] discussing the new changes to the PX-1000 with an insider from Philips. Evidence that there was at least one concerned insider at the time suggests that this was unlikely to have been a unilateral action by NSA done in secret. Was this an attempt at a NOBUS backdoor? If so, it is an even worse failure than the DUAL_EC_DRBG backdoor. The Crypto Museum reported [Cry22] on the break [Mar22] by Stefan Marsiske: "*With just 17 characters of ciphertext, Stef can fully recover the encryption key and break any PX-1000Cr message that was sent on that key, in just 4 seconds on a regular laptop in a single thread.*" This tells us that the break by Marsiske was likely something that NSA was able to perform one way or another in the early 1980s. Luckily, the anti-apartheid movement in South Africa was aware of these issues, and as a result, Tim Jenkin [Jen87] helped to build [GE07,Jen95] what the NSA calls *indigenous cryptography* without an NSA backdoor. Jenkin's work helped to secure the communications needs of the anti-apartheid movement and the African National Congress (ANC) leadership. It is interesting to note that both choices, the DES cipher and the PX-1000Cr cipher, appear as bad choices: both were weakened by NSA. The PX-1000cr cipher weakness seems to imply that the working definition of NOBUS is actually even weaker than previously realized with DUAL_EC. If NSA was behind the sabotage of the PX-1000cr cipher as is indicated by the evidence and if it is an example of a NOBUS style backdoor, we are looking at breaks of NOBUS backdoors within a single human lifetime, rather than never. Hardly an example of "*Nobody but the US(A)*".

The NSA has shown themselves time and time again to be structurally inclined towards sabotaging cryptography in their own interests; why do so many security, privacy, and cryptography professionals care to allow a known malicious actor into a security process? Often they have no choice if they want to follow national standards.

Other times it is because the users are not aware of using software created in secret by the NSA and the software is written to appeal to a specific community of interest. The backdoor is built in a way that can be exploited by parties other than NSA [27]. Consider the National Institute of Standards and Technology (NIST) from the United States of America as an example. NIST is required by law to consult with NSA according to the public comments of NIST representatives. When NIST is confronted at cryptography conferences about this relationship the NIST spokesperson will explain that they have to follow the law. Some NIST employees openly state that they avoid speaking about the NSA relationship with NIST out of fear of issues surrounding their carrying a US government security clearance.

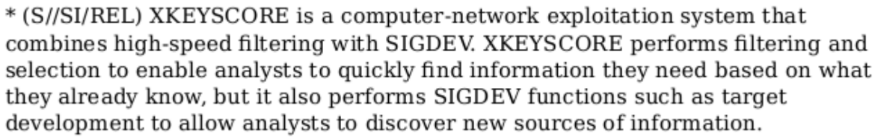
While Rogaway and Assange point to funding, we additional want to raise a question of *which laws* exactly? Should this relationship with an obvious top down order not be suspect to the cryptographic community? Why does anyone consider NIST to be better than NSA? Does anyone seriously believe that *all the involved NIST personnel with US government security clearances* were simply tricked by NSA? If they are so easy to trick, why do we suppose this will not repeat? If they are not being tricked, why should anyone trust them? Is it possible that good, safe, secure cryptographic systems come out from a possibly rigged process? An open and public review must be performed to restore trust in NIST. If this does not happen, the NIST cryptography team should be defunded and a non-captured organization with a focus on transparency should replace it.

As of the time of publishing this thesis, standardization of Post-Quantum Cryptography is underway as part of a NIST competition. An academic researcher has just broken [Beu22] an important Post-Quantum candidate, Rainbow. The paper title says it all: "*Breaking Rainbow Takes a Weekend on a Laptop*" and one wonders, did NSA not think of this or a better attack? There is an indication that NSA expressed concern with Rainbow; what was the cause of that concern? Was it merely to look as if they, too, had found a break? If NSA thought about Rainbow and they knew they could break it, why does the public need to learn about an actual break on Rainbow through the kindness of a brilliant academic? Does the NSA not employ the largest number of mathematicians in the world? The dual roles of NSA are the obvious answer. NSA from evidence is willing to allow broken cryptographic systems to be formulated, standardized, and used because it is still in their interests as a surveillance adversary. Anyone working on the defensive security side of NSA is acting, probably unintentionally, as a fig leaf for NSA's spying activities [28].

---

[27] Example from the Snowden Archive of an as of yet unreleased backdoor in fielded software that is most certainly not an exclusively exploitable backdoor by NSA. The software's secret key generation is sabotaged by design to ensure surveillance of the community of interest. There is a corresponding XKeyscore rule that has not yet been published. The goal of that rule is to gather up all ciphertext using this sabotaged system; it is clearly part of a larger strategy. As a flag in the ground for later, the thesis author presents the following SHA256 hash: `38b471eef9a87270087aac49fecbf549970dd5647a394f17ded8018b7c873c32`. There are additional examples from other sources that this is the general shape of the game being played with more than a few acts of sabotage by the NSA.

4.5 — XKeyscore



* (S//SI/REL) XKEYSCORE is a computer-network exploitation system that combines high-speed filtering with SIGDEV. XKEYSCORE performs filtering and selection to enable analysts to quickly find information they need based on what they already know, but it also performs SIGDEV functions such as target development to allow analysts to discover new sources of information.

Figure 4.26: (S//SI/REL) XKEYSCORE definition
Courtesy of NSA [Uni13b].

XKeyscore [Gre13d, Gal14b] is a distributed and generally programmable mass surveillance and exploitation system. Programs are written in a language referenced as XKS and GENESIS in internal NSA documents. XKS allows for embedding C, C++, Python, and other programming languages in XKS rules. In 2014, source code was published, as seen in Listing 4.1 by a team including the author of this thesis that shows [AGG+14b] the capabilities of XKeyscore targeting relays, Tor Directory Authorities, and Tor users [29]. Additional XKS rules show the targeting of other cryptographic systems such as the use of stegonography and data streams that appears to contain high entropy.

```
// START_DEFINITION
/**
```

---

[28] Quit already but before you go: collect evidence about misbehavior, safely and carefully talk to journalists about any concerns, and provide evidence to ensure that readers will be able to evaluate the evidence themselves. The public, who is interested, will make better decisions to protect our Republic if they are informed of what is really happening inside of NSA. Consider following Ellsberg's example. There are those who disagree and their disrespect for democratic decision making is evident by their support for secrecy despite ample evidence of malfeasance by NSA. This is to say nothing of the CIA's activities of which there is also ample evidence of malfeasance.

[29] Several NSA officials directly informed the author of this thesis that Tor traffic has been and continues to be collected domestically and internationally under a classified interpretation of Executive Order 12333 (EO12333). The justification is that some terrorists may use Tor, and so, it was explained that this makes it fair game for collection. Despite the reporting on the XKeyscore rules targeting Tor users and the Tor network in 2014 [AGG+14b], the Tor Project has not taken meaningful steps to confront the privacy threat posed to Tor users and operators by XKeyscore or similar systems. During our reporting in 2014, an employee and a contractor from the Tor Project came to our office. During their short and unpleasant visit, they argued that we should not publish the XKeyscore source code used to target Tor users and the Tor network. We felt it was clearly in the public interest, and they could not dismiss the public interest benefits with reason but pleaded with us to suppress the reporting anyway. They expressed concern that to publish the XKeyscore rules would scare people away from using Tor. This was their primary concern. When we consider Tor's original threat model, Tor clearly is not intended to protect against the NSA's capabilities. The request to suppress aspects of our factual reporting reflects the Tor Project's desire and willingness to be perceived as a tool that does protect against the NSA, despite their threat model saying the opposite. It is wrong to suppress and mislead the public about Tor protecting users from the NSA. Perhaps Tor will address the gap in their threat model with the reality of currently known adversary capabilities. Tor's threat model is much weaker than most people realize in that it does not even require a global adversary in many cases to break anonymity.

```
 * Fingerprint Tor authoritative directories enacting the directory protocol.
 */
fingerprint(’anonymizer/tor/node/authority’) = $tor_authority
  and ($tor_directory or preappid(/anonymizer\/tor\/directory/));
// END_DEFINITION

// START_DEFINITION
/*
Global Variable for Tor foreign directory servers. Searching for potential Tor
clients connecting to the Tor foreign directory servers on ports 80 and 443.
*/

$tor_foreign_directory_ip = ip(’193.23.244.244’ or ’194.109.206.212’ or
’86.59.21.38’ or ’213.115.239.118’ or ’212.112.245.170’) and port (’80’ or
’443’);
// END_DEFINITION

// START_DEFINITION
/*
this variable contains the 3 Tor directory servers hosted in FVEY countries.
Please do not update this variable with non-FVEY IPs. These are held in a
separate variable called $tor_foreign_directory_ip. Goal is to find potential
Tor clients connecting to the Tor directory servers.
*/
$tor_fvey_directory_ip = ip(’128.31.0.39’ or ’216.224.124.114’ or
’208.83.223.34’) and port (’80’ or ’443’);
// END_DEFINITION


// START_DEFINITION
requires grammar version 5
/**
 * Identify clients accessing Tor bridge information.
 */
fingerprint(’anonymizer/tor/bridge/tls’) =
ssl_x509_subject(’bridges.torproject.org’) or
```

```
ssl_dns_name(’bridges.torproject.org’);

/**
 * Database Tor bridge information extracted from confirmation emails.
 */
fingerprint(’anonymizer/tor/bridge/email’) =
email_address(’bridges@torproject.org’)
  and email_body(’https://bridges.torproject.org/’ : c++
  extractors: {{
    bridges[] = /bridge\s([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}):?([0-9]{2,4}?[^0-9])/;
  }}
  init: {{
    xks::undefine_name("anonymizer/tor/torbridges/emailconfirmation");
  }}
  main: {{
    static const std::string SCHEMA_OLD = "tor_bridges";
    static const std::string SCHEMA_NEW = "tor_routers";
    static const std::string FLAGS = "Bridge";
    if (bridges) {
      for (size_t i=0; i < bridges.size(); ++i) {
        std::string address = bridges[i][0] + ":" + bridges[i][1];
        DB[SCHEMA_OLD]["tor_bridge"] = address;
        DB.apply();
        DB[SCHEMA_NEW]["tor_ip"] = bridges[i][0];
        DB[SCHEMA_NEW]["tor_port_or"] = bridges[i][1];
        DB[SCHEMA_NEW]["tor_flags"] = FLAGS;
        DB.apply();
      }
      xks::fire_fingerprint("anonymizer/tor/directory/bridge");
    }
    return true;
  }});
// END_DEFINITION


// START_DEFINITION
/*
The fingerprint identifies sessions visiting the Tor
```

```
Project website from
non-fvey countries.
*/
fingerprint(’anonymizer/tor/torpoject_visit’)= http_host(’www.torproject.org’)
and not(xff_cc(’US’ OR ’GB’ OR ’CA’ OR ’AU’ OR ’NZ’));
// END_DEFINITION


// START_DEFINITION
/*
These variables define terms and websites relating to the TAILs (The Amnesic
Incognito Live System) software program, a comsec mechanism advocated by
extremists on extremist forums.
*/

$TAILS_terms=word(’tails’ or ’Amnesiac Incognito Live System’) and word(’linux’
or ’ USB ’ or ’ CD ’ or ’secure desktop’ or ’ IRC ’ or ’truecrypt’ or ’ tor ’);
$TAILS_websites=(’tails.boum.org/’) or (’linuxjournal.com/content/linux*’);
// END_DEFINITION

// START_DEFINITION
/*
This fingerprint identifies users searching for the TAILs (The Amnesic
Incognito Live System) software program, viewing documents relating to TAILs,
or viewing websites that detail TAILs.
*/
fingerprint(’ct_mo/TAILS’)=
fingerprint(’documents/comsec/tails_doc’) or web_search($TAILS_terms) or
url($TAILS_websites) or html_title($TAILS_websites);
// END_DEFINITION


// START_DEFINITION
requires grammar version 5
/**
 * Aggregate Tor hidden service addresses seen in raw traffic.
 */
```

```
mapreduce::plugin('anonymizer/tor/plugin/onion') =
  immediate_keyword(/(?:([a-z]+):\/\/){0,1}([a-z2-7]{16})\.onion(?::(\d+)){0,1}/c : c++
    includes: {{
      #include <boost/lexical_cast.hpp>
    }}
    proto: {{
      message onion_t {
        required string address = 1;
        optional string scheme = 2;
        optional string port = 3;
      }
    }}
    mapper<onion_t>: {{
      static const std::string prefix = "anonymizer/tor/hiddenservice/address/";

      onion_t onion;
      size_t matches = cur_args()->matches.size();
      for (size_t pos=0; pos < matches; ++pos) {
        const std::string &value = match(pos);
        if (value.size() == 16)
          onion.set_address(value);
        else if(!onion.has_scheme())
          onion.set_scheme(value);
        else
          onion.set_port(value);
      }

      if (!onion.has_address())
        return false;

      MAPPER.map(onion.address(), onion);
      xks::fire_fingerprint(prefix + onion.address());
      return true;
    }}
    reducer<onion_t>: {{
      for (values_t::const_iterator iter = VALUES.begin();
          iter != VALUES.end();
          ++iter) {
        DB["tor_onion_survey"]["onion_address"] = iter->address() + ".onion";
        if (iter->has_scheme())
          DB["tor_onion_survey"]["onion_scheme"] = iter->scheme();
        if (iter->has_port())
```

```
          DB["tor_onion_survey"]["onion_port"] = iter
             ->port();
        DB["tor_onion_survey"]["onion_count"] = boost
           ::lexical_cast<std::string>(
           TOTAL_VALUE_COUNT);
        DB.apply();
        DB.clear();
      }
      return true;
    }});

/**
 * Placeholder fingerprint for Tor hidden service
    addresses.
 * Real fingerpritns will be fired by the plugins
 *   'anonymizer/tor/plugin/onion/*'
 */
fingerprint('anonymizer/tor/hiddenservice/address') =
   nil;
// END_DEFINITION


// START_DEFINITION
appid('anonymizer/mailer/mixminion', 3.0, viewer=
   $ascii_viewer) =
        http_host('mixminion') or
        ip('128.31.0.34');
// END_DEFINITION
```

Listing 4.1: "XKeyscore source code published [AGG+14b] by NDR as xkeyscorerules100.txt"

In the source code listing 4.1 [30] we see a number of features including the ability to generally perform search over network traffic as well as the ability to store data extracted from the network into databases. The geographic location of systems is important to the authors of the rules. XKeyscore is able to select traffic based on hostname visibility into network flows. The emails sent by the Tor Bridge authority intercepted by XKeyscore and then the Tor bridge data is extracted and stored in a database. Visitors to the Tor Project website are identified as long as the user's IP does not originate in the United States, United Kingdom, Canada, Australia, or New Zealand. Similar flagging is done to would be users of the Tails Amnesiac Incognito Live System, or the rather mundane Linux Journal website. If someone should ever transmit a Tor Onion service address over the network unencrypted, the XKeyscore rule *mapreduce::plugin('anonymizer/tor/plugin/onion)* will automatically collect the candidate onion address and store the result in a database. An unscrupulous adversary might flood known XKeyscore collection points with onion addresses to fill that database with random addresses or in an attempt to elicit a connection from the NSA, they may send real onion addresses which are selectively released as honey tokens. The final rule shows a rule targeting a mixnet known as Mixminion. The *appid('anonymizer/mailer/mixminion',...)* rule is confounding as it seems to select all Mixminion traffic with the host *mixminion* or the IP address *128.31.0.34* for surveillance. Is that level of monitoring enough to break Mixminion? Is the goal of this XKS rule simply to enumerate all end user IP addresses? The collection of end user IP addresses seems to create an avenue for investigation that avoids the need to break the anonymity provided by Mixminion.

---

[30] The author's personal computer `208.83.223.34` is listed in this source code listing as it was a Tor Directory Authority at the time.

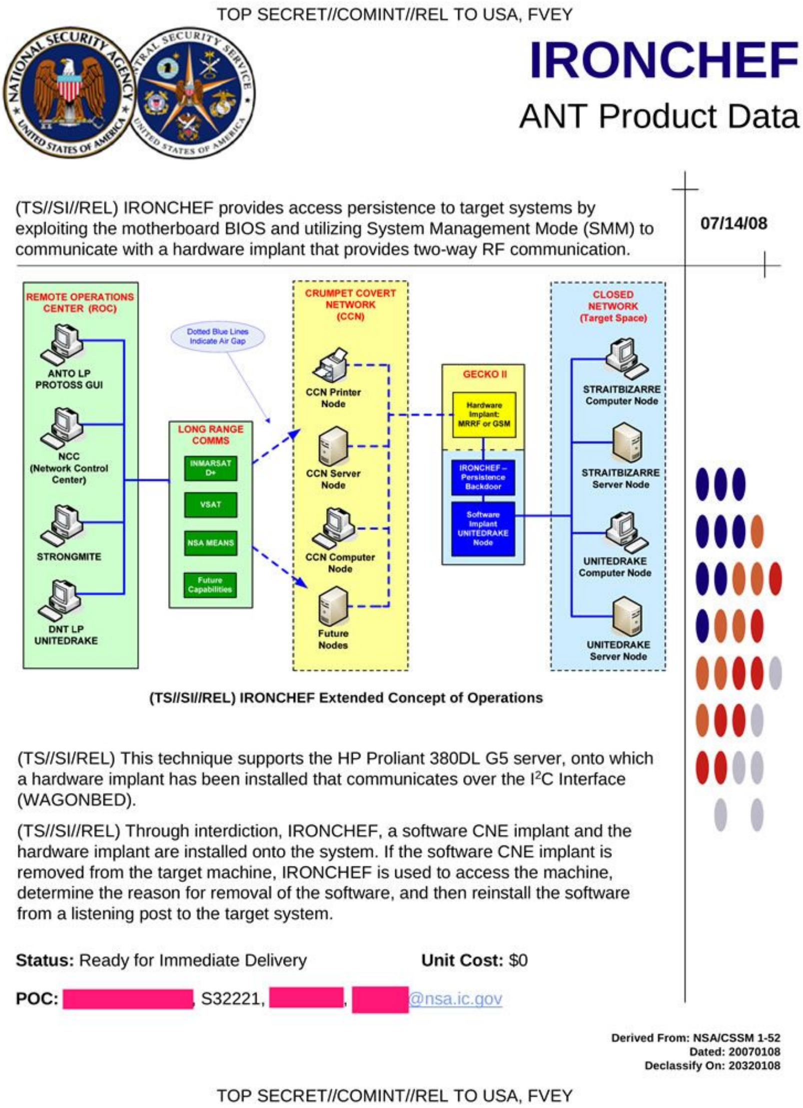
4.6 — ANT catalog

The ANT catalog [AHS13] is a fifty-page set of forty-nine classified documents that were published in late 2013 by Der Spiegel. For each item, we include the document in question, and a brief analysis. A number of referenced cover names are not directly documented. We have carefully studied related documents to define each term. When possible, we also cite related works that similarly analyze documents included in this thesis and other related publications. We have included the full documents as published by Der Spiegel, and we have similarly organized them thematically. The documents are included in full to ensure that they are properly archived. Many of these tools and capabilities have certainly been improved, and in some cases we find that there are now commercialized products that serve the same goals.

To help understand the ANT catalog, we turn to additional leaked sources to define otherwise obscure cover and code names that are used in the catalog. In 2016, a group known as the Shadowbrokers [Wik21r] began releasing NSA documents, first publishing a free sample and then offering other material at what they called the Equation Group auction. This refers to the NSA's Equation Group (EQGRP), which was later linked to NSA's Tailored Access Operations (TAO). Next, between 2016 and 2017, the Shadowbrokers released at least five compressed, encrypted archives which contain various programs for hacking and surveillance. The published documents include exploits, post-exploitation payloads, persistent backdoor implants, documentation about payloads and tactics, notes, and other miscellaneous documents. Most noteworthy was the NSA program known as ETERNALBLUE. Reportedly, this exploit was used widely to compromise hundreds of thousands of computers around the world. All five releases have been widely mirrored on the Internet and the contents have been the subject of industry discussion. All of these leaked documents quickly fell into the hands of extremely skilled as well as unskilled people, causing untold damage to the effectiveness of the NSA's high-quality exploits.

The ANT catalog entries that follow are grouped into thematic categories as they were originally published by Der Spiegel: servers, firewalls, routers, room surveillance, varied radio surveillance, wireless LAN, cellular telephony, computer display surveillance, keyboard surveillance, computer system implants, and USB devices.

Figure 4.27: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

IRONCHEF

ANT Product Data

(TS//SI//REL) IRONCHEF provides access persistence to target systems by exploiting the motherboard BIOS and utilizing System Management Mode (SMM) to communicate with a hardware implant that provides two-way RF communication.

07/14/08

REMOTE OPERATIONS CENTER (ROC)
ANTO LP PROTOSS GUI
NCC (Network Control Center)
STRONGMITE
DNT LP UNITEDRAKE
Dotted Blue Lines Indicate Air Gap
LONG RANGE COMMS
INMARSAT D+
VSAT
NSA MEANS
Future Capabilities
CRUMPET COVERT NETWORK (CCN)
CCN Printer Node
CCN Server Node
CCN Computer Node
Future Nodes
GECKO II
Hardware Implant: MRRF or GSM
IRONCHEF – Persistence Backdoor
Software Implant UNITEDRAKE Node
CLOSED NETWORK (Target Space)
STRAITBIZARRE Computer Node
STRAITBIZARRE Server Node
UNITEDRAKE Computer Node
UNITEDRAKE Server Node

(TS//SI//REL) IRONCHEF Extended Concept of Operations

(TS//SI/REL) This technique supports the HP Proliant 380DL G5 server, onto which a hardware implant has been installed that communicates over the I^2C Interface (WAGONBED).

(TS//SI//REL) Through interdiction, IRONCHEF, a software CNE implant and the hardware implant are installed onto the system. If the software CNE implant is removed from the target machine, IRONCHEF is used to access the machine, determine the reason for removal of the software, and then reinstall the software from a listening post to the target system.

Status: Ready for Immediate Delivery

Unit Cost: $0

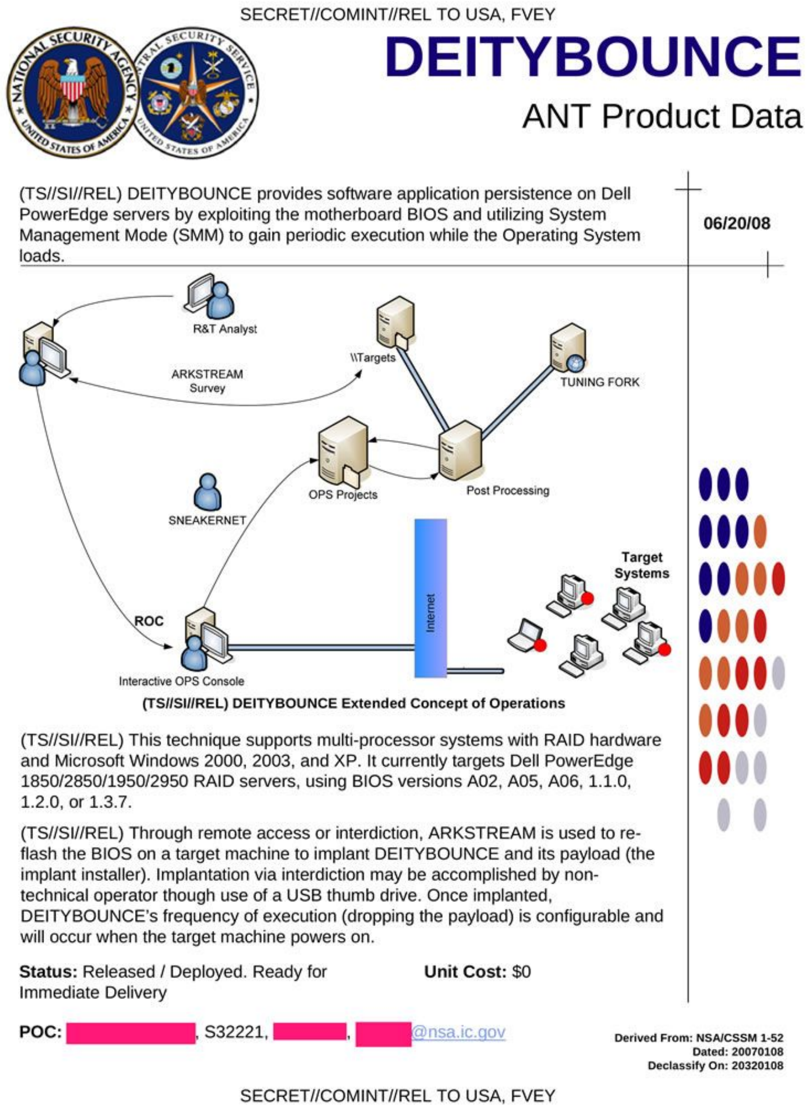
POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.1 – IRONCHEF. An Intel System Management Mode (SMM) BIOS rootkit used on system boot for backdoor persistence. It is used to deploy other payloads from the BIOS of a Hewlett-Packard Proliant 380DL G5 server. IRONCHEF is paired with a hardware implant such as WAGONBED as described in Section 4.6.18 for bridging an air-gap using the i^2c bus for communication between the BIOS, another backdoor, and the hardware backdoor. Configurable to carry a payload such as UNITEDRAKE in Section 4.6.50. The software backdoor is deployed through interdiction in tandem with the hardware backdoor. IRONCHEF survives upgrades or replacements of the operating system.

Figure 4.28: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

DEITYBOUNCE

ANT Product Data

(TS//SI//REL) DEITYBOUNCE provides software application persistence on Dell PowerEdge servers by exploiting the motherboard BIOS and utilizing System Management Mode (SMM) to gain periodic execution while the Operating System loads.

06/20/08

R&T Analyst
\\Targets
ARKSTREAM Survey
TUNING FORK
OPS Projects
Post Processing
SNEAKERNET
Target Systems
Internet
ROC
Interactive OPS Console

(TS//SI//REL) DEITYBOUNCE Extended Concept of Operations

(TS//SI//REL) This technique supports multi-processor systems with RAID hardware and Microsoft Windows 2000, 2003, and XP. It currently targets Dell PowerEdge 1850/2850/1950/2950 RAID servers, using BIOS versions A02, A05, A06, 1.1.0, 1.2.0, or 1.3.7.

(TS//SI//REL) Through remote access or interdiction, ARKSTREAM is used to re-flash the BIOS on a target machine to implant DEITYBOUNCE and its payload (the implant installer). Implantation via interdiction may be accomplished by non-technical operator though use of a USB thumb drive. Once implanted, DEITYBOUNCE's frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0

POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

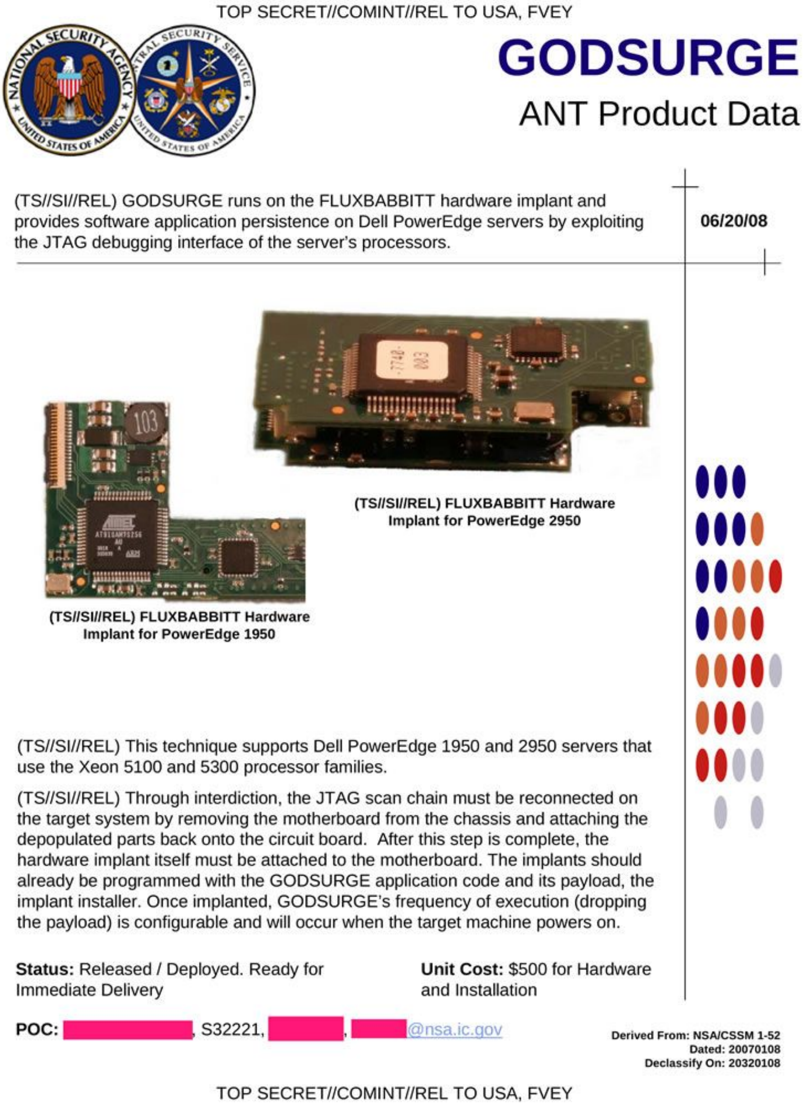
Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.2 – DEITYBOUNCE. An Intel System Management Mode (SMM) rootkit for the BIOS of Dell PowerEdge servers with a configurable frequency of execution for inserting and running additional payloads during operating system (OS) startup. The rootkit is deployed through remote exploitation or interdiction with ARKSTREAM. See also SWAP in Section 4.6.39 for additional uses of ARKSTREAM.

Dell claimed that they have a policy of non-cooperation [McC13] which rejects adding backdoors to their systems after the release of this document.

Figure 4.29: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

GODSURGE

ANT Product Data

(TS//SI//REL) GODSURGE runs on the FLUXBABBITT hardware implant and provides software application persistence on Dell PowerEdge servers by exploiting the JTAG debugging interface of the server's processors.

06/20/08

(TS//SI//REL) FLUXBABBITT Hardware Implant for PowerEdge 2950

(TS//SI//REL) FLUXBABBITT Hardware Implant for PowerEdge 1950

(TS//SI//REL) This technique supports Dell PowerEdge 1950 and 2950 servers that use the Xeon 5100 and 5300 processor families.

(TS//SI//REL) Through interdiction, the JTAG scan chain must be reconnected on the target system by removing the motherboard from the chassis and attaching the depopulated parts back onto the circuit board. After this step is complete, the hardware implant itself must be attached to the motherboard. The implants should already be programmed with the GODSURGE application code and its payload, the implant installer. Once implanted, GODSURGE's frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $500 for Hardware and Installation

POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

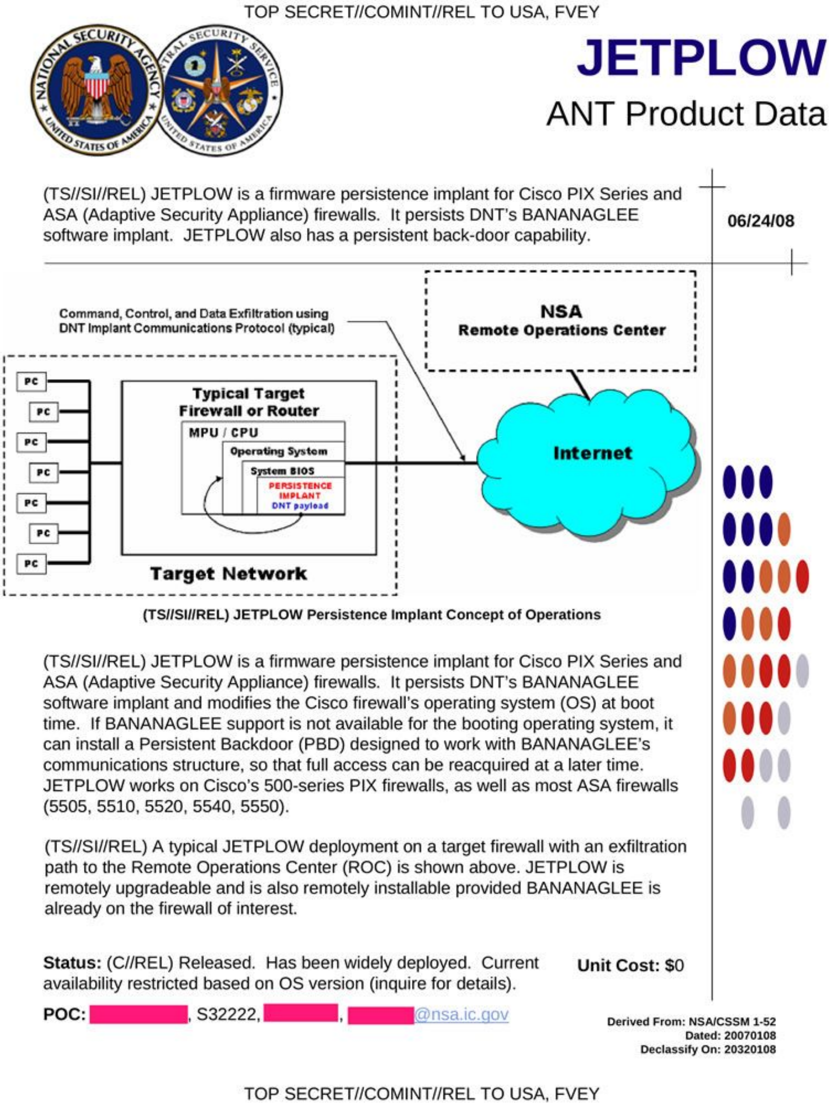
Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.3 – GODSURGE. GODSURGE is a software implant used in tandem with the FLUXBABBITT hardware implant module. FLUXBABBITT and GODSURGE specifically target Dell PowerEdge server models 1950 and 2950 that used Xeon 5100 and 5300 processor series as the CPU. These CPUs have public documented JTAG interfaces which are exploited for persistent control of the system before an OS takes control of the boot process. Using the FLUXBABBITT hardware platform, GODSURGE exploits the JTAG interface to target Dell PowerEdge server CPUs. GODSURGE uses the JTAG interface to install a persistent rootkit on a targeted system. The FLUXBABBITT JTAG hardware backdoor is installed through physical access to the system by means of interdiction [Wik21j] or black bag [Wik21c] operations.

Dell claims that they have a policy of non-cooperation [McC13] which rejects adding backdoors to their systems.

Figure 4.30: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

JETPLOW

ANT Product Data

(TS//SI//REL) JETPLOW is a firmware persistence implant for Cisco PIX Series and ASA (Adaptive Security Appliance) firewalls. It persists DNT's BANANAGLEE software implant. JETPLOW also has a persistent back-door capability.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Internet
Target Network

(TS//SI//REL) JETPLOW Persistence Implant Concept of Operations

(TS//SI//REL) JETPLOW is a firmware persistence implant for Cisco PIX Series and ASA (Adaptive Security Appliance) firewalls. It persists DNT's BANANAGLEE software implant and modifies the Cisco firewall's operating system (OS) at boot time. If BANANAGLEE support is not available for the booting operating system, it can install a Persistent Backdoor (PBD) designed to work with BANANAGLEE's communications structure, so that full access can be reacquired at a later time. JETPLOW works on Cisco's 500-series PIX firewalls, as well as most ASA firewalls (5505, 5510, 5520, 5540, 5550).

(TS//SI//REL) A typical JETPLOW deployment on a target firewall with an exfiltration path to the Remote Operations Center (ROC) is shown above. JETPLOW is remotely upgradeable and is also remotely installable provided BANANAGLEE is already on the firewall of interest.

Status: (C//REL) Released. Has been widely deployed. Current availability restricted based on OS version (inquire for details).

Unit Cost: $0

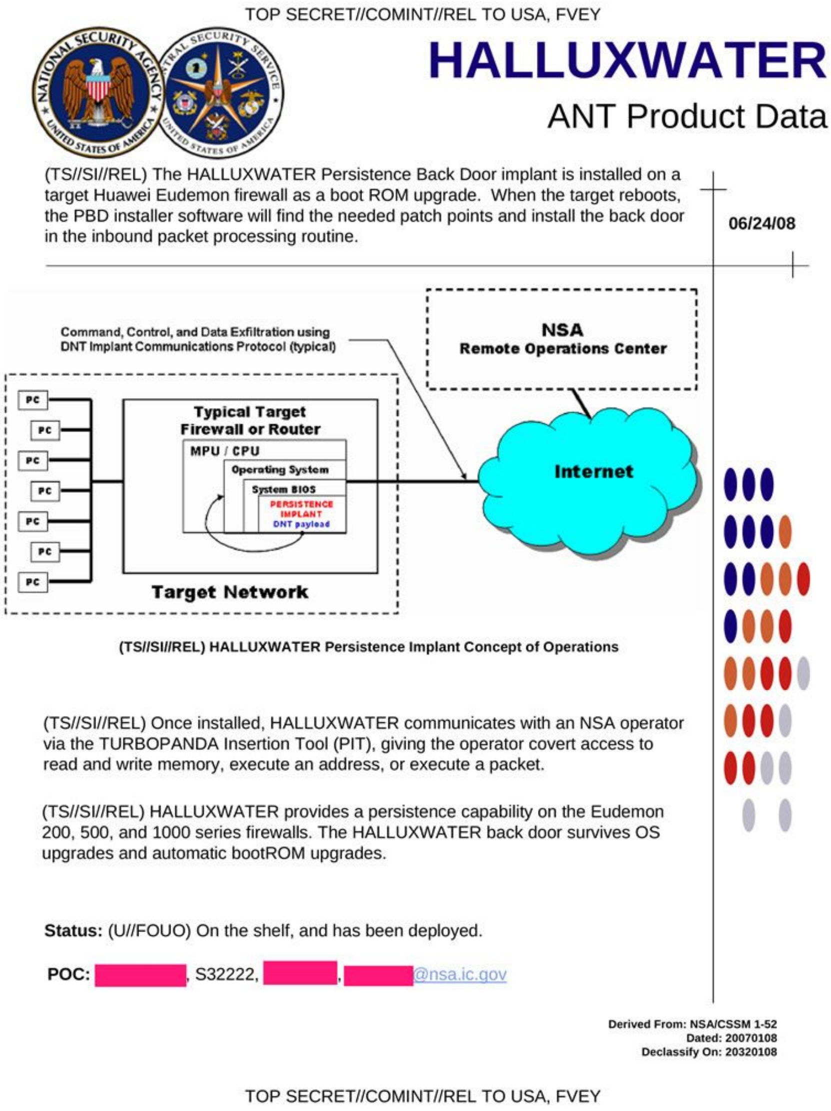
POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.4 – JETPLOW. A firmware resident persistent backdoor for Cisco PIX series and Adaptive Security Appliance (ASA) Firewalls that dynamically backdoors the OS at boot time. JETPLOW is used to deploy other payloads such as BANANAGLEE as mentioned in Section 4.6.50. JETPLOW has a minimal rootkit capability when the targeted OS is not supported by the payload. The rootkit is deployed through remote exploitation and/or interdiction. See also SCREAMINGPLOW in Section 4.6.50 as an alternative to JETPLOW that makes BANANAGLEE a persistent backdoor.

Figure 4.31: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

HALLUXWATER

ANT Product Data

(TS//SI//REL) The HALLUXWATER Persistence Back Door implant is installed on a target Huawei Eudemon firewall as a boot ROM upgrade. When the target reboots, the PBD installer software will find the needed patch points and install the back door in the inbound packet processing routine.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Internet
Target Network

(TS//SI//REL) HALLUXWATER Persistence Implant Concept of Operations

(TS//SI//REL) Once installed, HALLUXWATER communicates with an NSA operator via the TURBOPANDA Insertion Tool (PIT), giving the operator covert access to read and write memory, execute an address, or execute a packet.

(TS//SI//REL) HALLUXWATER provides a persistence capability on the Eudemon 200, 500, and 1000 series firewalls. The HALLUXWATER back door survives OS upgrades and automatic bootROM upgrades.

Status: (U//FOUO) On the shelf, and has been deployed.

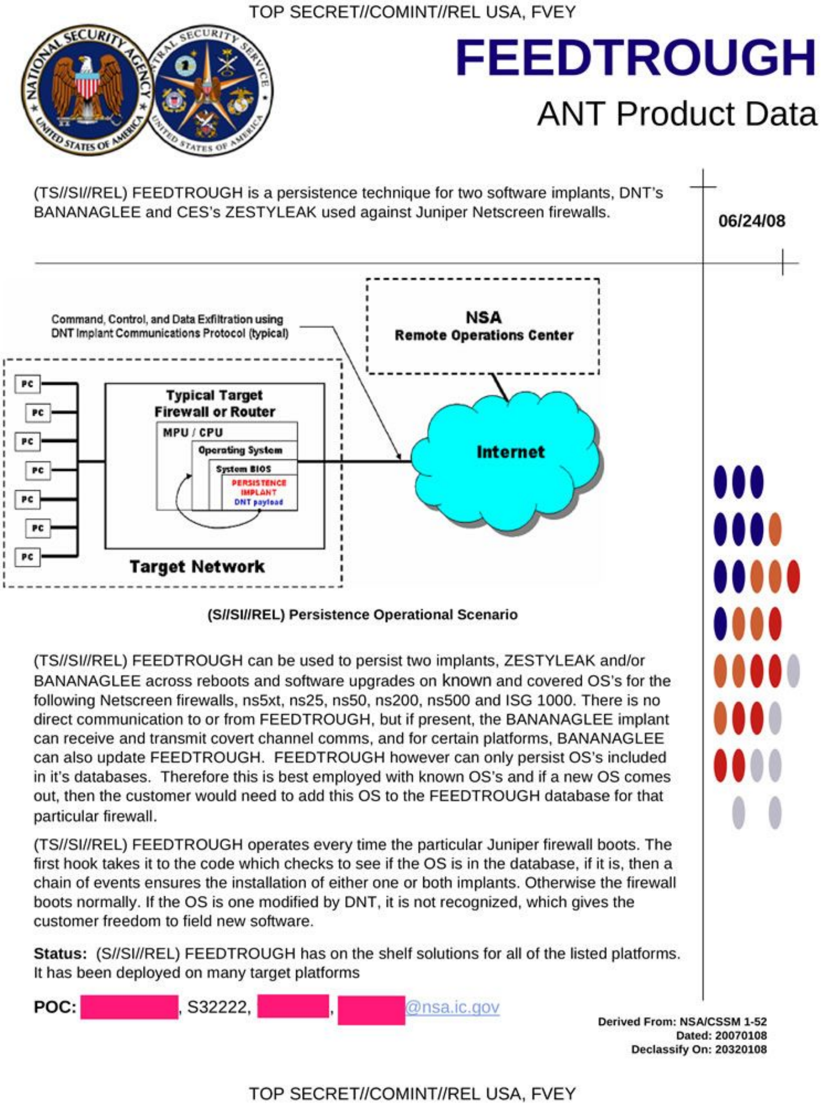
POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.5 – HALLUXWATER. A persistent rootkit executed on system boot used for covert control over Huawei Eudemon firewalls. Remote control through TURBOPANDA insertion tool as mentioned in Section 4.6.5. The rootkit is deployed through remote exploitation and/or interdiction, and survives OS or boot ROM upgrades. TURBOPANDA is a cover name for a joint project between the NSA and CIA.

Figure 4.32: Courtesy of Der Spiegel [AHS13].


Command, Control, and Data Exfiltration using
DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
Internet
PC
PC
PC
PC
PC
PC
PC
Typical Target
Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE
IMPLANT
DNT payload
Target Network

TOP SECRET//COMINT//REL USA, FVEY

FEEDTROUGH

ANT Product Data

(TS//SI//REL) FEEDTROUGH is a persistence technique for two software implants, DNT's BANANAGLEE and CES's ZESTYLEAK used against Juniper Netscreen firewalls.

06/24/08

(S//SI//REL) Persistence Operational Scenario

(TS//SI//REL) FEEDTROUGH can be used to persist two implants, ZESTYLEAK and/or BANANAGLEE across reboots and software upgrades on known and covered OS's for the following Netscreen firewalls, ns5xt, ns25, ns50, ns200, ns500 and ISG 1000. There is no direct communication to or from FEEDTROUGH, but if present, the BANANAGLEE implant can receive and transmit covert channel comms, and for certain platforms, BANANAGLEE can also update FEEDTROUGH. FEEDTROUGH however can only persist OS's included in it's databases. Therefore this is best employed with known OS's and if a new OS comes out, then the customer would need to add this OS to the FEEDTROUGH database for that particular firewall.

(TS//SI//REL) FEEDTROUGH operates every time the particular Juniper firewall boots. The first hook takes it to the code which checks to see if the OS is in the database, if it is, then a chain of events ensures the installation of either one or both implants. Otherwise the firewall boots normally. If the OS is one modified by DNT, it is not recognized, which gives the customer freedom to field new software.

Status: (S//SI//REL) FEEDTROUGH has on the shelf solutions for all of the listed platforms. It has been deployed on many target platforms

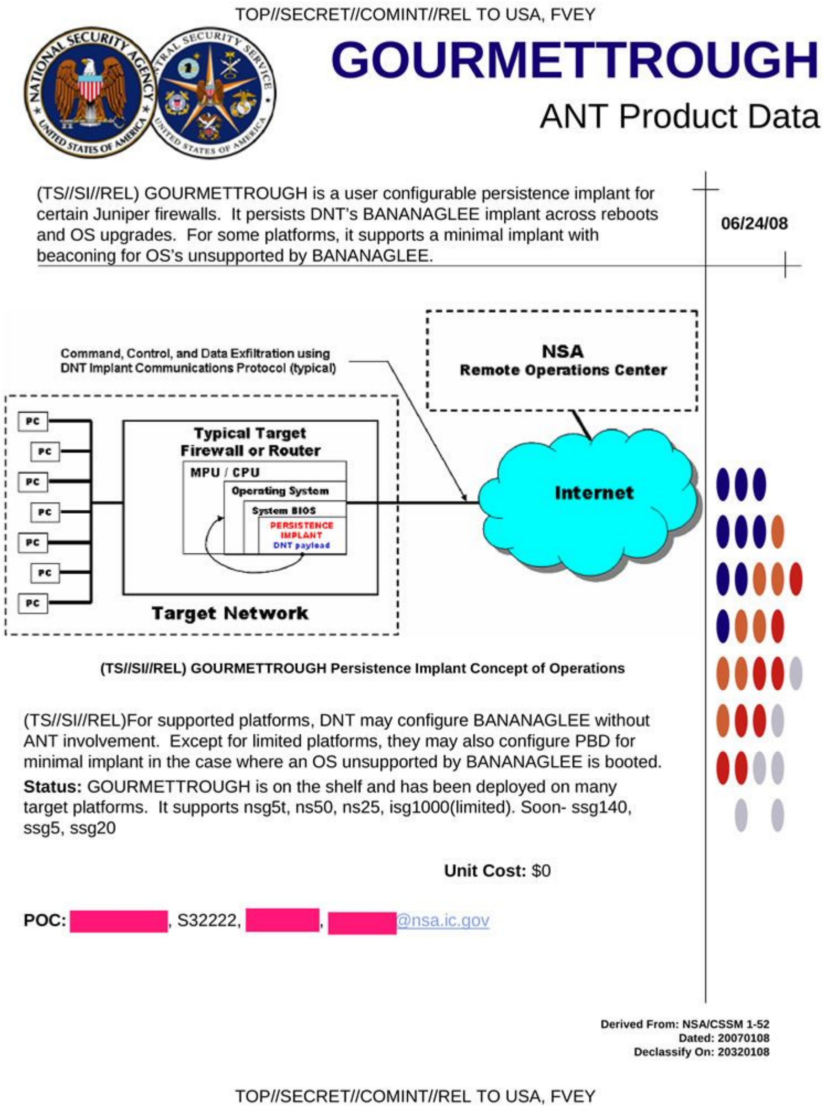
POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL USA, FVEY

4.6.6 – FEEDTROUGH. A BIOS rootkit used on system boot for persistence. Used to deploy other payloads from the BIOS of Juniper Netscreen firewalls. Configurable to carry payloads such as BANANAGLEE as mentioned in Section 4.6.50 and/or ZESTYLEAK as mentioned in Section 4.6.50. The rootkit is deployed through remote exploitation and/or interdiction.

Figure 4.33: Courtesy of Der Spiegel [AHS13].



TOP//SECRET//COMINT//REL TO USA, FVEY

GOURMETTROUGH

ANT Product Data

(TS//SI//REL) GOURMETTROUGH is a user configurable persistence implant for certain Juniper firewalls. It persists DNT's BANANAGLEE implant across reboots and OS upgrades. For some platforms, it supports a minimal implant with beaconing for OS's unsupported by BANANAGLEE.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Internet
Target Network

(TS//SI//REL) GOURMETTROUGH Persistence Implant Concept of Operations

(TS//SI//REL)For supported platforms, DNT may configure BANANAGLEE without ANT involvement. Except for limited platforms, they may also configure PBD for minimal implant in the case where an OS unsupported by BANANAGLEE is booted.

Status: GOURMETTROUGH is on the shelf and has been deployed on many target platforms. It supports nsg5t, ns50, ns25, isg1000(limited). Soon- ssg140, ssg5, ssg20

Unit Cost: $0

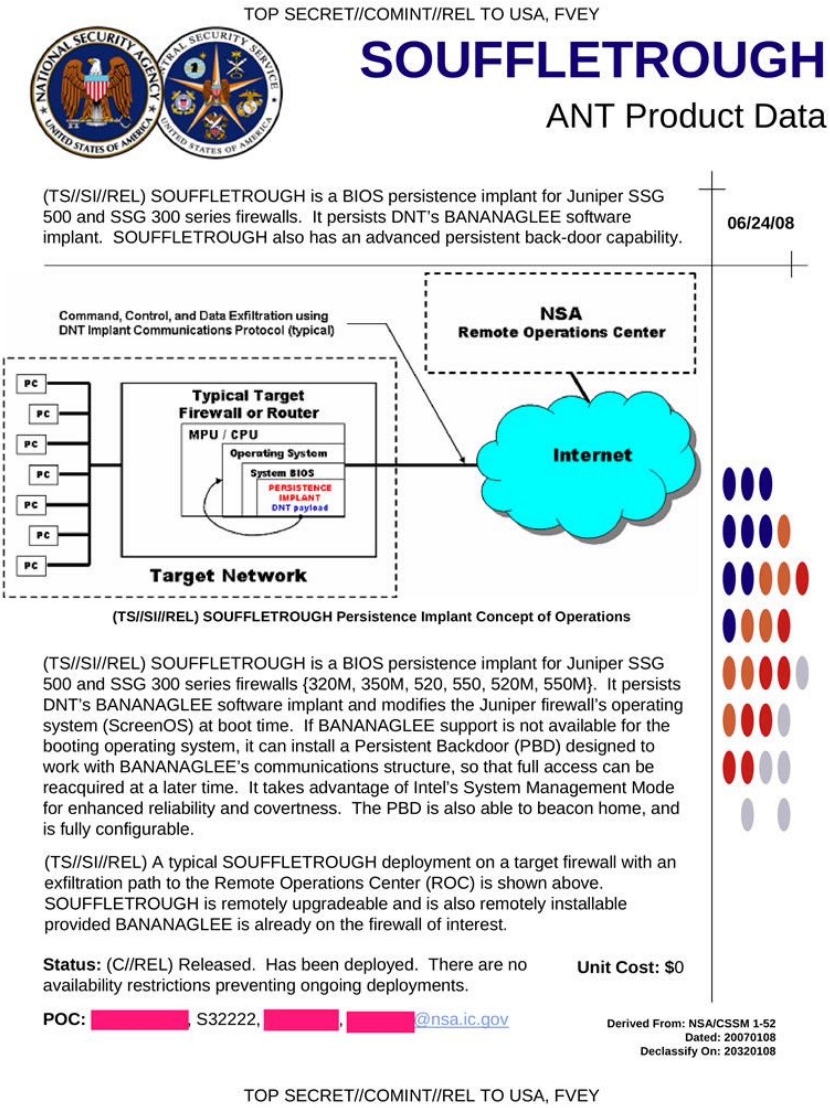
POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP//SECRET//COMINT//REL TO USA, FVEY

4.6.7 – GOURMETTROUGH. A BIOS rootkit used on system boot for persistence and used to deploy other payloads from the BIOS of Juniper Netscreen firewalls. Configurable to carry a payload such as BANANAGLEE as mentioned in Section 4.6.50. GOURMETTROUGH has a minimal rootkit capability when the targeted operating system is not supported by the payload. The rootkit is deployed through remote exploitation and/or interdiction.

Figure 4.34: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

SOUFFLETROUGH

ANT Product Data

(TS//SI//REL) SOUFFLETROUGH is a BIOS persistence implant for Juniper SSG 500 and SSG 300 series firewalls. It persists DNT's BANANAGLEE software implant. SOUFFLETROUGH also has an advanced persistent back-door capability.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Internet
Target Network

(TS//SI//REL) SOUFFLETROUGH Persistence Implant Concept of Operations

(TS//SI//REL) SOUFFLETROUGH is a BIOS persistence implant for Juniper SSG 500 and SSG 300 series firewalls {320M, 350M, 520, 550, 520M, 550M}. It persists DNT's BANANAGLEE software implant and modifies the Juniper firewall's operating system (ScreenOS) at boot time. If BANANAGLEE support is not available for the booting operating system, it can install a Persistent Backdoor (PBD) designed to work with BANANAGLEE's communications structure, so that full access can be reacquired at a later time. It takes advantage of Intel's System Management Mode for enhanced reliability and covertness. The PBD is also able to beacon home, and is fully configurable.

(TS//SI//REL) A typical SOUFFLETROUGH deployment on a target firewall with an exfiltration path to the Remote Operations Center (ROC) is shown above. SOUFFLETROUGH is remotely upgradeable and is also remotely installable provided BANANAGLEE is already on the firewall of interest.

Status: (C//REL) Released. Has been deployed. There are no availability restrictions preventing ongoing deployments.

Unit Cost: $0

POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

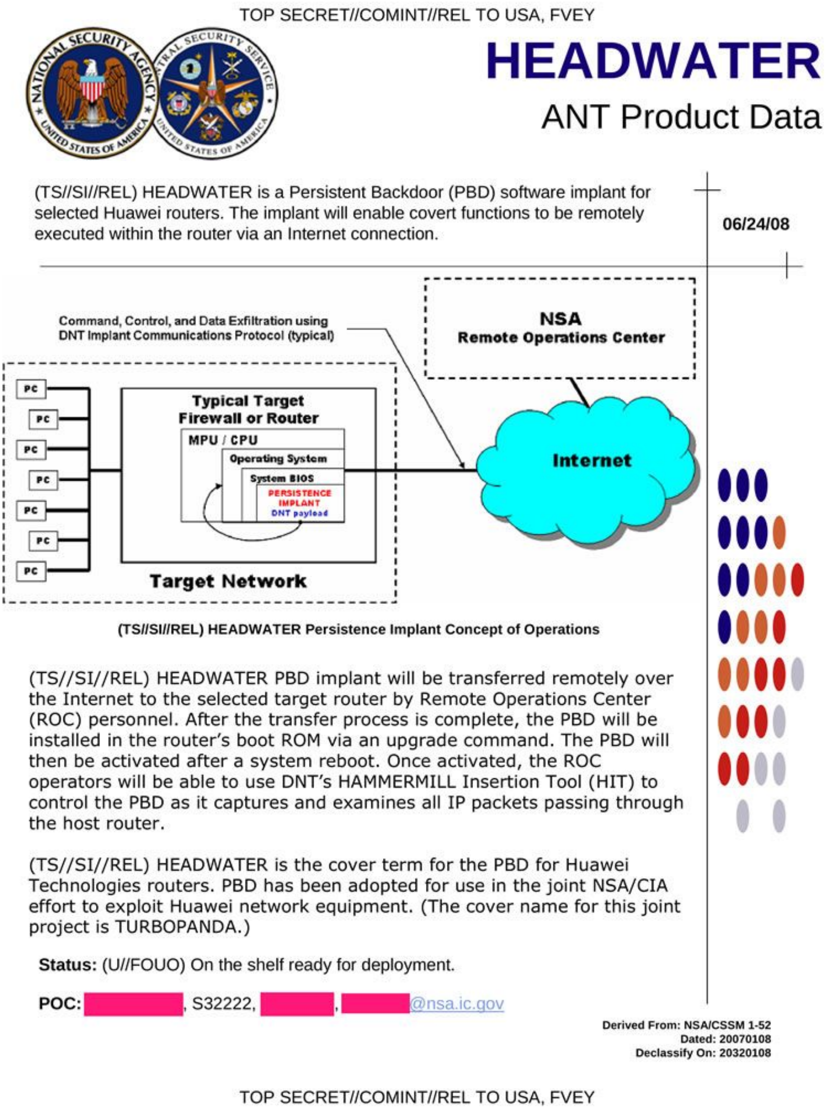
Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.8 – SOUFFLETROUGH. An Intel system management mode (SMM) BIOS rootkit used on system boot for persistence. Used to deploy other payloads from the BIOS of Juniper SSG 500 series and SSG 300 series firewalls. Configurable to carry a payload such as BANANAGLEE as mentioned in Section 4.6.50. SOUFFLETROUGH has a minimal rootkit capability when the targeted operating system is not supported by the payload. The rootkit is deployed through remote exploitation and/or interdiction.

4.6.9 – HEADWATER.

Figure 4.35: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

HEADWATER

ANT Product Data

(TS//SI//REL) HEADWATER is a Persistent Backdoor (PBD) software implant for selected Huawei routers. The implant will enable covert functions to be remotely executed within the router via an Internet connection.

06/24/08

Command, Control, and Data Exfiltration using
DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target
Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE
IMPLANT
DNT payload
Internet
Target Network

(TS//SI//REL) HEADWATER Persistence Implant Concept of Operations

(TS//SI//REL) HEADWATER PBD implant will be transferred remotely over the Internet to the selected target router by Remote Operations Center (ROC) personnel. After the transfer process is complete, the PBD will be installed in the router's boot ROM via an upgrade command. The PBD will then be activated after a system reboot. Once activated, the ROC operators will be able to use DNT's HAMMERMILL Insertion Tool (HIT) to control the PBD as it captures and examines all IP packets passing through the host router.

(TS//SI//REL) HEADWATER is the cover term for the PBD for Huawei Technologies routers. PBD has been adopted for use in the joint NSA/CIA effort to exploit Huawei network equipment. (The cover name for this joint project is TURBOPANDA.)

Status: (U//FOUO) On the shelf ready for deployment.

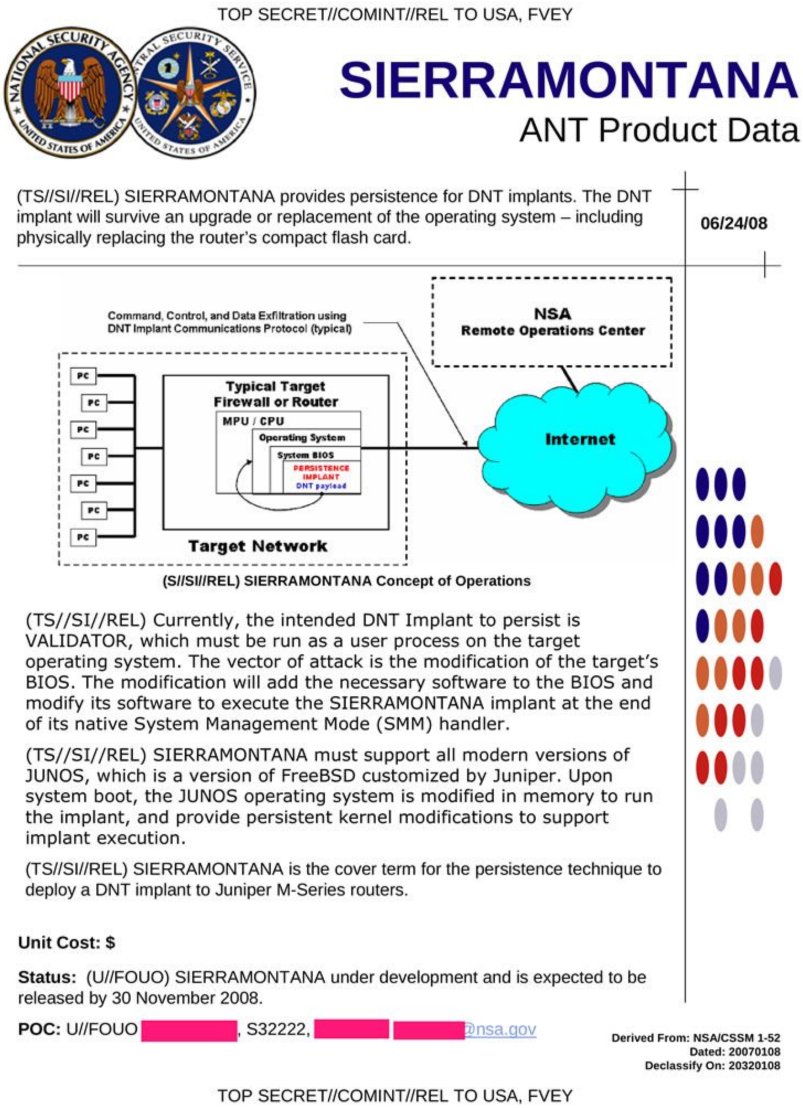
POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

A persistent rootkit executed on system boot used for covert control over Huawei routers. Remote control through HAMMERMILL as mentioned in Section 4.6.50 as HAMMERMILL Insertion Tool (HIT). The rootkit is deployed through remote exploitation and/or interdiction, and survives OS or boot ROM upgrades. HEADWATER is able to capture and process all IP packets passing through the compromised router. HEADWATER is a cover term for persistent backdoor capabilities against Huawei Technologies routers, related to HALLUXWATER as mentioned in Section 4.6.5. TURBOPANDA as mentioned in Section 4.6.5 is the cover name for a joint project between the NSA and CIA for related capabilities.

Figure 4.36: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

SIERRAMONTANA

ANT Product Data

(TS//SI//REL) SIERRAMONTANA provides persistence for DNT implants. The DNT implant will survive an upgrade or replacement of the operating system – including physically replacing the router's compact flash card.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Internet
Target Network

(S//SI//REL) SIERRAMONTANA Concept of Operations

(TS//SI//REL) Currently, the intended DNT Implant to persist is VALIDATOR, which must be run as a user process on the target operating system. The vector of attack is the modification of the target's BIOS. The modification will add the necessary software to the BIOS and modify its software to execute the SIERRAMONTANA implant at the end of its native System Management Mode (SMM) handler.

(TS//SI//REL) SIERRAMONTANA must support all modern versions of JUNOS, which is a version of FreeBSD customized by Juniper. Upon system boot, the JUNOS operating system is modified in memory to run the implant, and provide persistent kernel modifications to support implant execution.

(TS//SI//REL) SIERRAMONTANA is the cover term for the persistence technique to deploy a DNT implant to Juniper M-Series routers.

Unit Cost: $

Status: (U//FOUO) SIERRAMONTANA under development and is expected to be released by 30 November 2008.

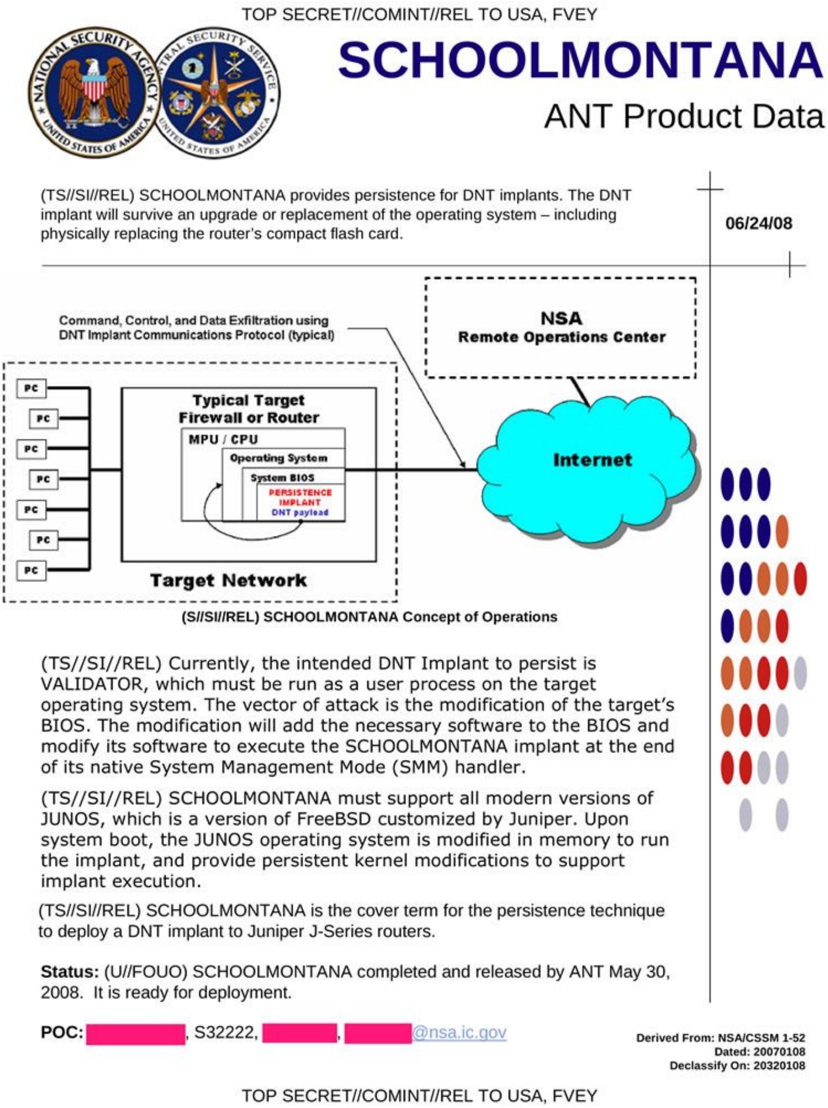
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.10 – SIERRAMONTANA. An Intel System Management Mode (SMM) BIOS rootkit used on system boot for persistence. Used to deploy other payloads from the BIOS of Juniper JUNOS M-Series routers. Configurable to carry a payload such as BANANAGLEE as mentioned in Section 4.6.50. SIERRAMONTANA has a minimal rootkit capability when the targeted operating system is not supported by the payload. The rootkit is deployed through remote exploitation and/or interdiction and survives upgrades or replacements of the OS, including replacement of the compact flash card in the router.

Figure 4.37: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

SCHOOLMONTANA

ANT Product Data

(TS//SI//REL) SCHOOLMONTANA provides persistence for DNT implants. The DNT implant will survive an upgrade or replacement of the operating system – including physically replacing the router's compact flash card.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA Remote Operations Center
Internet
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Target Network

(S//SI//REL) SCHOOLMONTANA Concept of Operations

(TS//SI//REL) Currently, the intended DNT Implant to persist is VALIDATOR, which must be run as a user process on the target operating system. The vector of attack is the modification of the target's BIOS. The modification will add the necessary software to the BIOS and modify its software to execute the SCHOOLMONTANA implant at the end of its native System Management Mode (SMM) handler.

(TS//SI//REL) SCHOOLMONTANA must support all modern versions of JUNOS, which is a version of FreeBSD customized by Juniper. Upon system boot, the JUNOS operating system is modified in memory to run the implant, and provide persistent kernel modifications to support implant execution.

(TS//SI//REL) SCHOOLMONTANA is the cover term for the persistence technique to deploy a DNT implant to Juniper J-Series routers.

Status: (U//FOUO) SCHOOLMONTANA completed and released by ANT May 30, 2008. It is ready for deployment.

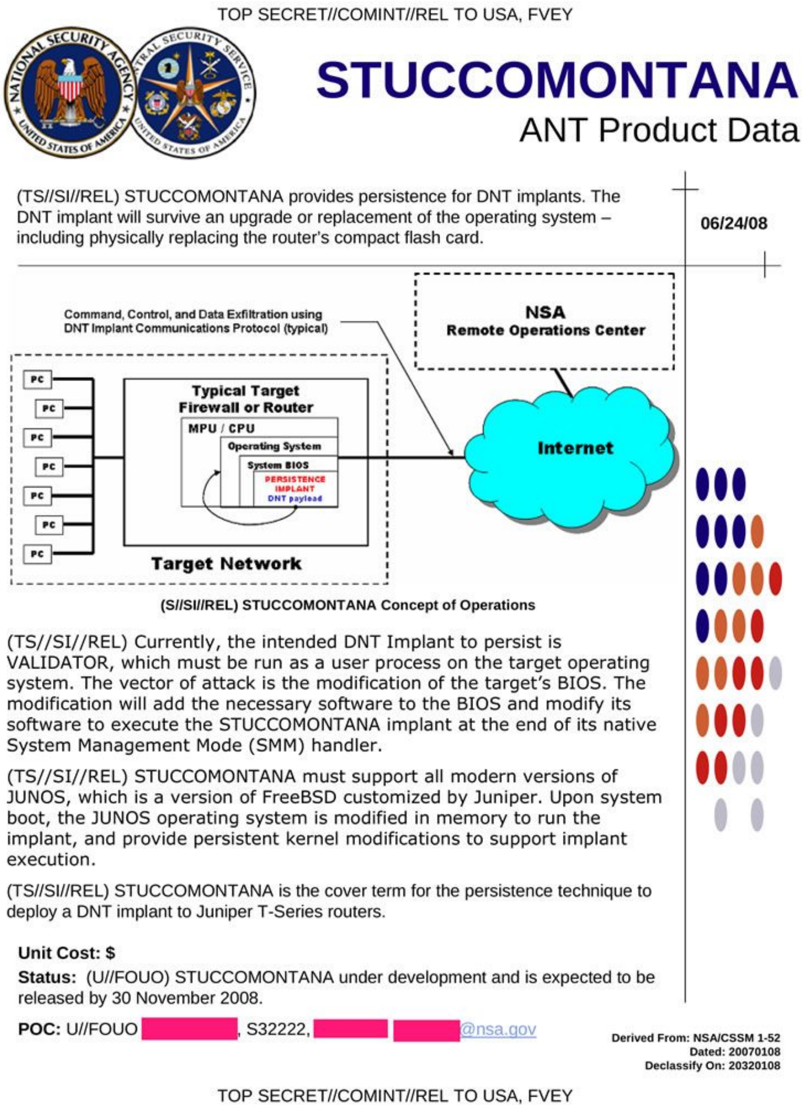
POC: [redacted], S32222, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.11 – SCHOOLMONTANA. An Intel System Management Mode (SMM) BIOS rootkit used on system boot for persistence. Used to deploy other payloads from the BIOS of Juniper JUNOS J-Series routers. Configurable to carry a payload such as VALIDATOR as mentioned in Section 4.6.50. SCHOOLMONTANA has a minimal rootkit capability when the targeted operating system is not supported by the payload. The rootkit is deployed through remote exploitation and/or interdiction and survives upgrades or replacements of the OS, including replacement of the compact flash card in the router.

Figure 4.38: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

STUCCOMONTANA

ANT Product Data

(TS//SI//REL) STUCCOMONTANA provides persistence for DNT implants. The DNT implant will survive an upgrade or replacement of the operating system – including physically replacing the router's compact flash card.

06/24/08

Command, Control, and Data Exfiltration using DNT Implant Communications Protocol (typical)
NSA
Remote Operations Center
PC
PC
PC
PC
PC
PC
PC
Typical Target Firewall or Router
MPU / CPU
Operating System
System BIOS
PERSISTENCE IMPLANT
DNT payload
Internet
Target Network

(S//SI//REL) STUCCOMONTANA Concept of Operations

(TS//SI//REL) Currently, the intended DNT Implant to persist is VALIDATOR, which must be run as a user process on the target operating system. The vector of attack is the modification of the target's BIOS. The modification will add the necessary software to the BIOS and modify its software to execute the STUCCOMONTANA implant at the end of its native System Management Mode (SMM) handler.

(TS//SI//REL) STUCCOMONTANA must support all modern versions of JUNOS, which is a version of FreeBSD customized by Juniper. Upon system boot, the JUNOS operating system is modified in memory to run the implant, and provide persistent kernel modifications to support implant execution.

(TS//SI//REL) STUCCOMONTANA is the cover term for the persistence technique to deploy a DNT implant to Juniper T-Series routers.

Unit Cost: $

Status: (U//FOUO) STUCCOMONTANA under development and is expected to be released by 30 November 2008.

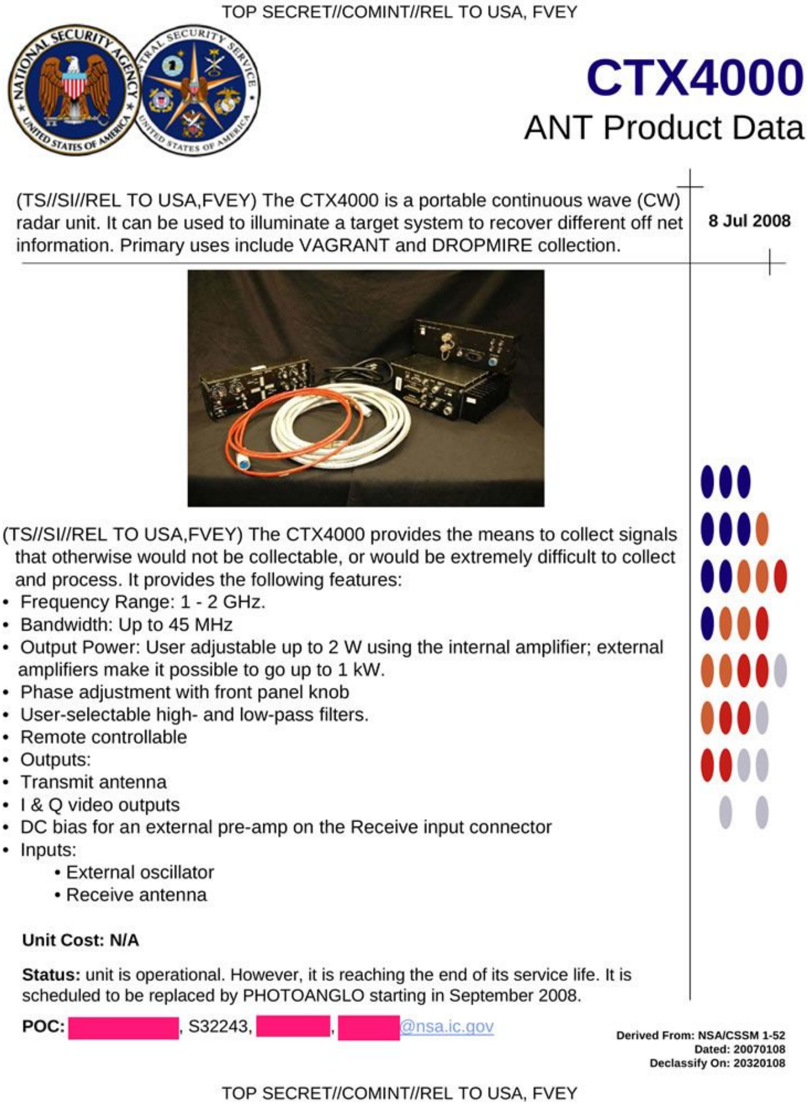
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.12 – STUCCOMONTANA. An Intel System Management Mode (SMM) BIOS rootkit used on system boot for persistence. Used to deploy other payloads from the BIOS of Juniper JUNOS J-Series routers. Configurable to carry a payload such as VALIDATOR as mentioned in Section 4.6.50. STUCCOMONTANA has a minimal rootkit capability when the targeted operating system is not supported by the payload. The rootkit is deployed through remote exploitation and/or interdiction and survives upgrades or replacements of the OS, including replacement of the compact flash card in the router.

Figure 4.39: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

CTX4000

ANT Product Data

(TS//SI//REL TO USA,FVEY) The CTX4000 is a portable continuous wave (CW) radar unit. It can be used to illuminate a target system to recover different off net information. Primary uses include VAGRANT and DROPMIRE collection.

8 Jul 2008

(TS//SI//REL TO USA,FVEY) The CTX4000 provides the means to collect signals that otherwise would not be collectable, or would be extremely difficult to collect and process. It provides the following features:

- Frequency Range: 1 - 2 GHz.
- Bandwidth: Up to 45 MHz
- Output Power: User adjustable up to 2 W using the internal amplifier; external amplifiers make it possible to go up to 1 kW.
- Phase adjustment with front panel knob
- User-selectable high- and low-pass filters.
- Remote controllable
- Outputs:
- Transmit antenna
- I & Q video outputs
- DC bias for an external pre-amp on the Receive input connector
- Inputs:
    - External oscillator
    - Receive antenna

Unit Cost: N/A

Status: unit is operational. However, it is reaching the end of its service life. It is scheduled to be replaced by PHOTOANGLO starting in September 2008.

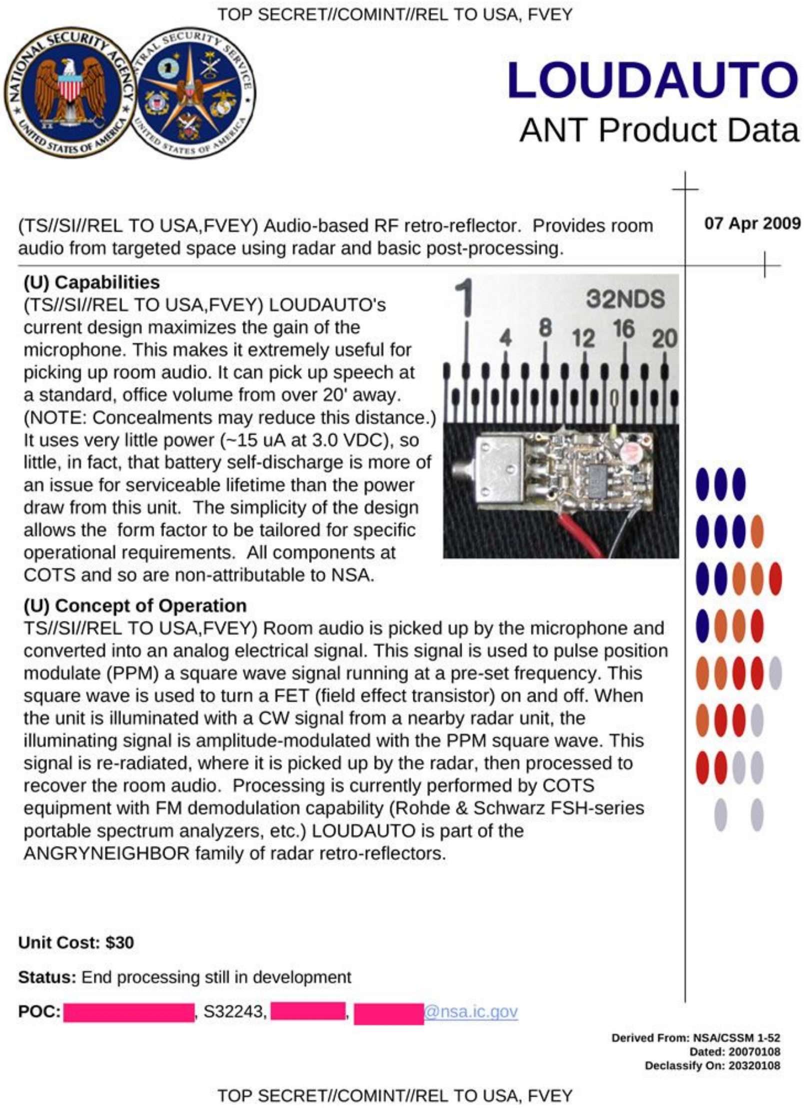
POC: [redacted], S32243, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.13 – CTX4000. An active radar device that generates an unmodulated, continuous wave (CW) signal that is between 1000Mhz and 2000Mhz with 45Mhz of bandwidth. Used for VAGRANT and DROPMIRE collection of signals; see also NIGHTWATCH as seen in Section 4.6.15, RAGEMASTER as seen in Section 4.6.35 implants. Scheduled for replacement by PHOTOANGLO as seen in Section 4.6.16.

Figure 4.40: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

LOUDAUTO

ANT Product Data

(TS//SI//REL TO USA,FVEY) Audio-based RF retro-reflector. Provides room audio from targeted space using radar and basic post-processing.

07 Apr 2009

(U) Capabilities

(TS//SI//REL TO USA,FVEY) LOUDAUTO's current design maximizes the gain of the microphone. This makes it extremely useful for picking up room audio. It can pick up speech at a standard, office volume from over 20' away. (NOTE: Concealments may reduce this distance.) It uses very little power (~15 uA at 3.0 VDC), so little, in fact, that battery self-discharge is more of an issue for serviceable lifetime than the power draw from this unit. The simplicity of the design allows the form factor to be tailored for specific operational requirements. All components at COTS and so are non-attributable to NSA.

(U) Concept of Operation

TS//SI//REL TO USA,FVEY) Room audio is picked up by the microphone and converted into an analog electrical signal. This signal is used to pulse position modulate (PPM) a square wave signal running at a pre-set frequency. This square wave is used to turn a FET (field effect transistor) on and off. When the unit is illuminated with a CW signal from a nearby radar unit, the illuminating signal is amplitude-modulated with the PPM square wave. This signal is re-radiated, where it is picked up by the radar, then processed to recover the room audio. Processing is currently performed by COTS equipment with FM demodulation capability (Rohde & Schwarz FSH-series portable spectrum analyzers, etc.) LOUDAUTO is part of the ANGRYNEIGHBOR family of radar retro-reflectors.

Unit Cost: $30

Status: End processing still in development

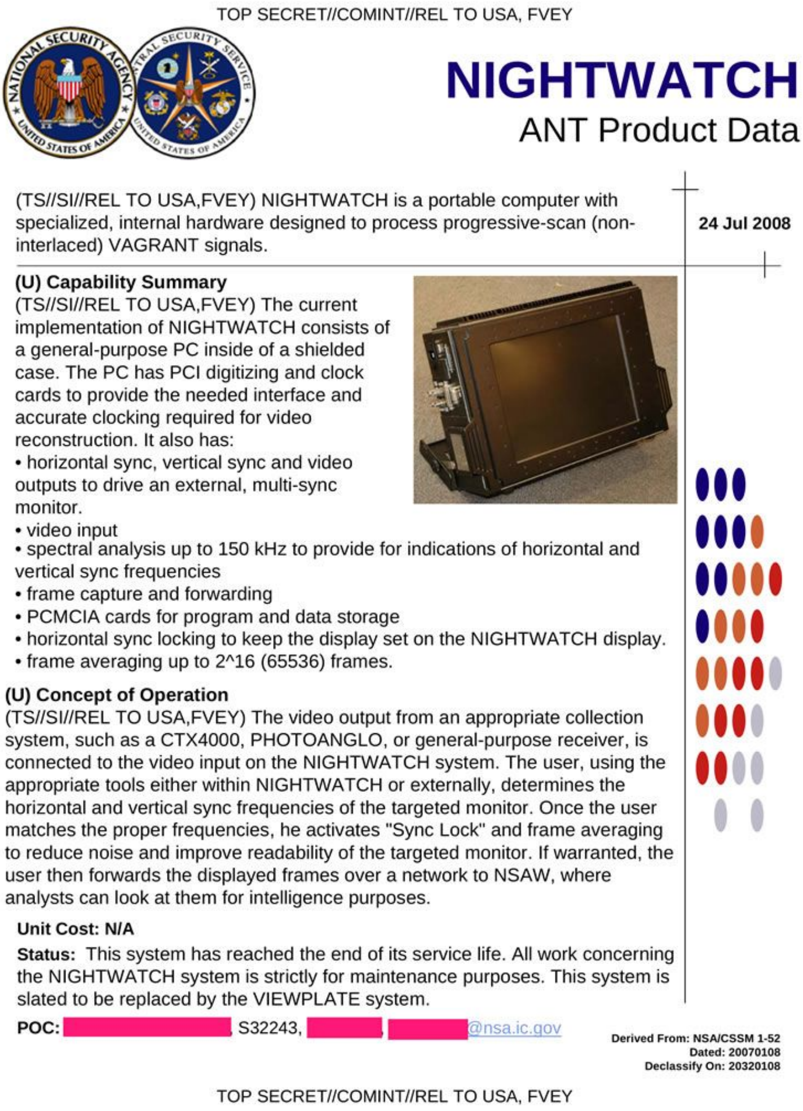
POC: [redacted], S32243, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.14 – LOUDAUTO. An FM audio radar retro-reflector for use with continuous-wave radar generators such as the CTX4000 as described in Section 4.6.13. LOUDAUTO is part of the ANGRYNEIGHBOR family of backdoors as seen in Sections 4.6.15, 4.6.35, 4.6.36, and 4.6.17. Remarkable is that this audio surveillance device is obviously vulnerable to fourth party collection with an FM radio.

Figure 4.41: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

NIGHTWATCH

ANT Product Data

(TS//SI//REL TO USA,FVEY) NIGHTWATCH is a portable computer with specialized, internal hardware designed to process progressive-scan (non-interlaced) VAGRANT signals.

24 Jul 2008

(U) Capability Summary

(TS//SI//REL TO USA,FVEY) The current implementation of NIGHTWATCH consists of a general-purpose PC inside of a shielded case. The PC has PCI digitizing and clock cards to provide the needed interface and accurate clocking required for video reconstruction. It also has:

- horizontal sync, vertical sync and video outputs to drive an external, multi-sync monitor.
- video input
- spectral analysis up to 150 kHz to provide for indications of horizontal and vertical sync frequencies
- frame capture and forwarding
- PCMCIA cards for program and data storage
- horizontal sync locking to keep the display set on the NIGHTWATCH display.
- frame averaging up to 2^16 (65536) frames.

(U) Concept of Operation

(TS//SI//REL TO USA,FVEY) The video output from an appropriate collection system, such as a CTX4000, PHOTOANGLO, or general-purpose receiver, is connected to the video input on the NIGHTWATCH system. The user, using the appropriate tools either within NIGHTWATCH or externally, determines the horizontal and vertical sync frequencies of the targeted monitor. Once the user matches the proper frequencies, he activates "Sync Lock" and frame averaging to reduce noise and improve readability of the targeted monitor. If warranted, the user then forwards the displayed frames over a network to NSAW, where analysts can look at them for intelligence purposes.

Unit Cost: N/A

Status: This system has reached the end of its service life. All work concerning the NIGHTWATCH system is strictly for maintenance purposes. This system is slated to be replaced by the VIEWPLATE system.

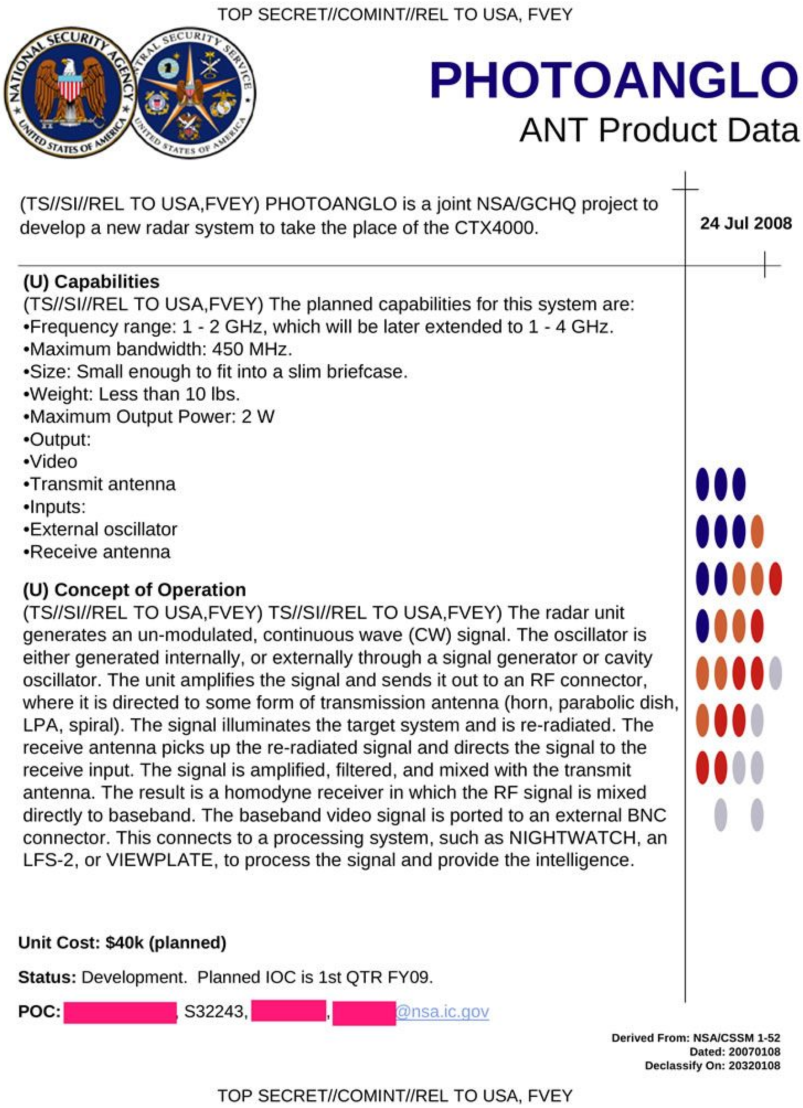
POC: [redacted] S32243, [redacted] [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.15 – NIGHTWATCH. A receiver for various radar retro-reflector systems such as CTX4000 as described in Section 4.6.13 or its replacement PHOTOANGLO as described in Section 4.6.16. Useful for watching video such as the kind exfiltrated by RAGEMASTER as described in Section 4.6.35. The data from the display can be forwarded for collection and analysis. The VIEWPLATE system is the planned successor to NIGHTWATCH. NIGHTWATCH is part of the ANGRYNEIGHBOR family of backdoors as seen in Sections 4.6.14, 4.6.35, 4.6.36, and 4.6.17.

Figure 4.42: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

PHOTOANGLO

ANT Product Data

(TS//SI//REL TO USA,FVEY) PHOTOANGLO is a joint NSA/GCHQ project to develop a new radar system to take the place of the CTX4000.

24 Jul 2008

(U) Capabilities

(TS//SI//REL TO USA,FVEY) The planned capabilities for this system are:

- Frequency range: 1 - 2 GHz, which will be later extended to 1 - 4 GHz.
- Maximum bandwidth: 450 MHz.
- Size: Small enough to fit into a slim briefcase.
- Weight: Less than 10 lbs.
- Maximum Output Power: 2 W
- Output:
- Video
- Transmit antenna
- Inputs:
- External oscillator
- Receive antenna

(U) Concept of Operation

(TS//SI//REL TO USA,FVEY) TS//SI//REL TO USA,FVEY) The radar unit generates an un-modulated, continuous wave (CW) signal. The oscillator is either generated internally, or externally through a signal generator or cavity oscillator. The unit amplifies the signal and sends it out to an RF connector, where it is directed to some form of transmission antenna (horn, parabolic dish, LPA, spiral). The signal illuminates the target system and is re-radiated. The receive antenna picks up the re-radiated signal and directs the signal to the receive input. The signal is amplified, filtered, and mixed with the transmit antenna. The result is a homodyne receiver in which the RF signal is mixed directly to baseband. The baseband video signal is ported to an external BNC connector. This connects to a processing system, such as NIGHTWATCH, an LFS-2, or VIEWPLATE, to process the signal and provide the intelligence.

Unit Cost: $40k (planned)

Status: Development. Planned IOC is 1st QTR FY09.

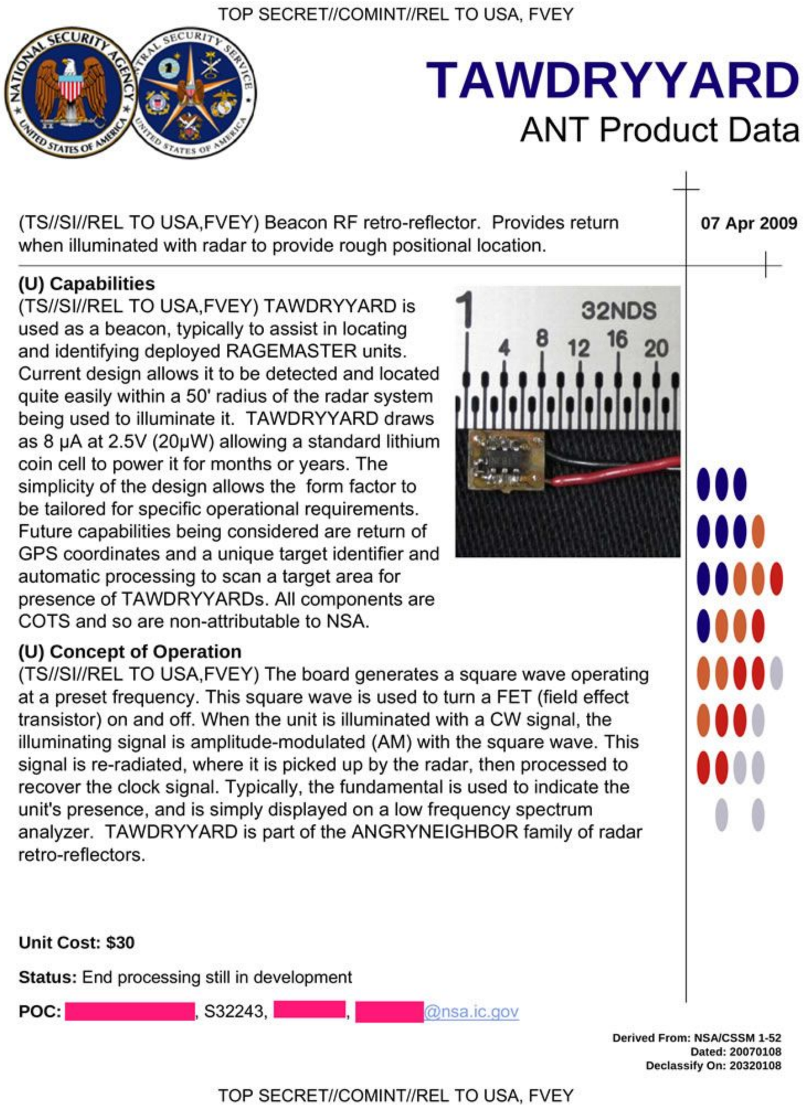
POC: [redacted], S32243, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.16 – PHOTOANGLO. An active radar device generating an unmodulated, continuous wave (CW) signal that is between 1000Mhz and 2000Mhz with 450Mhz of bandwidth. Used for VAGRANT and DROPMIRE collection of signals; see also NIGHTWATCH as seen in Section 4.6.15, RAGEMASTER as seen in Section 4.6.35. Replacement for CTX4000 as seen in Section 4.6.13 CW generator. PHOTOANGLO is part of the ANGRYNEIGHBOR family of backdoors as seen in Sections 4.6.14, 4.6.35, 4.6.36, and 4.6.17.

Figure 4.43: Courtesy of Der Spiegel [AHS13].


1
32NDS
4
8
12
16
20

TOP SECRET//COMINT//REL TO USA, FVEY

TAWDRYYARD

ANT Product Data

(TS//SI//REL TO USA,FVEY) Beacon RF retro-reflector. Provides return when illuminated with radar to provide rough positional location.

07 Apr 2009

(U) Capabilities

(TS//SI//REL TO USA,FVEY) TAWDRYYARD is used as a beacon, typically to assist in locating and identifying deployed RAGEMASTER units. Current design allows it to be detected and located quite easily within a 50' radius of the radar system being used to illuminate it. TAWDRYYARD draws as 8 μA at 2.5V (20μW) allowing a standard lithium coin cell to power it for months or years. The simplicity of the design allows the form factor to be tailored for specific operational requirements. Future capabilities being considered are return of GPS coordinates and a unique target identifier and automatic processing to scan a target area for presence of TAWDRYYARDs. All components are COTS and so are non-attributable to NSA.

(U) Concept of Operation

(TS//SI//REL TO USA,FVEY) The board generates a square wave operating at a preset frequency. This square wave is used to turn a FET (field effect transistor) on and off. When the unit is illuminated with a CW signal, the illuminating signal is amplitude-modulated (AM) with the square wave. This signal is re-radiated, where it is picked up by the radar, then processed to recover the clock signal. Typically, the fundamental is used to indicate the unit's presence, and is simply displayed on a low frequency spectrum analyzer. TAWDRYYARD is part of the ANGRYNEIGHBOR family of radar retro-reflectors.

Unit Cost: $30

Status: End processing still in development

POC: [redacted], S32243, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.17 – TAWDRYYARD. A continuous-wave (CW) radar backdoor for geolocation of devices. TAWDRYYARD is part of the ANGRYNEIGHBOR family of backdoors as seen in Sections 4.6.14, 4.6.16, 4.6.15, 4.6.35, and 4.6.36.

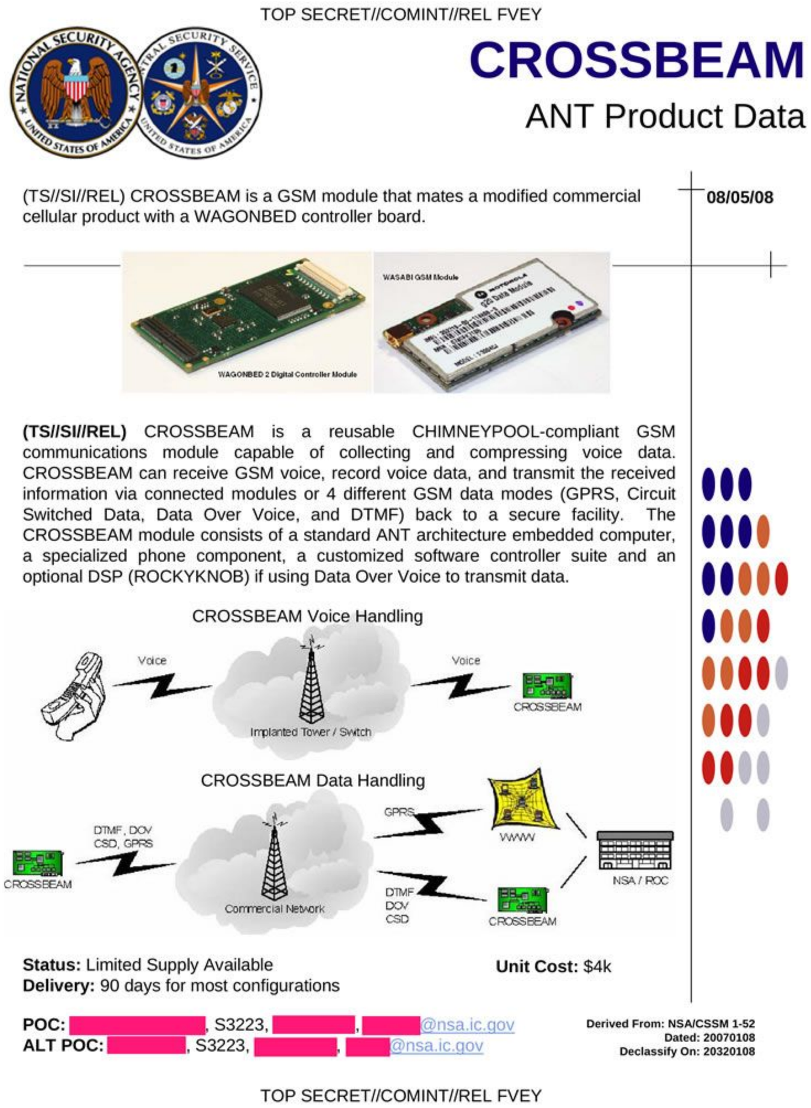


Figure 4.44: Courtesy of Der Spiegel [AHS13].

TOP SECRET//COMINT//REL FVEY

CROSSBEAM

ANT Product Data

(TS//SI//REL) CROSSBEAM is a GSM module that mates a modified commercial cellular product with a WAGONBED controller board.

08/05/08

WAGONBED 2 Digital Controller Module
WASABI GSM Module

(TS//SI//REL) CROSSBEAM is a reusable CHIMNEYPOOL-compliant GSM communications module capable of collecting and compressing voice data. CROSSBEAM can receive GSM voice, record voice data, and transmit the received information via connected modules or 4 different GSM data modes (GPRS, Circuit Switched Data, Data Over Voice, and DTMF) back to a secure facility. The CROSSBEAM module consists of a standard ANT architecture embedded computer, a specialized phone component, a customized software controller suite and an optional DSP (ROCKYKNOB) if using Data Over Voice to transmit data.

CROSSBEAM Voice Handling

Voice
Implanted Tower / Switch
Voice
CROSSBEAM

CROSSBEAM Data Handling

DTMF, DOV
CSD, GPRS
CROSSBEAM
Commercial Network
GPRS
WWW
DTMF
DOV
CSD
CROSSBEAM
NSA / ROC

Status: Limited Supply Available
Delivery: 90 days for most configurations

Unit Cost: $4k

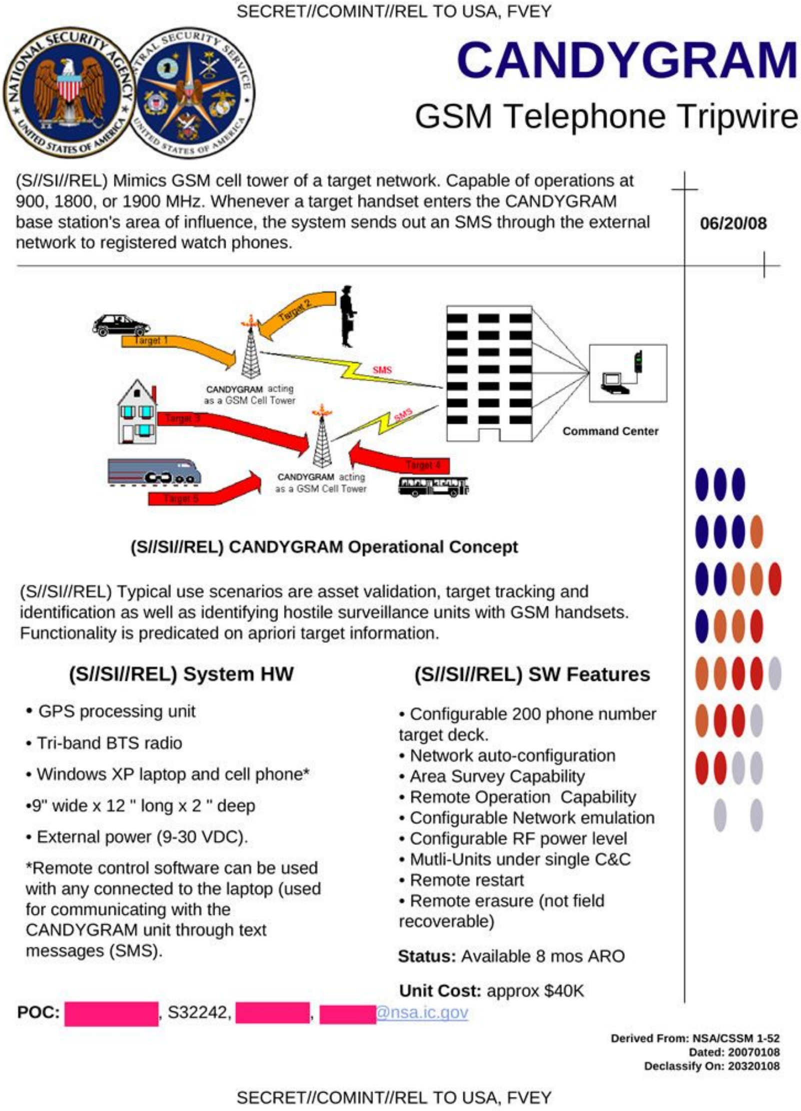
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL FVEY

4.6.18 – CROSSBEAM. A communication module implant that is paired with a commercial GSM device such as the pictured Motorola modem and a WAGONBED controller board. Exfiltration is possible over non-Internet communications channels, and even non-digital data-based channels such as voice channels of a commercial cellular network.

Figure 4.45: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

CANDYGRAM

GSM Telephone Tripwire

(S//SI//REL) Mimics GSM cell tower of a target network. Capable of operations at 900, 1800, or 1900 MHz. Whenever a target handset enters the CANDYGRAM base station's area of influence, the system sends out an SMS through the external network to registered watch phones.

06/20/08

Target 1
Target 2
SMS
CANDYGRAM acting as a GSM Cell Tower
Target 3
SMS
Command Center
Target 4
Target 5
CANDYGRAM acting as a GSM Cell Tower

(S//SI//REL) CANDYGRAM Operational Concept

(S//SI//REL) Typical use scenarios are asset validation, target tracking and identification as well as identifying hostile surveillance units with GSM handsets. Functionality is predicated on apriori target information.

(S//SI//REL) System HW

- GPS processing unit
- Tri-band BTS radio
- Windows XP laptop and cell phone*
- 9" wide x 12 " long x 2 " deep
- External power (9-30 VDC).

*Remote control software can be used with any connected to the laptop (used for communicating with the CANDYGRAM unit through text messages (SMS).

(S//SI//REL) SW Features

- Configurable 200 phone number target deck.
- Network auto-configuration
- Area Survey Capability
- Remote Operation Capability
- Configurable Network emulation
- Configurable RF power level
- Mutli-Units under single C&C
- Remote restart
- Remote erasure (not field recoverable)

Status: Available 8 mos ARO

Unit Cost: approx $40K

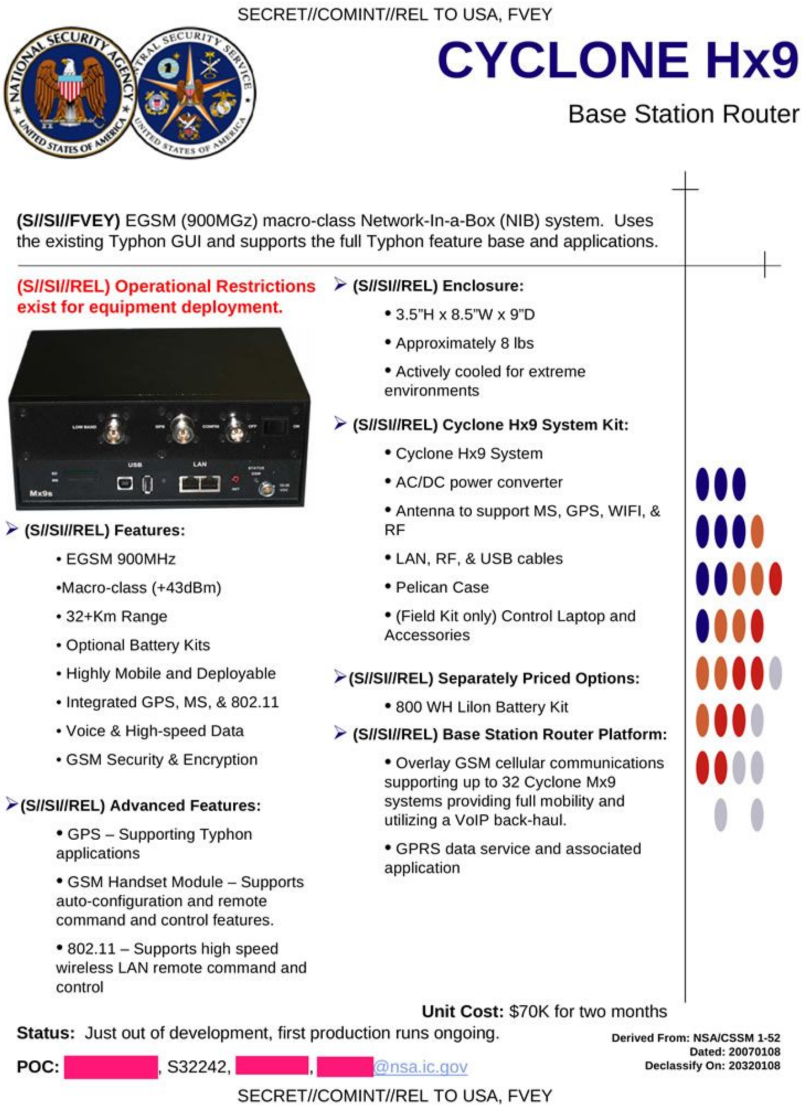
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.19 – CANDYGRAM. A cell phone base station used for surveillance, also known as an IMSI-catcher [PS14], that is used for targeting people who have GSM telephones. Used for targeting human beings for geolocation based on their cellular telephone. The CANDYGRAM backdoor has an unspecified text message remote control interface. References to a program known as LANDSHARK show up in the EBSR as mentioned in Section 4.6.21 and it is presented as similar to CANDYGRAM.

Figure 4.46: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

CYCLONE Hx9

Base Station Router

(S//SI//FVEY) EGSM (900MGz) macro-class Network-In-a-Box (NIB) system. Uses the existing Typhon GUI and supports the full Typhon feature base and applications.

(S//SI//REL) Operational Restrictions exist for equipment deployment.

USB
LAN
Mx9s

- **(S//SI//REL) Features:**
  - EGSM 900MHz
  - Macro-class (+43dBm)
  - 32+Km Range
  - Optional Battery Kits
  - Highly Mobile and Deployable
  - Integrated GPS, MS, & 802.11
  - Voice & High-speed Data
  - GSM Security & Encryption
- **(S//SI//REL) Advanced Features:**
  - GPS – Supporting Typhon applications
  - GSM Handset Module – Supports auto-configuration and remote command and control features.
  - 802.11 – Supports high speed wireless LAN remote command and control
- **(S//SI//REL) Enclosure:**
  - 3.5"H x 8.5"W x 9"D
  - Approximately 8 lbs
  - Actively cooled for extreme environments
- **(S//SI//REL) Cyclone Hx9 System Kit:**
  - Cyclone Hx9 System
  - AC/DC power converter
  - Antenna to support MS, GPS, WIFI, & RF
  - LAN, RF, & USB cables
  - Pelican Case
  - (Field Kit only) Control Laptop and Accessories
- **(S//SI//REL) Separately Priced Options:**
  - 800 WH LiIon Battery Kit
- **(S//SI//REL) Base Station Router Platform:**
  - Overlay GSM cellular communications supporting up to 32 Cyclone Mx9 systems providing full mobility and utilizing a VoIP back-haul.
  - GPRS data service and associated application

Unit Cost: $70K for two months

Status: Just out of development, first production runs ongoing.

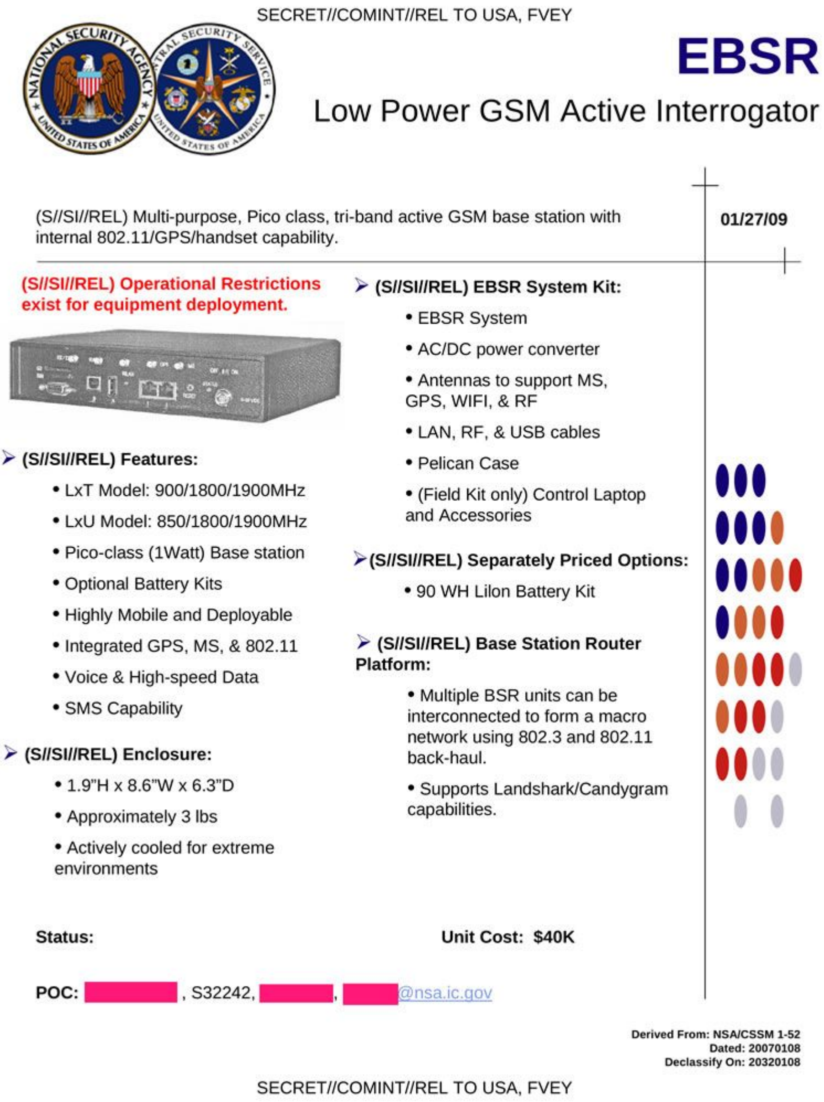
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.20 – CYCLONE Hx9. An IMSI-catcher for 900Mhz EGSM for use with TYPHON. Range of 32 kilometers. Optional battery-powered operation. See also TYPHON-HX in Section 4.6.25.

Figure 4.47: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

EBSR

Low Power GSM Active Interrogator

(S//SI//REL) Multi-purpose, Pico class, tri-band active GSM base station with internal 802.11/GPS/handset capability.

01/27/09

(S//SI//REL) Operational Restrictions exist for equipment deployment.

➢ **(S//SI//REL) Features:**

- LxT Model: 900/1800/1900MHz
- LxU Model: 850/1800/1900MHz
- Pico-class (1Watt) Base station
- Optional Battery Kits
- Highly Mobile and Deployable
- Integrated GPS, MS, & 802.11
- Voice & High-speed Data
- SMS Capability

➢ **(S//SI//REL) Enclosure:**

- 1.9"H x 8.6"W x 6.3"D
- Approximately 3 lbs
- Actively cooled for extreme environments

➢ **(S//SI//REL) EBSR System Kit:**

- EBSR System
- AC/DC power converter
- Antennas to support MS, GPS, WIFI, & RF
- LAN, RF, & USB cables
- Pelican Case
- (Field Kit only) Control Laptop and Accessories

➢ **(S//SI//REL) Separately Priced Options:**

- 90 WH LiIon Battery Kit

➢ **(S//SI//REL) Base Station Router Platform:**

- Multiple BSR units can be interconnected to form a macro network using 802.3 and 802.11 back-haul.
- Supports Landshark/Candygram capabilities.

Status:

Unit Cost: $40K

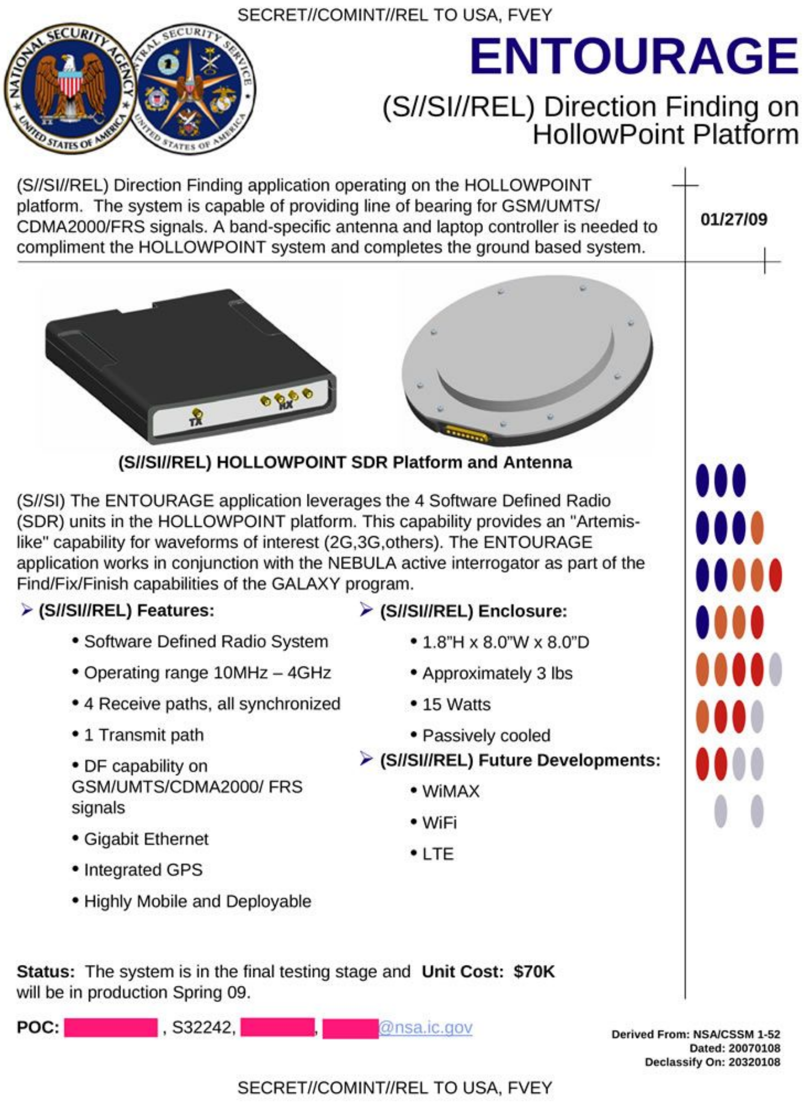
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.21 – EBSR. A low-power IMSI-catcher with support for tri-band GSM frequencies. See CANDYGRAM as shown in Section 4.6.19 for related capabilities. References to a program known as LANDSHARK.

Figure 4.48: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

ENTOURAGE

(S//SI//REL) Direction Finding on HollowPoint Platform

(S//SI//REL) Direction Finding application operating on the HOLLOWPOINT platform. The system is capable of providing line of bearing for GSM/UMTS/CDMA2000/FRS signals. A band-specific antenna and laptop controller is needed to compliment the HOLLOWPOINT system and completes the ground based system.

01/27/09

(S//SI//REL) HOLLOWPOINT SDR Platform and Antenna

(S//SI) The ENTOURAGE application leverages the 4 Software Defined Radio (SDR) units in the HOLLOWPOINT platform. This capability provides an "Artemis-like" capability for waveforms of interest (2G,3G,others). The ENTOURAGE application works in conjunction with the NEBULA active interrogator as part of the Find/Fix/Finish capabilities of the GALAXY program.

- **(S//SI//REL) Features:**
  - Software Defined Radio System
  - Operating range 10MHz – 4GHz
  - 4 Receive paths, all synchronized
  - 1 Transmit path
  - DF capability on GSM/UMTS/CDMA2000/ FRS signals
  - Gigabit Ethernet
  - Integrated GPS
  - Highly Mobile and Deployable
- **(S//SI//REL) Enclosure:**
  - 1.8"H x 8.0"W x 8.0"D
  - Approximately 3 lbs
  - 15 Watts
  - Passively cooled
- **(S//SI//REL) Future Developments:**
  - WiMAX
  - WiFi
  - LTE

Status: The system is in the final testing stage and will be in production Spring 09.

Unit Cost: $70K

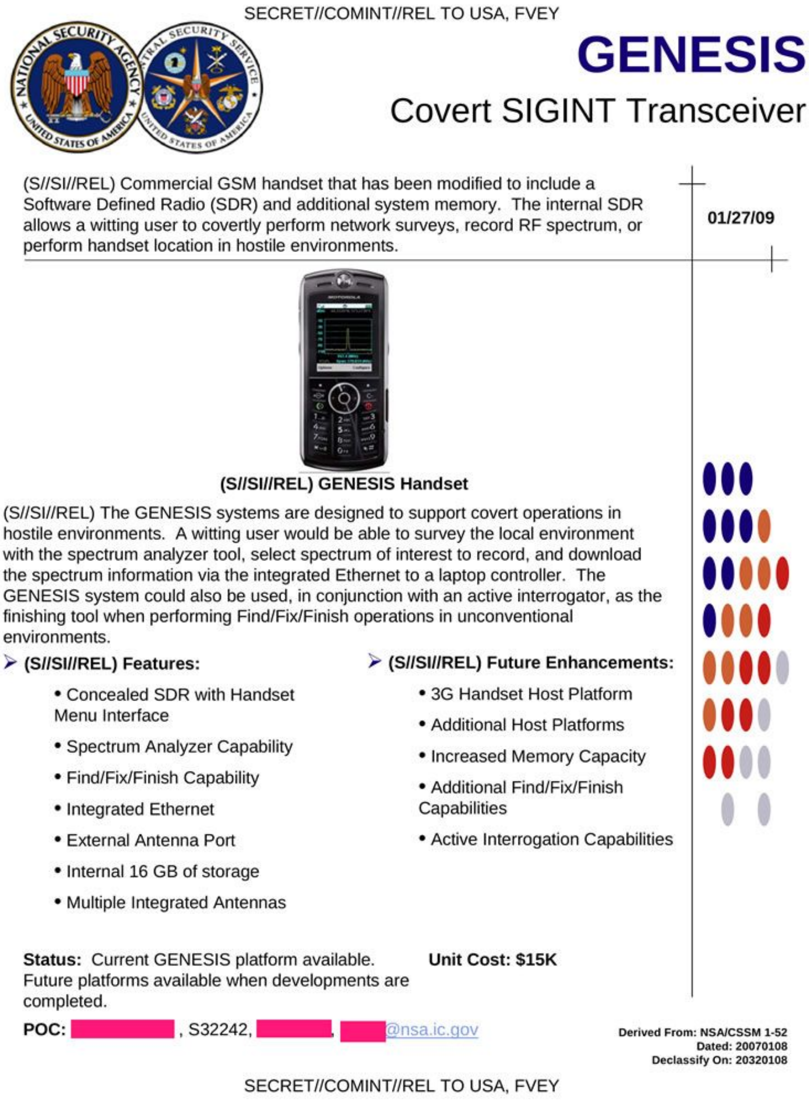
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.22 – ENTOURAGE. An application for driving the HOLLOWPOINT hardware that consists of four Software Defined Radio (SDR) units. ENTOURAGE is used for direction finding (DF) of signals of interest such as GSM, UTMS, CDMA2000, FRS. Planned future support for WiMAX, WiFi, and LTE is more than likely now implemented in this device or in a successor device. If the target is in range of a NEBULA base station as seen in Section 4.6.24, ENTOURAGE is able to track the target thanks to active probing by NEBULA as part of a so-called *Find/Fix/Finish* process.

Figure 4.49: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

GENESIS

Covert SIGINT Transceiver

(S//SI//REL) Commercial GSM handset that has been modified to include a Software Defined Radio (SDR) and additional system memory. The internal SDR allows a witting user to covertly perform network surveys, record RF spectrum, or perform handset location in hostile environments.

01/27/09

(S//SI//REL) GENESIS Handset

(S//SI//REL) The GENESIS systems are designed to support covert operations in hostile environments. A witting user would be able to survey the local environment with the spectrum analyzer tool, select spectrum of interest to record, and download the spectrum information via the integrated Ethernet to a laptop controller. The GENESIS system could also be used, in conjunction with an active interrogator, as the finishing tool when performing Find/Fix/Finish operations in unconventional environments.

(S//SI//REL) Features:

- Concealed SDR with Handset Menu Interface
- Spectrum Analyzer Capability
- Find/Fix/Finish Capability
- Integrated Ethernet
- External Antenna Port
- Internal 16 GB of storage
- Multiple Integrated Antennas

(S//SI//REL) Future Enhancements:

- 3G Handset Host Platform
- Additional Host Platforms
- Increased Memory Capacity
- Additional Find/Fix/Finish Capabilities
- Active Interrogation Capabilities

Status: Current GENESIS platform available. Future platforms available when developments are completed.

Unit Cost: $15K

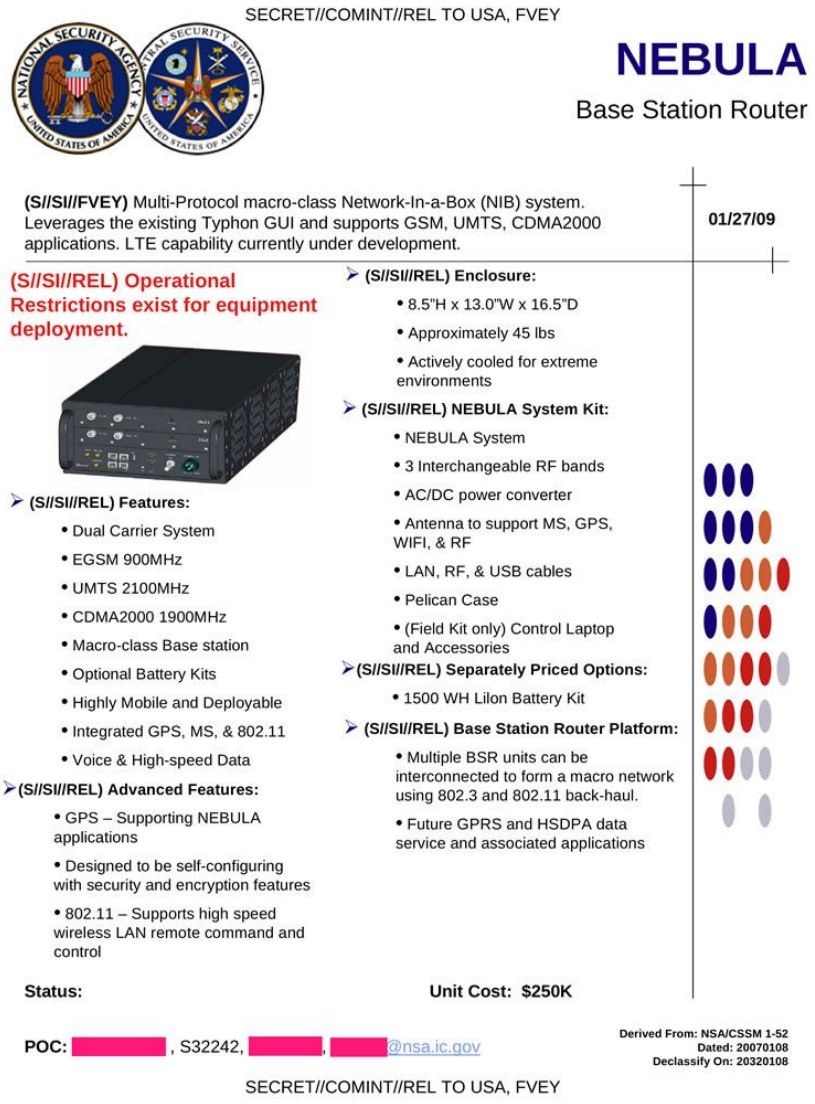
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.23 – GENESIS. A small handheld Software Defined Radio (SDR) for covert radio and network operation surveillance. It is in the form factor of a contemporary (2008) GSM handset and is intended to be used in covert operations such as *Find/Fix/Finish*. It includes features not normally found in GSM handsets such as a spectrum analysis tool and Ethernet.

Figure 4.50: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

NEBULA

Base Station Router

01/27/09

(S//SI//FVEY) Multi-Protocol macro-class Network-In-a-Box (NIB) system. Leverages the existing Typhon GUI and supports GSM, UMTS, CDMA2000 applications. LTE capability currently under development.

(S//SI//REL) Operational Restrictions exist for equipment deployment.

➢ **(S//SI//REL) Features:**

- Dual Carrier System
- EGSM 900MHz
- UMTS 2100MHz
- CDMA2000 1900MHz
- Macro-class Base station
- Optional Battery Kits
- Highly Mobile and Deployable
- Integrated GPS, MS, & 802.11
- Voice & High-speed Data

➢ **(S//SI//REL) Advanced Features:**

- GPS – Supporting NEBULA applications
- Designed to be self-configuring with security and encryption features
- 802.11 – Supports high speed wireless LAN remote command and control

➢ **(S//SI//REL) Enclosure:**

- 8.5"H x 13.0"W x 16.5"D
- Approximately 45 lbs
- Actively cooled for extreme environments

➢ **(S//SI//REL) NEBULA System Kit:**

- NEBULA System
- 3 Interchangeable RF bands
- AC/DC power converter
- Antenna to support MS, GPS, WIFI, & RF
- LAN, RF, & USB cables
- Pelican Case
- (Field Kit only) Control Laptop and Accessories

➢ **(S//SI//REL) Separately Priced Options:**

- 1500 WH LiIon Battery Kit

➢ **(S//SI//REL) Base Station Router Platform:**

- Multiple BSR units can be interconnected to form a macro network using 802.3 and 802.11 back-haul.
- Future GPRS and HSDPA data service and associated applications

Status:

Unit Cost: $250K

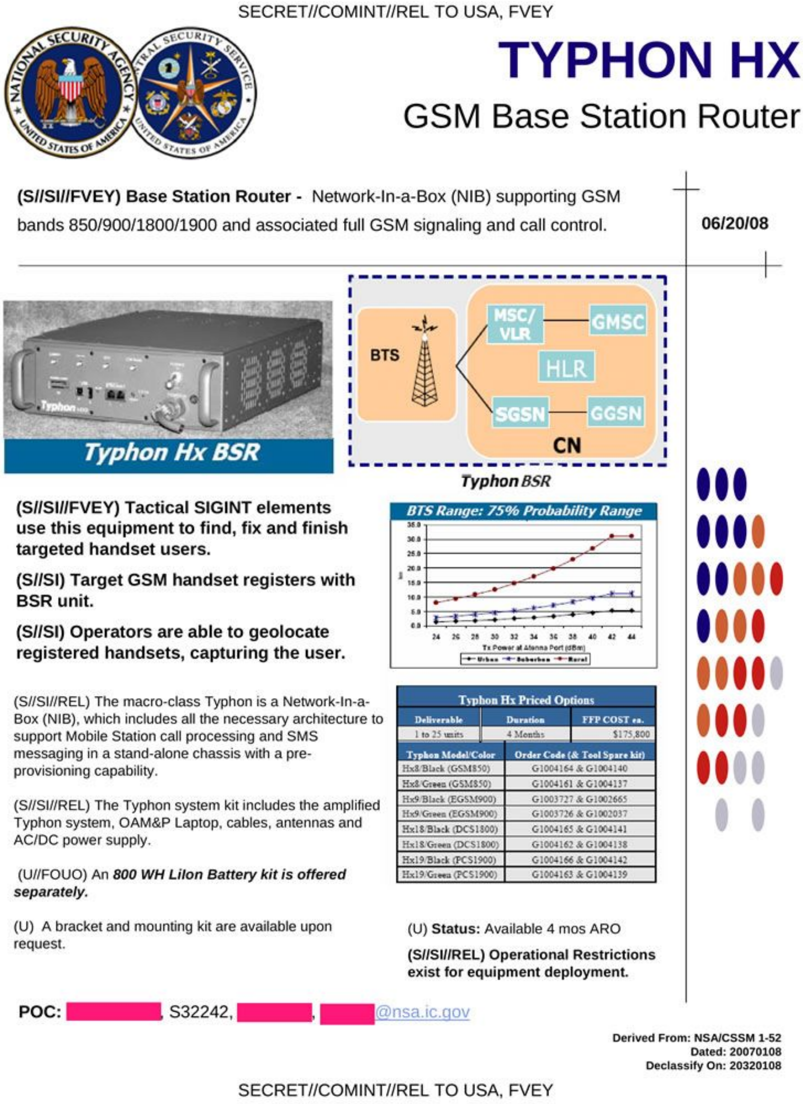
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.24 – NEBULA. An IMSI-catcher used to geolocate mobile phone users using the NEBULA device as their base station. Support for GSM, UMTS, and CDMA2000 radio protocols. See also TYPHON-HX as shown in Section 4.6.25, as well as CYCLONE-HX9 as shown in Section 4.6.20 and WATERWITCH as shown in Section 4.6.26. Future plans to expand to LTE.

Figure 4.51: Courtesy of Der Spiegel [AHS13].


Typhon Hx BSR
BTS
MSC/ VLR
GMSC
HLR
SGSN
GGSN
CN
Typhon BSR
BTS Range: 75% Probability Range
35.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0
24
26
28
30
32
34
36
38
40
42
44
Tx Power at Atenna Port (dBm)
Urban
Suburban
Rural

SECRET//COMINT//REL TO USA, FVEY

TYPHON HX

GSM Base Station Router

(S//SI//FVEY) Base Station Router - Network-In-a-Box (NIB) supporting GSM bands 850/900/1800/1900 and associated full GSM signaling and call control.

06/20/08

(S//SI//FVEY) Tactical SIGINT elements use this equipment to find, fix and finish targeted handset users.

(S//SI) Target GSM handset registers with BSR unit.

(S//SI) Operators are able to geolocate registered handsets, capturing the user.

(S//SI//REL) The macro-class Typhon is a Network-In-a-Box (NIB), which includes all the necessary architecture to support Mobile Station call processing and SMS messaging in a stand-alone chassis with a pre-provisioning capability.

(S//SI//REL) The Typhon system kit includes the amplified Typhon system, OAM&P Laptop, cables, antennas and AC/DC power supply.

(U//FOUO) An ***800 WH LiIon Battery kit is offered separately.***

(U) A bracket and mounting kit are available upon request.

Typhon Hx Priced Options

Deliverable	Duration	FFP COST ea.
1 to 25 units	4 Months	$175,800

Typhon Model/Color	Order Code (& Tool Spare kit)
Hx8/Black (GSM850)	G1004164 & G1004140
Hx8/Green (GSM850)	G1004161 & G1004137
Hx9/Black (EGSM900)	G1003727 & G1002665
Hx9/Green (EGSM900)	G1003726 & G1002037
Hx18/Black (DCS1800)	G1004165 & G1004141
Hx18/Green (DCS1800)	G1004162 & G1004138
Hx19/Black (PCS1900)	G1004166 & G1004142
Hx19/Green (PCS1900)	G1004163 & G1004139

(U) **Status:** Available 4 mos ARO

(S//SI//REL) Operational Restrictions exist for equipment deployment.

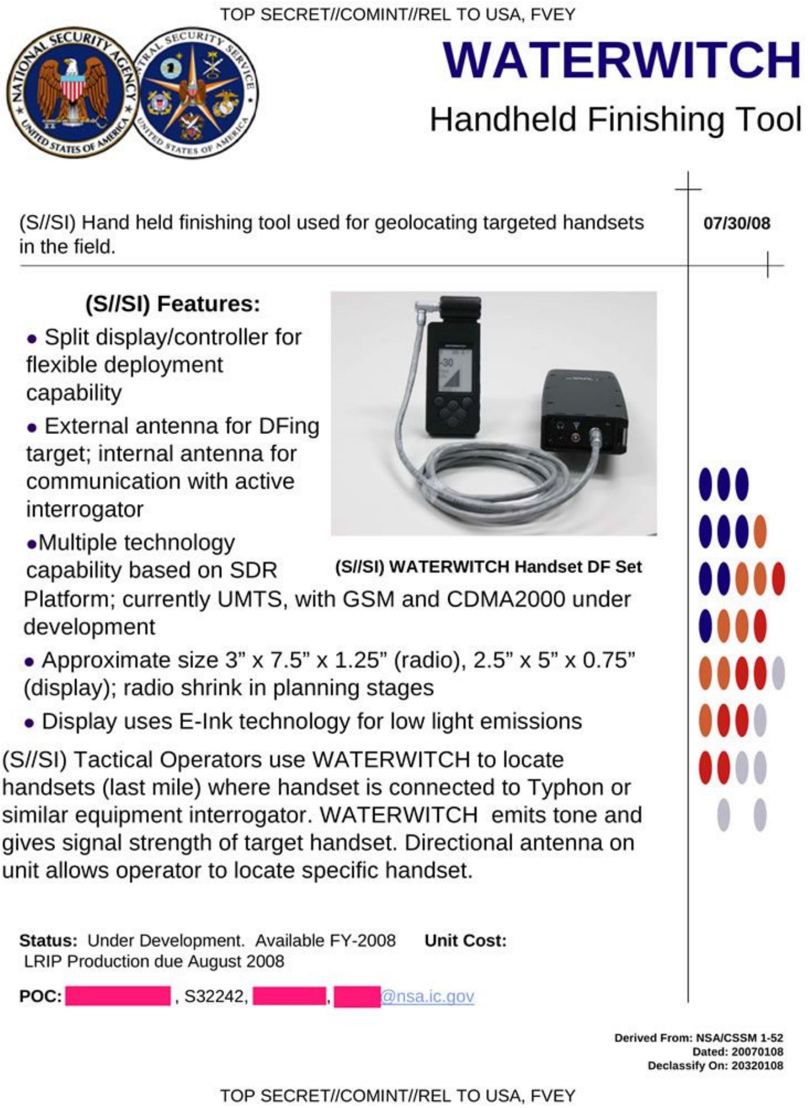
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.25 – TYPHON HX. An IMSI-catcher used to geolocate mobile phone users using the TYPHON-HX device as their base station. See also CYCLONE-HX9 as shown in Section 4.6.20 and WATERWITCH in as shown in Section 4.6.26.

Figure 4.52: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

WATERWITCH

Handheld Finishing Tool

(S//SI) Hand held finishing tool used for geolocating targeted handsets in the field.

07/30/08

(S//SI) Features:

- Split display/controller for flexible deployment capability
- External antenna for DFing target; internal antenna for communication with active interrogator
- Multiple technology capability based on SDR Platform; currently UMTS, with GSM and CDMA2000 under development
- Approximate size 3” x 7.5” x 1.25” (radio), 2.5” x 5” x 0.75” (display); radio shrink in planning stages
- Display uses E-Ink technology for low light emissions

(S//SI) WATERWITCH Handset DF Set

(S//SI) Tactical Operators use WATERWITCH to locate handsets (last mile) where handset is connected to Typhon or similar equipment interrogator. WATERWITCH emits tone and gives signal strength of target handset. Directional antenna on unit allows operator to locate specific handset.

Status: Under Development. Available FY-2008 **Unit Cost:**
LRIP Production due August 2008

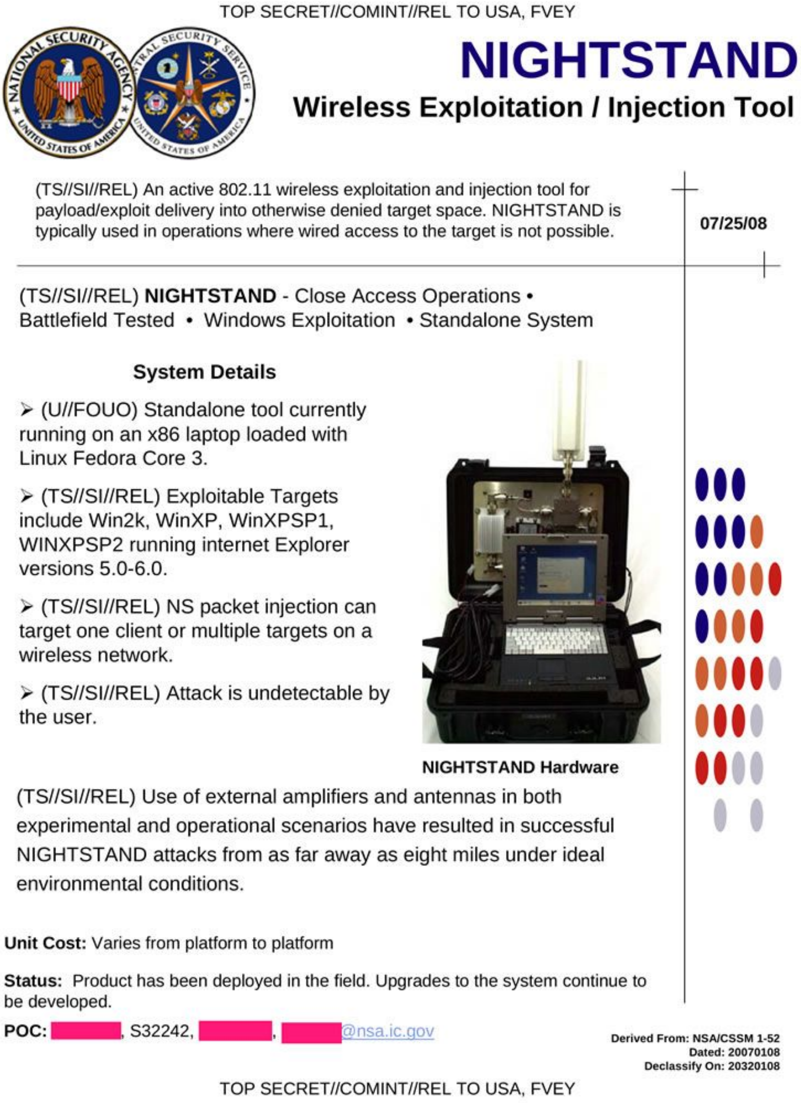
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.26 – WATERWITCH. A hand-held tool to geolocate cellular telephones. Usable with other TYPHON systems such as TYPHON-HX in as shown in Section 4.6.25 and CYCLONE-HX9 as shown in Section 4.6.20.

Figure 4.53: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

NIGHTSTAND

Wireless Exploitation / Injection Tool

(TS//SI//REL) An active 802.11 wireless exploitation and injection tool for payload/exploit delivery into otherwise denied target space. NIGHTSTAND is typically used in operations where wired access to the target is not possible.

07/25/08

(TS//SI//REL) **NIGHTSTAND** - Close Access Operations • Battlefield Tested • Windows Exploitation • Standalone System

System Details

- (U//FOUO) Standalone tool currently running on an x86 laptop loaded with Linux Fedora Core 3.
- (TS//SI//REL) Exploitable Targets include Win2k, WinXP, WinXPSP1, WINXPSP2 running internet Explorer versions 5.0-6.0.
- (TS//SI//REL) NS packet injection can target one client or multiple targets on a wireless network.
- (TS//SI//REL) Attack is undetectable by the user.

NIGHTSTAND Hardware

(TS//SI//REL) Use of external amplifiers and antennas in both experimental and operational scenarios have resulted in successful NIGHTSTAND attacks from as far away as eight miles under ideal environmental conditions.

Unit Cost: Varies from platform to platform

Status: Product has been deployed in the field. Upgrades to the system continue to be developed.

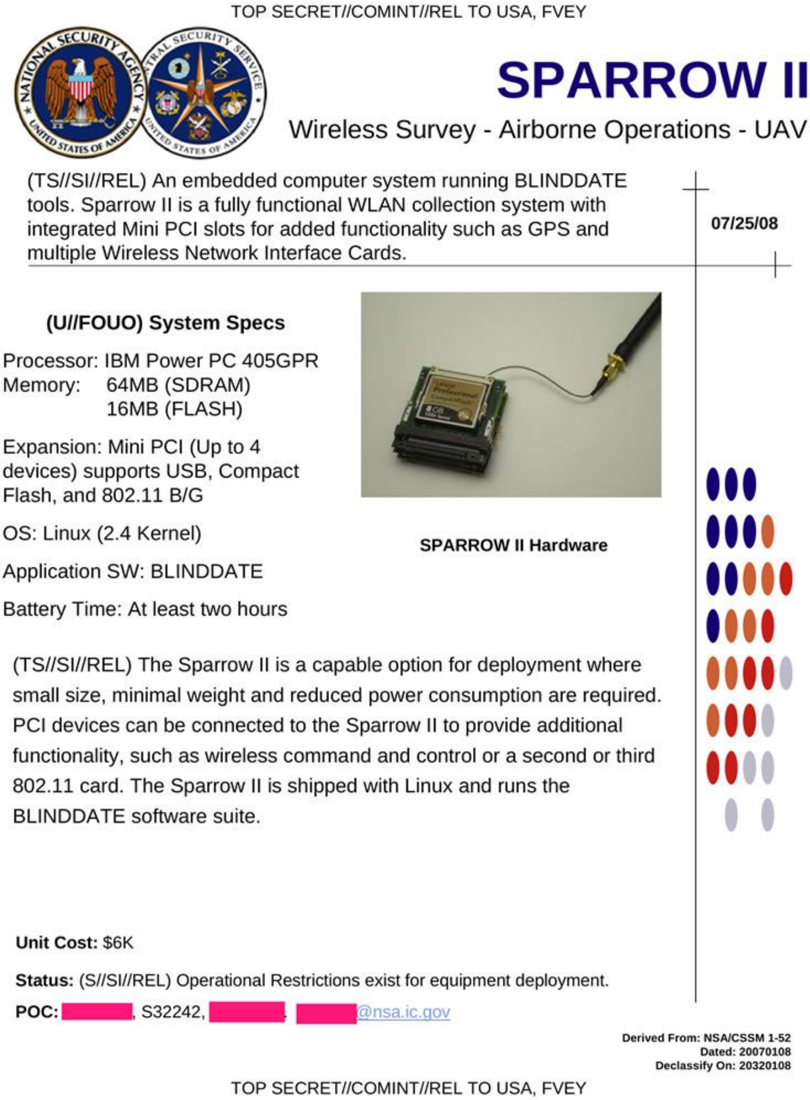
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.27 – NIGHTSTAND. A rich man's KARMA [DZM05] or KARMETASPLOIT [Oez09] running GNU/Linux (Fedora). Used for attacking WiFi-enabled systems. Vectors include exploitation of wireless (802.11) network kernel drivers. Targets various Microsoft Windows versions.

Figure 4.54: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

SPARROW II

Wireless Survey - Airborne Operations - UAV

(TS//SI//REL) An embedded computer system running BLINDDATE tools. Sparrow II is a fully functional WLAN collection system with integrated Mini PCI slots for added functionality such as GPS and multiple Wireless Network Interface Cards.

07/25/08

(U//FOUO) System Specs

Processor: IBM Power PC 405GPR
Memory: 64MB (SDRAM)
16MB (FLASH)

Expansion: Mini PCI (Up to 4 devices) supports USB, Compact Flash, and 802.11 B/G

OS: Linux (2.4 Kernel)

Application SW: BLINDDATE

Battery Time: At least two hours

SPARROW II Hardware

(TS//SI//REL) The Sparrow II is a capable option for deployment where small size, minimal weight and reduced power consumption are required. PCI devices can be connected to the Sparrow II to provide additional functionality, such as wireless command and control or a second or third 802.11 card. The Sparrow II is shipped with Linux and runs the BLINDDATE software suite.

Unit Cost: $6K

Status: (S//SI//REL) Operational Restrictions exist for equipment deployment.

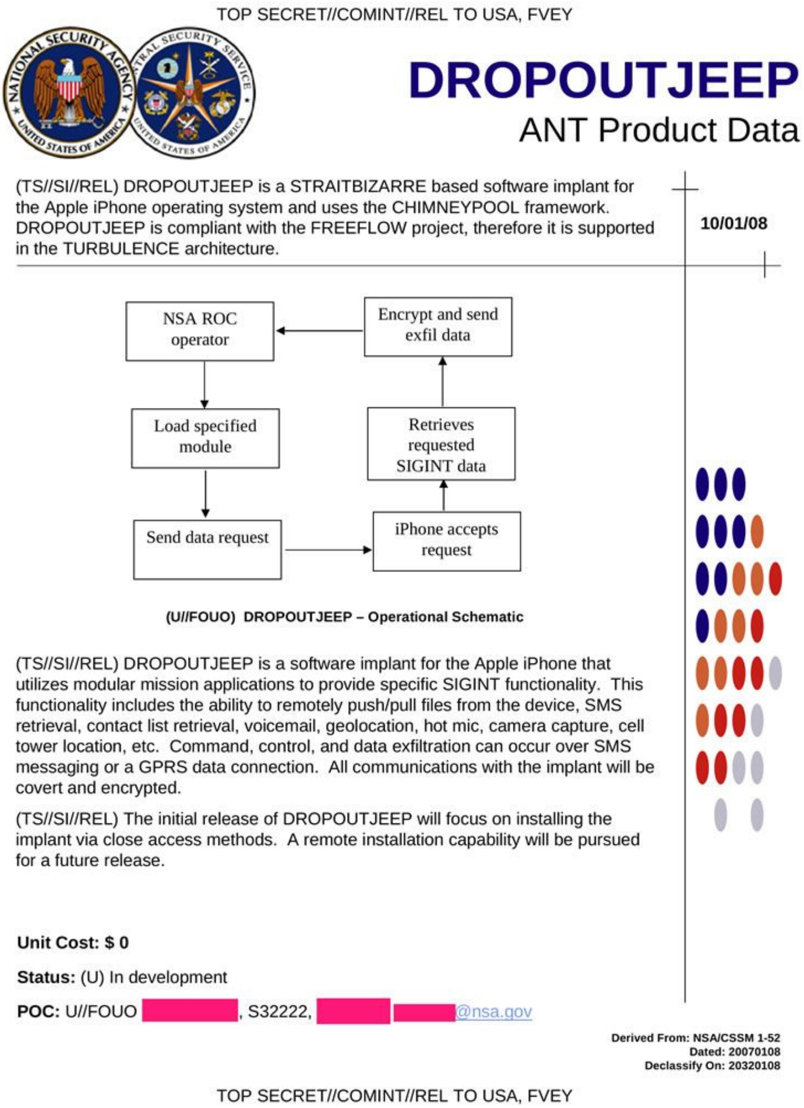
POC: [redacted], S32242, [redacted]. [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.28 – SPARROW-II. A battery-powered embedded PowerPC (405GPR) computer with 64 MB of SDRAM and 16 MB of FLASH storage running GNU/Linux. Used for wireless LAN data collection. Expandable for additional capabilities through the mini-PCI bus. Uses the BLINDDATE software for attacking wireless networks. See also NIGHTSTAND in Section 4.6.27.

Figure 4.55: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

DROPOUTJEEP

ANT Product Data

(TS//SI//REL) DROPOUTJEEP is a STRAITBIZARRE based software implant for the Apple iPhone operating system and uses the CHIMNEYPOOL framework. DROPOUTJEEP is compliant with the FREEFLOW project, therefore it is supported in the TURBULENCE architecture.

10/01/08

NSA ROC operator
Encrypt and send exfil data
Load specified module
Retrieves requested SIGINT data
Send data request
iPhone accepts request

(U//FOUO) DROPOUTJEEP – Operational Schematic

(TS//SI//REL) DROPOUTJEEP is a software implant for the Apple iPhone that utilizes modular mission applications to provide specific SIGINT functionality. This functionality includes the ability to remotely push/pull files from the device, SMS retrieval, contact list retrieval, voicemail, geolocation, hot mic, camera capture, cell tower location, etc. Command, control, and data exfiltration can occur over SMS messaging or a GPRS data connection. All communications with the implant will be covert and encrypted.

(TS//SI//REL) The initial release of DROPOUTJEEP will focus on installing the implant via close access methods. A remote installation capability will be pursued for a future release.

Unit Cost: **$ 0**

Status: (U) In development

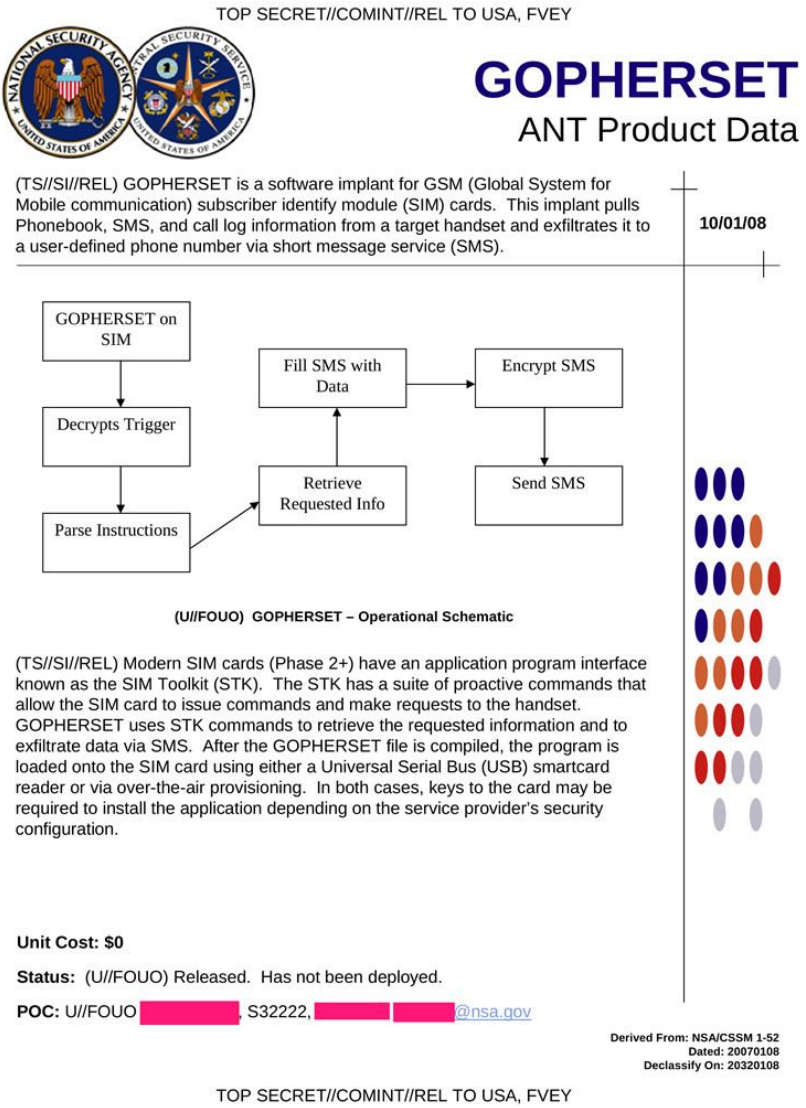
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.29 – DROPOUTJEEP. An iOS software implant that allows a remote operator to exfiltrate data from an Apple iPhone. Among other capabilities, can activate the microphone for real time audio monitoring of the surrounding area even when the phone otherwise appears inactive. Modern variations of this kind of iOS cellphone implant are extremely powerful and these implants are produced by a variety of commercial and government adversaries.

Figure 4.56: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

GOPHERSET

ANT Product Data

(TS//SI//REL) GOPHERSET is a software implant for GSM (Global System for Mobile communication) subscriber identify module (SIM) cards. This implant pulls Phonebook, SMS, and call log information from a target handset and exfiltrates it to a user-defined phone number via short message service (SMS).

10/01/08

GOPHERSET on SIM
Decrypts Trigger
Parse Instructions
Retrieve Requested Info
Fill SMS with Data
Encrypt SMS
Send SMS

(U//FOUO) GOPHERSET – Operational Schematic

(TS//SI//REL) Modern SIM cards (Phase 2+) have an application program interface known as the SIM Toolkit (STK). The STK has a suite of proactive commands that allow the SIM card to issue commands and make requests to the handset. GOPHERSET uses STK commands to retrieve the requested information and to exfiltrate data via SMS. After the GOPHERSET file is compiled, the program is loaded onto the SIM card using either a Universal Serial Bus (USB) smartcard reader or via over-the-air provisioning. In both cases, keys to the card may be required to install the application depending on the service provider's security configuration.

Unit Cost: $0

Status: (U//FOUO) Released. Has not been deployed.

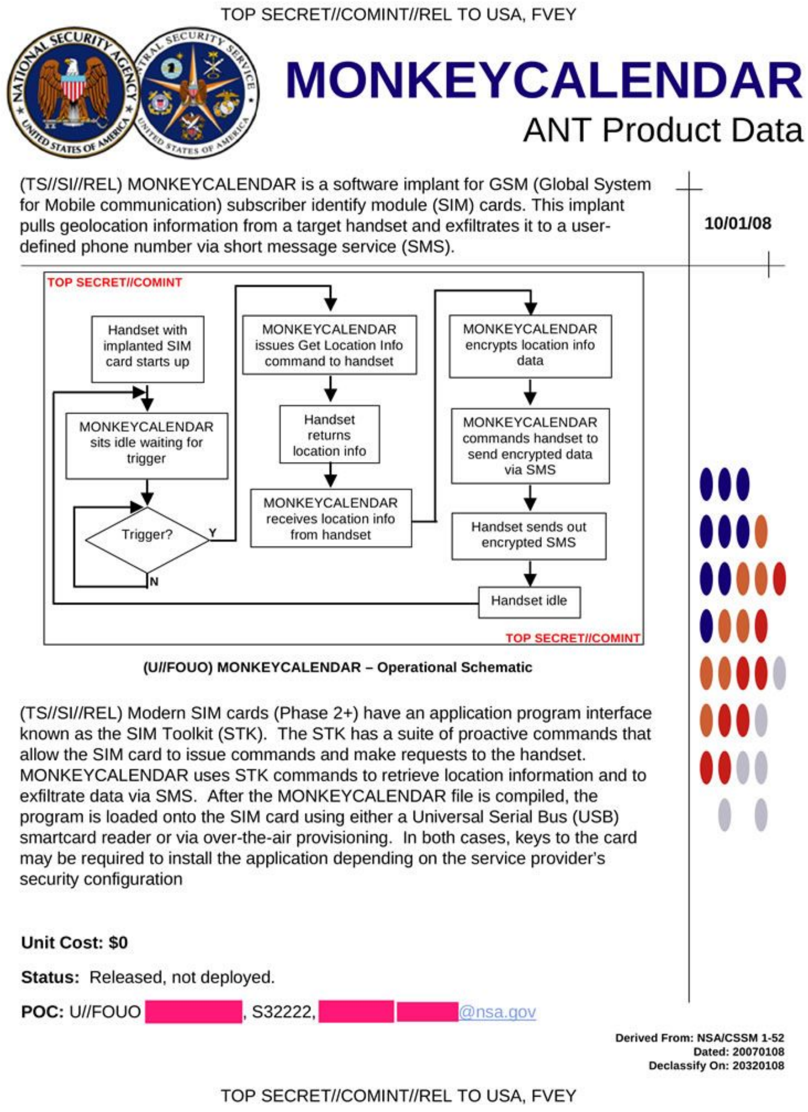
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.30 – GOPHERSET. A software backdoor that is potentially installable remotely or with physical access on subscriber identity module (SIM) cards that are used in cellular handsets. The backdoor is implemented using the SIM toolkit. Covert command and control with communication is provided through the use of encrypted SMS with an unspecified protocol. See MONKEYCALENDAR as described in Section 4.6.31.

Figure 4.57: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

MONKEYCALENDAR

ANT Product Data

(TS//SI//REL) MONKEYCALENDAR is a software implant for GSM (Global System for Mobile communication) subscriber identify module (SIM) cards. This implant pulls geolocation information from a target handset and exfiltrates it to a user-defined phone number via short message service (SMS).

10/01/08

TOP SECRET//COMINT
Handset with implanted SIM card starts up
MONKEYCALENDAR sits idle waiting for trigger
Trigger?
Y
N
MONKEYCALENDAR issues Get Location Info command to handset
Handset returns location info
MONKEYCALENDAR receives location info from handset
MONKEYCALENDAR encrypts location info data
MONKEYCALENDAR commands handset to send encrypted data via SMS
Handset sends out encrypted SMS
Handset idle
TOP SECRET//COMINT

(U//FOUO) MONKEYCALENDAR – Operational Schematic

(TS//SI//REL) Modern SIM cards (Phase 2+) have an application program interface known as the SIM Toolkit (STK). The STK has a suite of proactive commands that allow the SIM card to issue commands and make requests to the handset. MONKEYCALENDAR uses STK commands to retrieve location information and to exfiltrate data via SMS. After the MONKEYCALENDAR file is compiled, the program is loaded onto the SIM card using either a Universal Serial Bus (USB) smartcard reader or via over-the-air provisioning. In both cases, keys to the card may be required to install the application depending on the service provider's security configuration

Unit Cost: $0

Status: Released, not deployed.

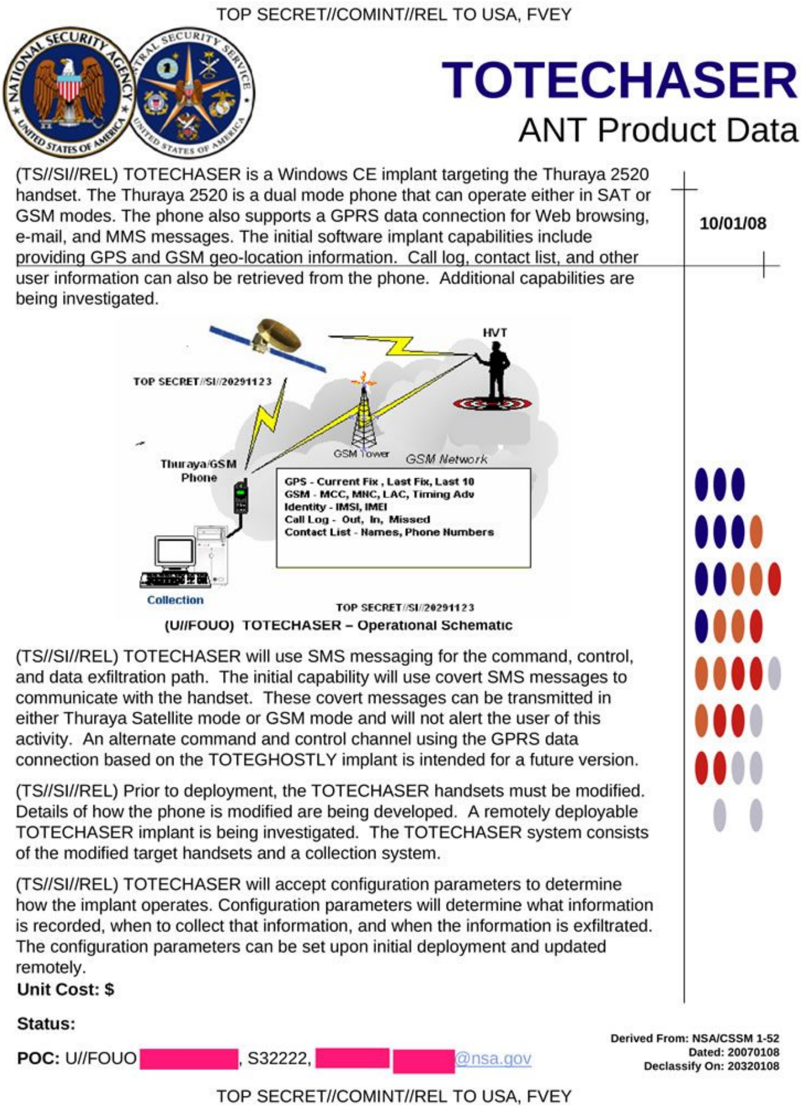
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.31 – MONKEYCALENDAR. A software backdoor that is potentially installable remotely or with physical access on SIM cards that are used in cellular handsets. The backdoor is implemented using the SIM toolkit. Covert command and control with communication provided through the use of encrypted SMS with an unspecified protocol. See GOPHERSET as described in Section 4.6.30.

Figure 4.58: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

TOTECHASER

ANT Product Data

(TS//SI//REL) TOTECHASER is a Windows CE implant targeting the Thuraya 2520 handset. The Thuraya 2520 is a dual mode phone that can operate either in SAT or GSM modes. The phone also supports a GPRS data connection for Web browsing, e-mail, and MMS messages. The initial software implant capabilities include providing GPS and GSM geo-location information. Call log, contact list, and other user information can also be retrieved from the phone. Additional capabilities are being investigated.

10/01/08

GPS - Current Fix , Last Fix, Last 10
GSM - MCC, MNC, LAC, Timing Adv
Identity - IMSI, IMEI
Call Log - Out, In, Missed
Contact List - Names, Phone Numbers

(U//FOUO) TOTECHASER – Operational Schematic

(TS//SI//REL) TOTECHASER will use SMS messaging for the command, control, and data exfiltration path. The initial capability will use covert SMS messages to communicate with the handset. These covert messages can be transmitted in either Thuraya Satellite mode or GSM mode and will not alert the user of this activity. An alternate command and control channel using the GPRS data connection based on the TOTEGHOSTLY implant is intended for a future version.

(TS//SI//REL) Prior to deployment, the TOTECHASER handsets must be modified. Details of how the phone is modified are being developed. A remotely deployable TOTECHASER implant is being investigated. The TOTECHASER system consists of the modified target handsets and a collection system.

(TS//SI//REL) TOTECHASER will accept configuration parameters to determine how the implant operates. Configuration parameters will determine what information is recorded, when to collect that information, and when the information is exfiltrated. The configuration parameters can be set upon initial deployment and updated remotely.

Unit Cost: $

Status:

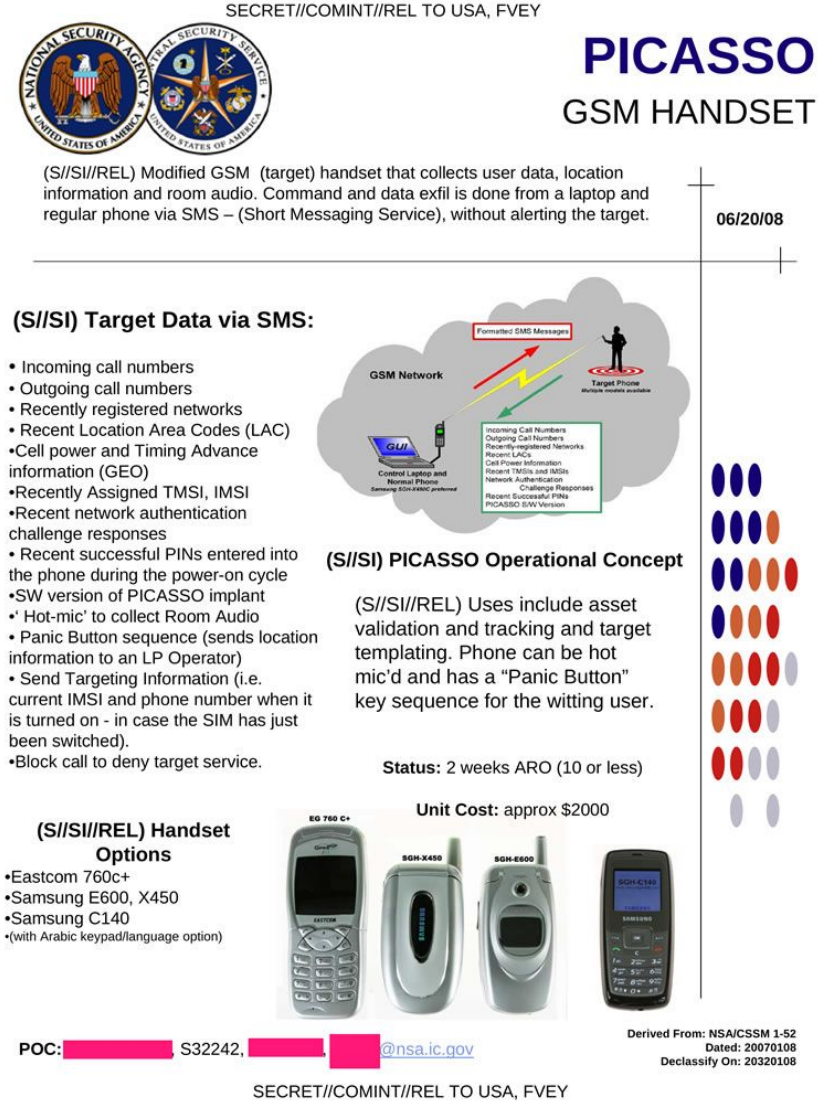
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.32 – TOTECHASER. A Windows CE backdoor installed on a target mobile satellite telephone (Thuraya) with physical access such as interdiction or a black bag [Wik21c] operation. Uses SMS and GPRS data to exfiltrate data from the telephone.

Figure 4.59: Courtesy of Der Spiegel [AHS13].



SECRET//COMINT//REL TO USA, FVEY

PICASSO

GSM HANDSET

(S//SI//REL) Modified GSM (target) handset that collects user data, location information and room audio. Command and data exfil is done from a laptop and regular phone via SMS – (Short Messaging Service), without alerting the target.

06/20/08

(S//SI) Target Data via SMS:

- Incoming call numbers
- Outgoing call numbers
- Recently registered networks
- Recent Location Area Codes (LAC)
- Cell power and Timing Advance information (GEO)
- Recently Assigned TMSI, IMSI
- Recent network authentication challenge responses
- Recent successful PINs entered into the phone during the power-on cycle
- SW version of PICASSO implant
- ‘ Hot-mic’ to collect Room Audio
- Panic Button sequence (sends location information to an LP Operator)
- Send Targeting Information (i.e. current IMSI and phone number when it is turned on - in case the SIM has just been switched).
- Block call to deny target service.

Formatted SMS Messages
GSM Network
Target Phone
Multiple models available
GUI
Control Laptop and Normal Phone
Samsung SGH-X450C preferred
Incoming Call Numbers
Outgoing Call Numbers
Recently-registered Networks
Recent LACs
Cell Power Information
Recent TMSIs and IMSIs
Network Authentication Challenge Responses
Recent Successful PINs
PICASSO S/W Version

(S//SI) PICASSO Operational Concept

(S//SI//REL) Uses include asset validation and tracking and target templating. Phone can be hot mic'd and has a “Panic Button” key sequence for the witting user.

Status: 2 weeks ARO (10 or less)

Unit Cost: approx $2000

(S//SI//REL) Handset Options

- Eastcom 760c+
- Samsung E600, X450
- Samsung C140
- (with Arabic keypad/language option)

EG 760 C+
SGH-X450
SGH-E600

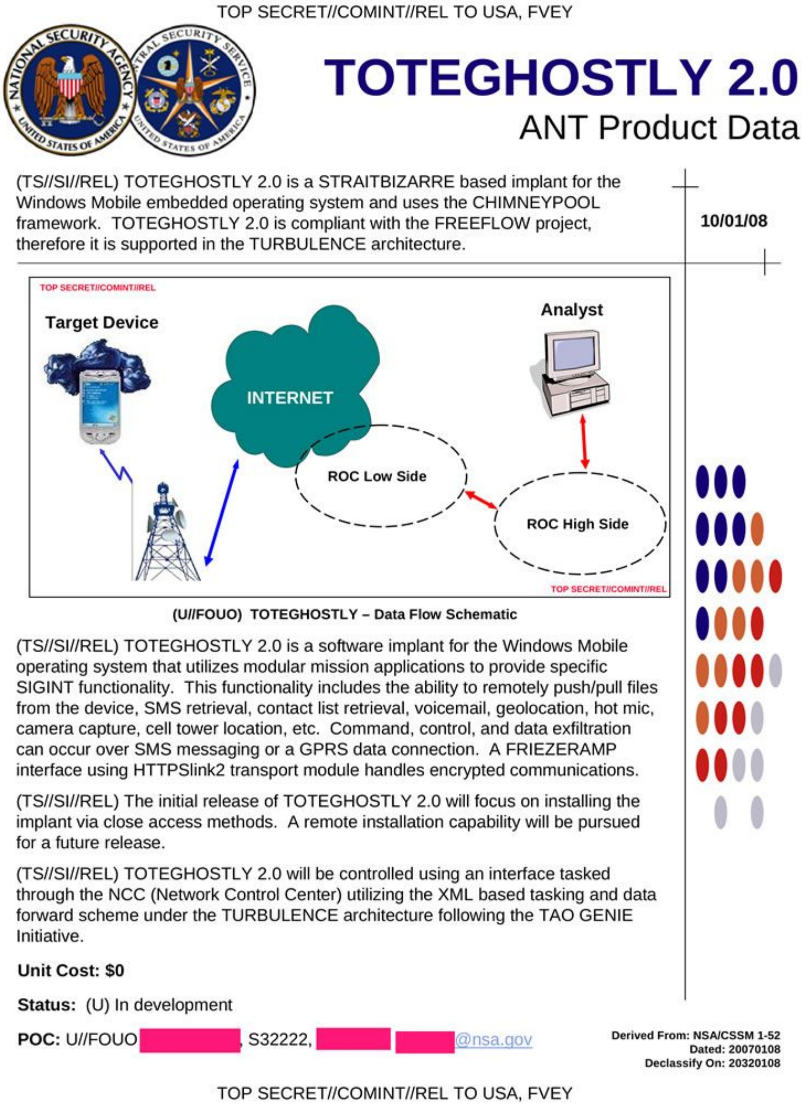
POC: [redacted], S32242, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

SECRET//COMINT//REL TO USA, FVEY

4.6.33 – PICASSO. A series of customized off-the-shelf Samsung and Eastcom GSM cellular telephones that are backdoored for surveillance. Optional Arabic keypad and OS localization. Encrypted SMS control channel using unknown cryptographic protocol.

Figure 4.60: Courtesy of Der Spiegel [AHS13].


TOP SECRET//COMINT//REL
Target Device
INTERNET
Analyst
ROC Low Side
ROC High Side
TOP SECRET//COMINT//REL

TOP SECRET//COMINT//REL TO USA, FVEY

TOTEGHOSTLY 2.0

ANT Product Data

(TS//SI//REL) TOTEGHOSTLY 2.0 is a STRAITBIZARRE based implant for the Windows Mobile embedded operating system and uses the CHIMNEYPOOL framework. TOTEGHOSTLY 2.0 is compliant with the FREEFLOW project, therefore it is supported in the TURBULENCE architecture.

10/01/08

(U//FOUO) TOTEGHOSTLY – Data Flow Schematic

(TS//SI//REL) TOTEGHOSTLY 2.0 is a software implant for the Windows Mobile operating system that utilizes modular mission applications to provide specific SIGINT functionality. This functionality includes the ability to remotely push/pull files from the device, SMS retrieval, contact list retrieval, voicemail, geolocation, hot mic, camera capture, cell tower location, etc. Command, control, and data exfiltration can occur over SMS messaging or a GPRS data connection. A FRIEZERAMP interface using HTTPSlink2 transport module handles encrypted communications.

(TS//SI//REL) The initial release of TOTEGHOSTLY 2.0 will focus on installing the implant via close access methods. A remote installation capability will be pursued for a future release.

(TS//SI//REL) TOTEGHOSTLY 2.0 will be controlled using an interface tasked through the NCC (Network Control Center) utilizing the XML based tasking and data forward scheme under the TURBULENCE architecture following the TAO GENIE Initiative.

Unit Cost: $0

Status: (U) In development

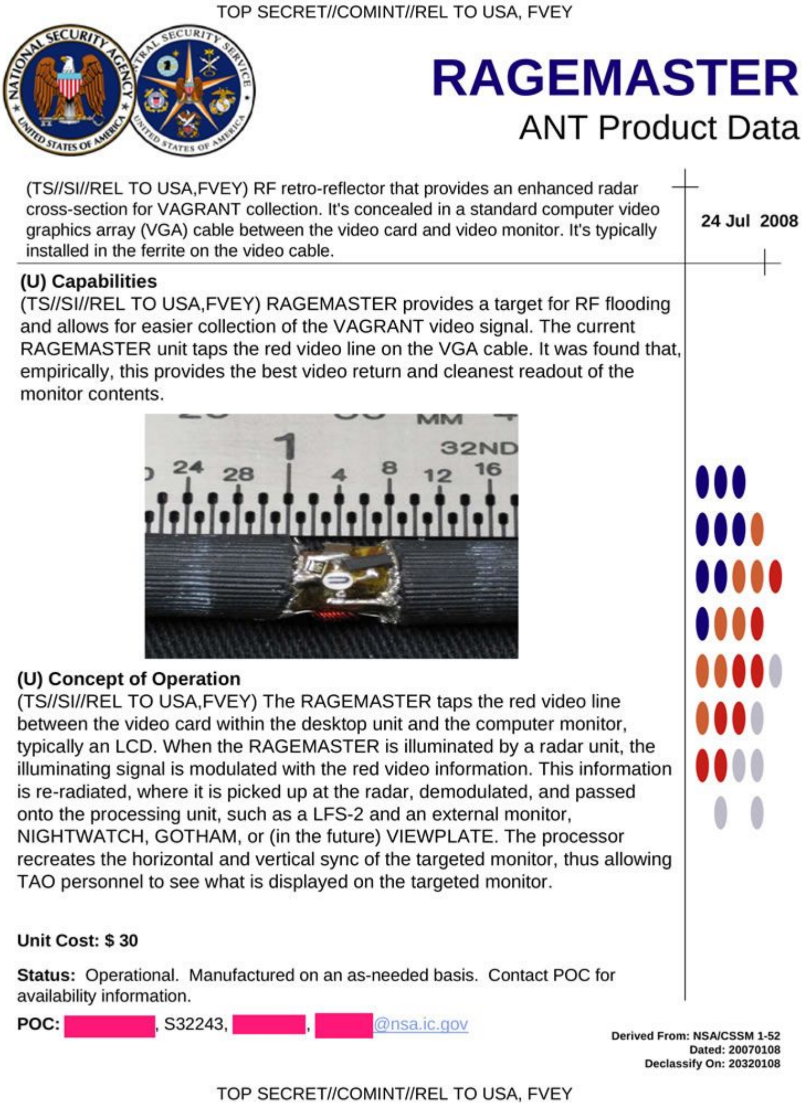
POC: U//FOUO [redacted], S32222, [redacted] [redacted]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.34 – TOTEGHOSTLY 2.0. A Windows Mobile backdoor installed on a target mobile telephone with physical access such as interdiction or a black bag operation. Uses SMS and GPRS data to exfiltrate data from the telephone.

Figure 4.61: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

RAGEMASTER

ANT Product Data

(TS//SI//REL TO USA,FVEY) RF retro-reflector that provides an enhanced radar cross-section for VAGRANT collection. It's concealed in a standard computer video graphics array (VGA) cable between the video card and video monitor. It's typically installed in the ferrite on the video cable.

24 Jul 2008

(U) Capabilities

(TS//SI//REL TO USA,FVEY) RAGEMASTER provides a target for RF flooding and allows for easier collection of the VAGRANT video signal. The current RAGEMASTER unit taps the red video line on the VGA cable. It was found that, empirically, this provides the best video return and cleanest readout of the monitor contents.

(U) Concept of Operation

(TS//SI//REL TO USA,FVEY) The RAGEMASTER taps the red video line between the video card within the desktop unit and the computer monitor, typically an LCD. When the RAGEMASTER is illuminated by a radar unit, the illuminating signal is modulated with the red video information. This information is re-radiated, where it is picked up at the radar, demodulated, and passed onto the processing unit, such as a LFS-2 and an external monitor, NIGHTWATCH, GOTHAM, or (in the future) VIEWPLATE. The processor recreates the horizontal and vertical sync of the targeted monitor, thus allowing TAO personnel to see what is displayed on the targeted monitor.

Unit Cost: $ 30

Status: Operational. Manufactured on an as-needed basis. Contact POC for availability information.

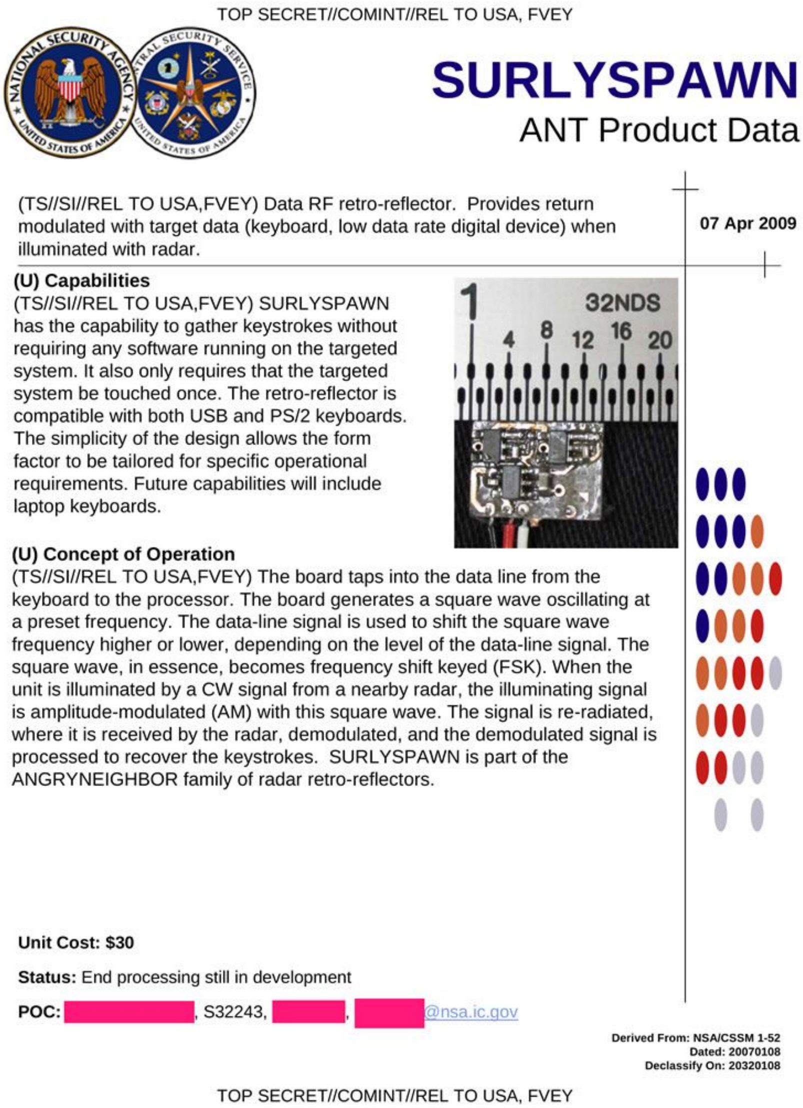
POC: [redacted], S32243, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.35 – RAGEMASTER. A radio frequency retro-reflector for use with a CTX4000 CW radar generator as shown in Section 4.6.13 and VAGRANT as shown in Section 4.6.13. RAGEMASTER is an entirely passive electronic device that is hidden inside a target's video monitor cable. When remotely powered by a CW radar generator, the video signal of the monitor is visible to an operator of VAGRANT. RAGEMASTER is part of the ANGRYNEIGHBOR family of backdoors as seen in Sections 4.6.14, 4.6.16, 4.6.36, and 4.6.17.

Figure 4.62: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

SURLYSPAWN

ANT Product Data

(TS//SI//REL TO USA,FVEY) Data RF retro-reflector. Provides return modulated with target data (keyboard, low data rate digital device) when illuminated with radar.

07 Apr 2009

(U) Capabilities

(TS//SI//REL TO USA,FVEY) SURLYSPAWN has the capability to gather keystrokes without requiring any software running on the targeted system. It also only requires that the targeted system be touched once. The retro-reflector is compatible with both USB and PS/2 keyboards. The simplicity of the design allows the form factor to be tailored for specific operational requirements. Future capabilities will include laptop keyboards.

(U) Concept of Operation

(TS//SI//REL TO USA,FVEY) The board taps into the data line from the keyboard to the processor. The board generates a square wave oscillating at a preset frequency. The data-line signal is used to shift the square wave frequency higher or lower, depending on the level of the data-line signal. The square wave, in essence, becomes frequency shift keyed (FSK). When the unit is illuminated by a CW signal from a nearby radar, the illuminating signal is amplitude-modulated (AM) with this square wave. The signal is re-radiated, where it is received by the radar, demodulated, and the demodulated signal is processed to recover the keystrokes. SURLYSPAWN is part of the ANGRYNEIGHBOR family of radar retro-reflectors.

Unit Cost: $30

Status: End processing still in development

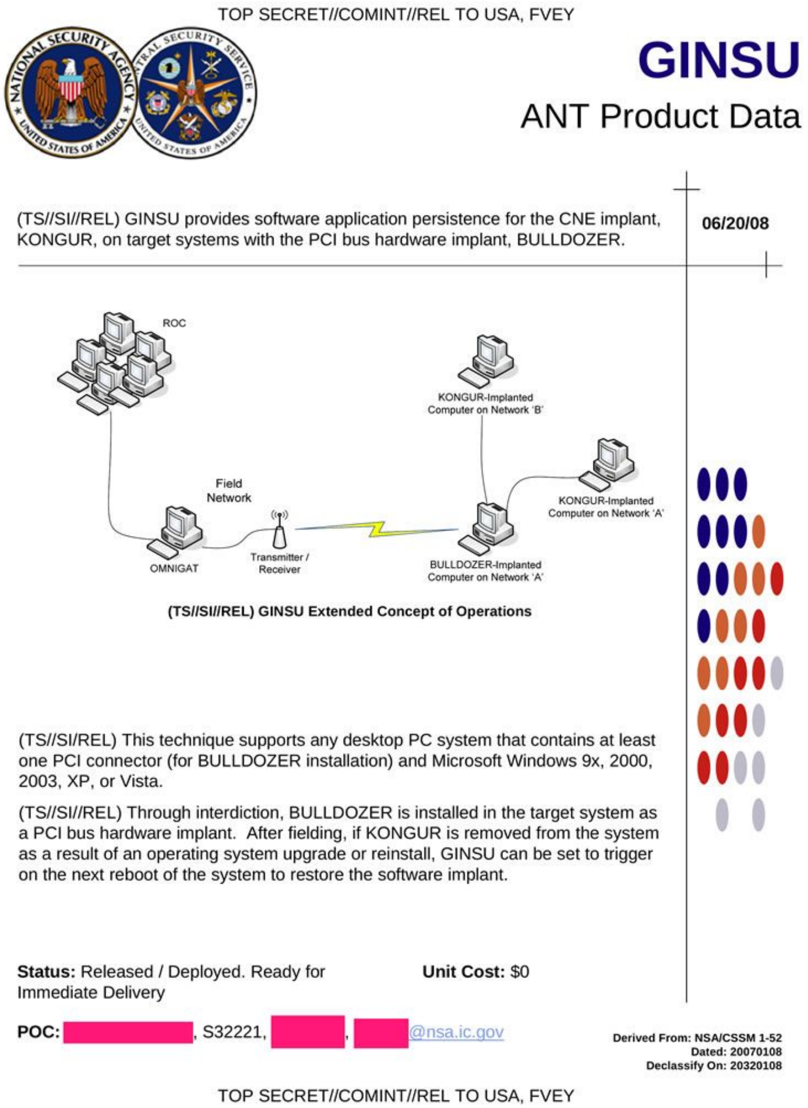
POC: [redacted], S32243, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.36 – SURLYSPAWN. A radar retro-reflector keylogger working with USB and PS2 keyboards as part of the ANGRYNEIGHBOR family of backdoors as seen in Sections 4.6.14, 4.6.16, 4.6.15, 4.6.35, and 4.6.17.

Figure 4.63: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

GINSU

ANT Product Data

(TS//SI//REL) GINSU provides software application persistence for the CNE implant, KONGUR, on target systems with the PCI bus hardware implant, BULLDOZER.

06/20/08

ROC
KONGUR-Implanted Computer on Network 'B'
KONGUR-Implanted Computer on Network 'A'
Field Network
OMNIGAT
Transmitter / Receiver
BULLDOZER-Implanted Computer on Network 'A'

(TS//SI//REL) GINSU Extended Concept of Operations

(TS//SI/REL) This technique supports any desktop PC system that contains at least one PCI connector (for BULLDOZER installation) and Microsoft Windows 9x, 2000, 2003, XP, or Vista.

(TS//SI//REL) Through interdiction, BULLDOZER is installed in the target system as a PCI bus hardware implant. After fielding, if KONGUR is removed from the system as a result of an operating system upgrade or reinstall, GINSU can be set to trigger on the next reboot of the system to restore the software implant.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0

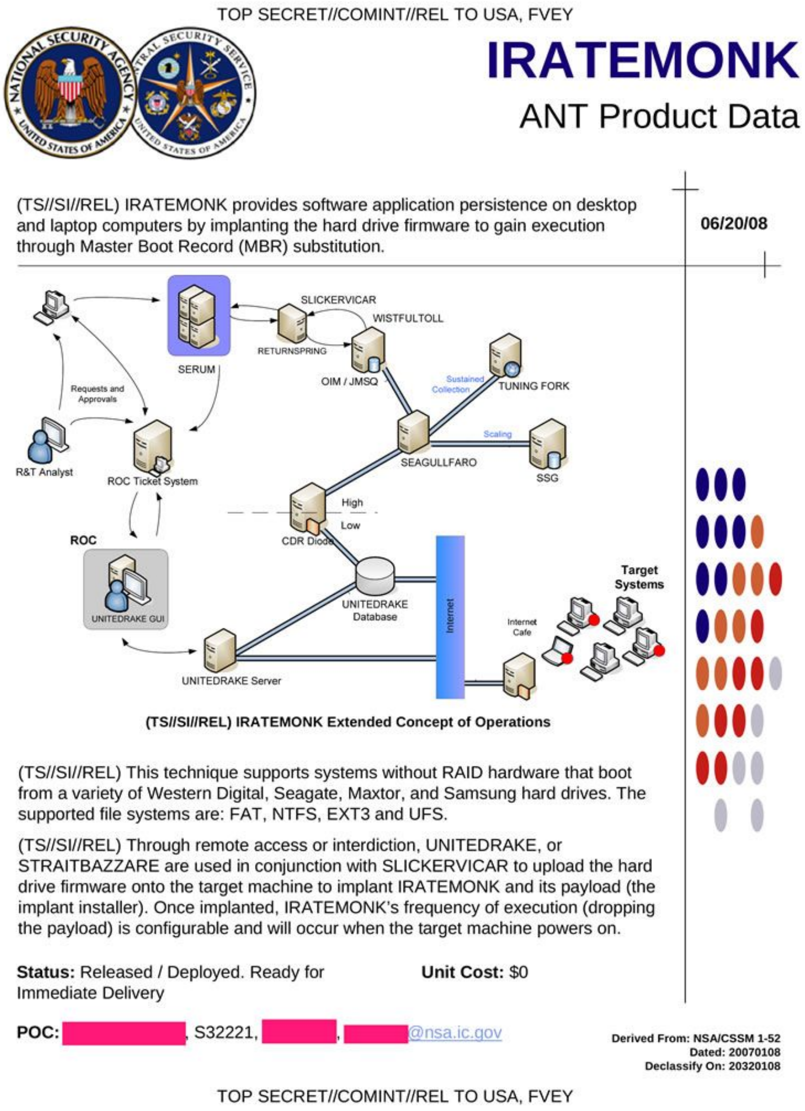
POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.37 – GINSU. A software component for use with the BULLDOZER PCI hardware implant and the KONGUR implant. GINSU installs and makes persistent the KONGUR rootkit on targeted Microsoft Windows systems. The BULLDOZER PCI hardware backdoor is installed through physical access to the system such as through interdiction or during black bag operations.

Figure 4.64: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

IRATEMONK

ANT Product Data

(TS//SI//REL) IRATEMONK provides software application persistence on desktop and laptop computers by implanting the hard drive firmware to gain execution through Master Boot Record (MBR) substitution.

06/20/08

SLICKERVICAR
WISTFULTOLL
RETURNSPRING
SERUM
OIM / JMSQ
Sustained Collection
TUNING FORK
Requests and Approvals
Scaling
SEAGULLFARO
SSG
R&T Analyst
ROC Ticket System
High
Low
ROC
CDR Diode
UNITEDRAKE GUI
UNITEDRAKE Database
Internet
Target Systems
Internet Cafe
UNITEDRAKE Server

(TS//SI//REL) IRATEMONK Extended Concept of Operations

(TS//SI//REL) This technique supports systems without RAID hardware that boot from a variety of Western Digital, Seagate, Maxtor, and Samsung hard drives. The supported file systems are: FAT, NTFS, EXT3 and UFS.

(TS//SI//REL) Through remote access or interdiction, UNITEDRAKE, or STRAITBAZZARE are used in conjunction with SLICKERVICAR to upload the hard drive firmware onto the target machine to implant IRATEMONK and its payload (the implant installer). Once implanted, IRATEMONK's frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0

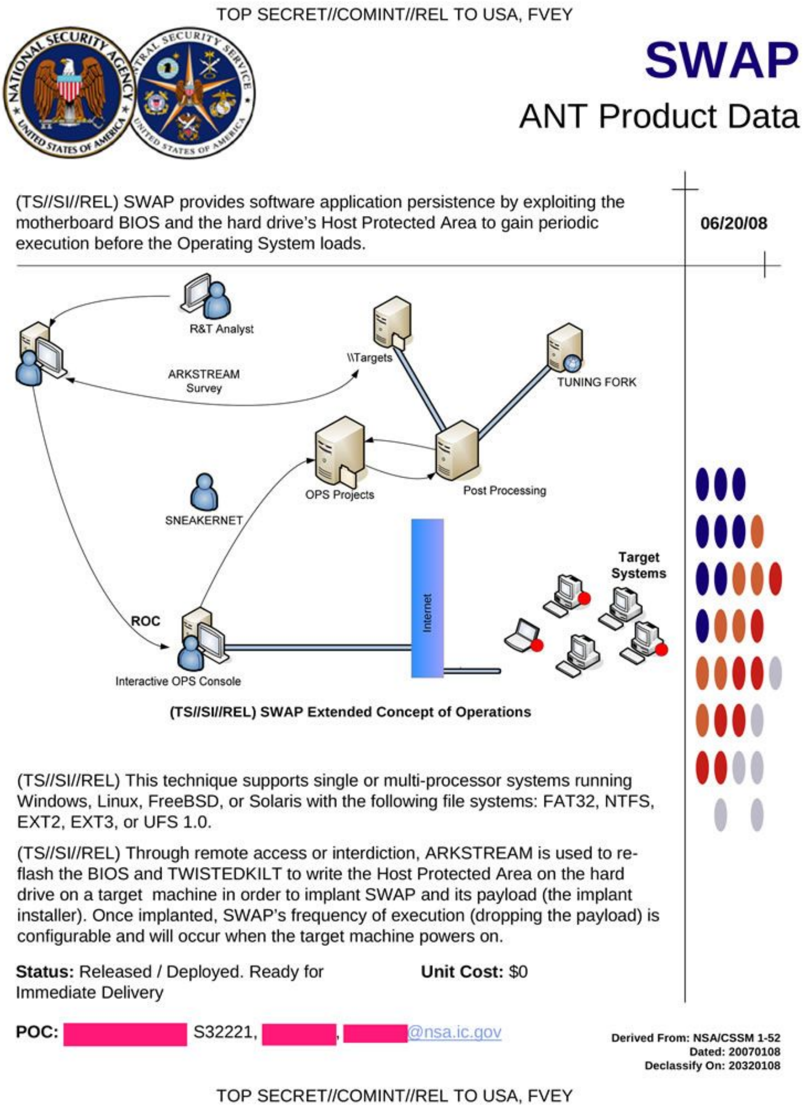
POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.38 – IRATEMONK. A backdoor that runs in the firmware of a computer hard drive. Supports Samsung, Seagate, Maxtor, and Western Digital drives. Installation of the backdoor is performed remotely through exploitation or through physical access such as with interdiction. IRATEMONK replaces the Master Boot Record (MBR) upon boot.

Figure 4.65: Courtesy of Der Spiegel [AHS13].


R&T Analyst
\\Targets
TUNING FORK
ARKSTREAM
Survey
OPS Projects
Post Processing
SNEAKERNET
Target
Systems
ROC
Internet
Interactive OPS Console

TOP SECRET//COMINT//REL TO USA, FVEY

SWAP

ANT Product Data

(TS//SI//REL) SWAP provides software application persistence by exploiting the motherboard BIOS and the hard drive's Host Protected Area to gain periodic execution before the Operating System loads.

06/20/08

(TS//SI//REL) SWAP Extended Concept of Operations

(TS//SI//REL) This technique supports single or multi-processor systems running Windows, Linux, FreeBSD, or Solaris with the following file systems: FAT32, NTFS, EXT2, EXT3, or UFS 1.0.

(TS//SI//REL) Through remote access or interdiction, ARKSTREAM is used to re-flash the BIOS and TWISTEDKILT to write the Host Protected Area on the hard drive on a target machine in order to implant SWAP and its payload (the implant installer). Once implanted, SWAP's frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0

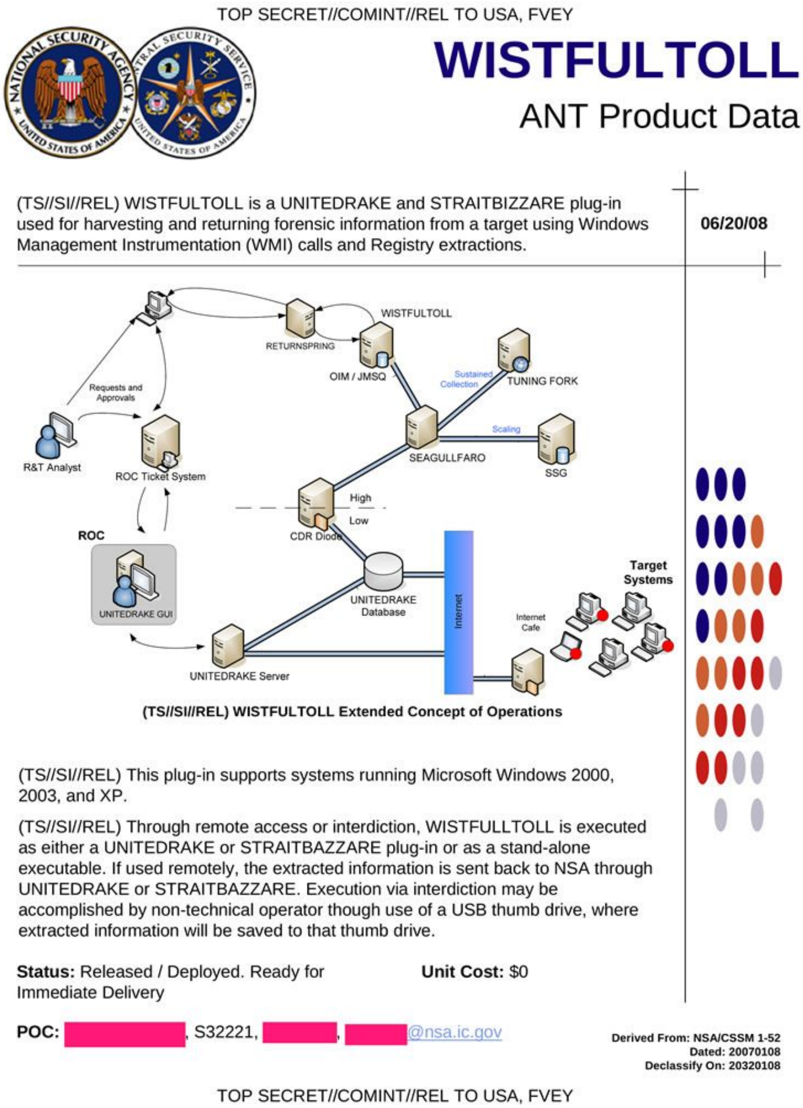
POC: [redacted] S32221, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.39 – SWAP. A persistent BIOS backdoor in the Host Protected Area (HPA) of a hard drive that targets many different operating systems. ARKSTREAM reflashes the targeted system's BIOS and TWISTEDKILT is used to write a persistent backdoor to the HPA of the hard drive. See also DEITYBOUNCE in Section 4.6.2 for additional uses of ARKSTREAM.

Figure 4.66: Courtesy of Der Spiegel [AHS13].


WISTFULTOLL
RETURNSPRING
OIM / JMSQ
Sustained Collection
TUNING FORK
Requests and Approvals
R&T Analyst
ROC Ticket System
SEAGULLFARO
Scaling
SSG
ROC
High
Low
CDR Diode
UNITEDRAKE GUI
UNITEDRAKE Database
Internet
Target Systems
Internet Cafe
UNITEDRAKE Server

TOP SECRET//COMINT//REL TO USA, FVEY

WISTFULTOLL

ANT Product Data

(TS//SI//REL) WISTFULTOLL is a UNITEDRAKE and STRAITBIZZARE plug-in used for harvesting and returning forensic information from a target using Windows Management Instrumentation (WMI) calls and Registry extractions.

06/20/08

(TS//SI//REL) WISTFULTOLL Extended Concept of Operations

(TS//SI//REL) This plug-in supports systems running Microsoft Windows 2000, 2003, and XP.

(TS//SI//REL) Through remote access or interdiction, WISTFULLTOLL is executed as either a UNITEDRAKE or STRAITBAZZARE plug-in or as a stand-alone executable. If used remotely, the extracted information is sent back to NSA through UNITEDRAKE or STRAITBAZZARE. Execution via interdiction may be accomplished by non-technical operator though use of a USB thumb drive, where extracted information will be saved to that thumb drive.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0

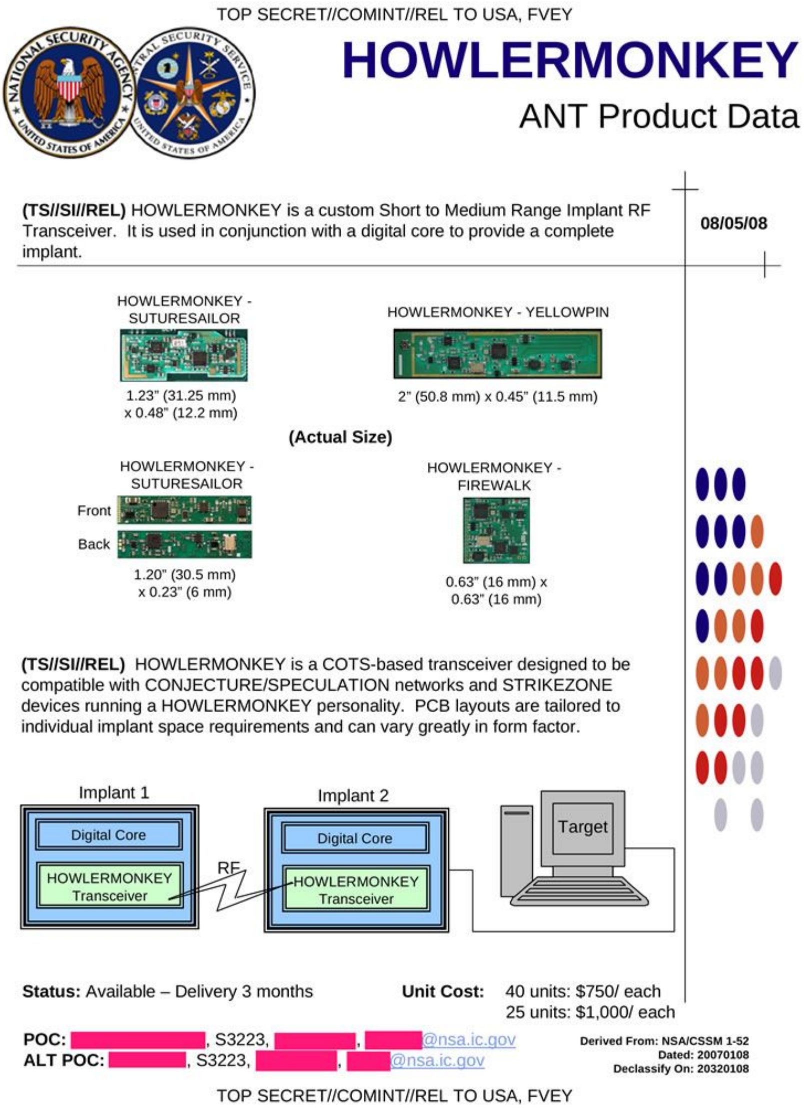
POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.40 – WISTFULTOLL. A forensic data-harvesting software module for other implants such as UNITEDRAKE as seen in Section 4.6.50 and STRAITBIZZARE as seen in Section 4.6.50. Windows Management Instrumentation (WMI) and Windows Registry extraction are claimed features for Microsoft Windows 2000, XP, and Server 2003. Exfiltration of data over the network is possible using UNITEDRAKE or STRAITBIZZARE exfiltration communication channels, as well as with physical access such as interdiction or black bag operations using a USB mass storage device. Emphasized is the ability for "*non-technical operators*" to use this tool during interdiction.

Figure 4.67: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

HOWLERMONKEY

ANT Product Data

(TS//SI//REL) HOWLERMONKEY is a custom Short to Medium Range Implant RF Transceiver. It is used in conjunction with a digital core to provide a complete implant.

08/05/08

HOWLERMONKEY - SUTURESAILOR
1.23" (31.25 mm) x 0.48" (12.2 mm)
HOWLERMONKEY - YELLOWPIN
2" (50.8 mm) x 0.45" (11.5 mm)
(Actual Size)
HOWLERMONKEY - SUTURESAILOR
Front
Back
1.20" (30.5 mm) x 0.23" (6 mm)
HOWLERMONKEY - FIREWALK
0.63" (16 mm) x 0.63" (16 mm)

(TS//SI//REL) HOWLERMONKEY is a COTS-based transceiver designed to be compatible with CONJECTURE/SPECULATION networks and STRIKEZONE devices running a HOWLERMONKEY personality. PCB layouts are tailored to individual implant space requirements and can vary greatly in form factor.

Implant 1
Digital Core
HOWLERMONKEY Transceiver
RF
Implant 2
Digital Core
HOWLERMONKEY Transceiver
Target

Status: Available – Delivery 3 months

Unit Cost: 40 units: $750/ each
25 units: $1,000/ each

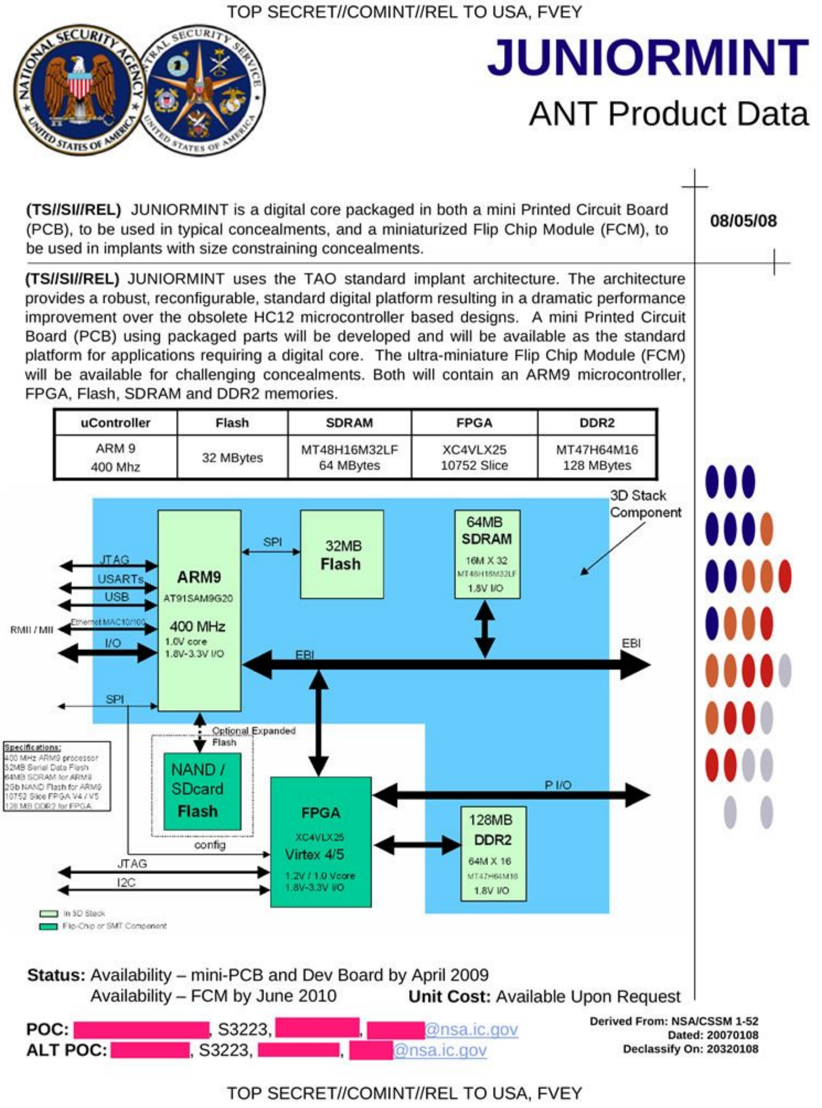
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.41 – HOWLERMONKEY. A family of short to medium range radio transceiver hardware implants. Used with other hardware backdoors such as FIREWALK as seen in Section 4.6.49. It is advertised to use a common commercially available transceiver, implying that related hardware backdoors likely have similar radio frequency signatures.

Figure 4.68: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

JUNIORMINT

ANT Product Data

08/05/08

(TS//SI//REL) JUNIORMINT is a digital core packaged in both a mini Printed Circuit Board (PCB), to be used in typical concealments, and a miniaturized Flip Chip Module (FCM), to be used in implants with size constraining concealments.

(TS//SI//REL) JUNIORMINT uses the TAO standard implant architecture. The architecture provides a robust, reconfigurable, standard digital platform resulting in a dramatic performance improvement over the obsolete HC12 microcontroller based designs. A mini Printed Circuit Board (PCB) using packaged parts will be developed and will be available as the standard platform for applications requiring a digital core. The ultra-miniature Flip Chip Module (FCM) will be available for challenging concealments. Both will contain an ARM9 microcontroller, FPGA, Flash, SDRAM and DDR2 memories.

uController	Flash	SDRAM	FPGA	DDR2
ARM 9 400 Mhz	32 MBytes	MT48H16M32LF 64 MBytes	XC4VLX25 10752 Slice	MT47H64M16 128 MBytes

3D Stack Component
JTAG
USARTs
USB
RMII / MII
Ethernet MAC10/100
I/O
SPI
ARM9
AT91SAM9G20
400 MHz
1.0V core
1.8V-3.3V I/O
SPI
32MB Flash
64MB SDRAM
16M X 32
MT48H16M32LF
1.8V I/O
EBI
EBI
Optional Expanded Flash
NAND / SDcard Flash
config
FPGA
XC4VLX25
Virtex 4/5
1.2V / 1.0 Vcore
1.8V-3.3V I/O
P I/O
128MB DDR2
64M X 16
MT47H64M16
1.8V I/O
JTAG
I2C
In 3D Stack
Flip-Chip or SMT Component

Specifications:
400 MHz ARM9 processor
32MB Serial Data Flash
64MB SDRAM for ARM9
2Gb NAND Flash for ARM9
10752 Slice FPGA V4 / V5
128 MB DDR2 for FPGA

Status: Availability – mini-PCB and Dev Board by April 2009
Availability – FCM by June 2010

Unit Cost: Available Upon Request

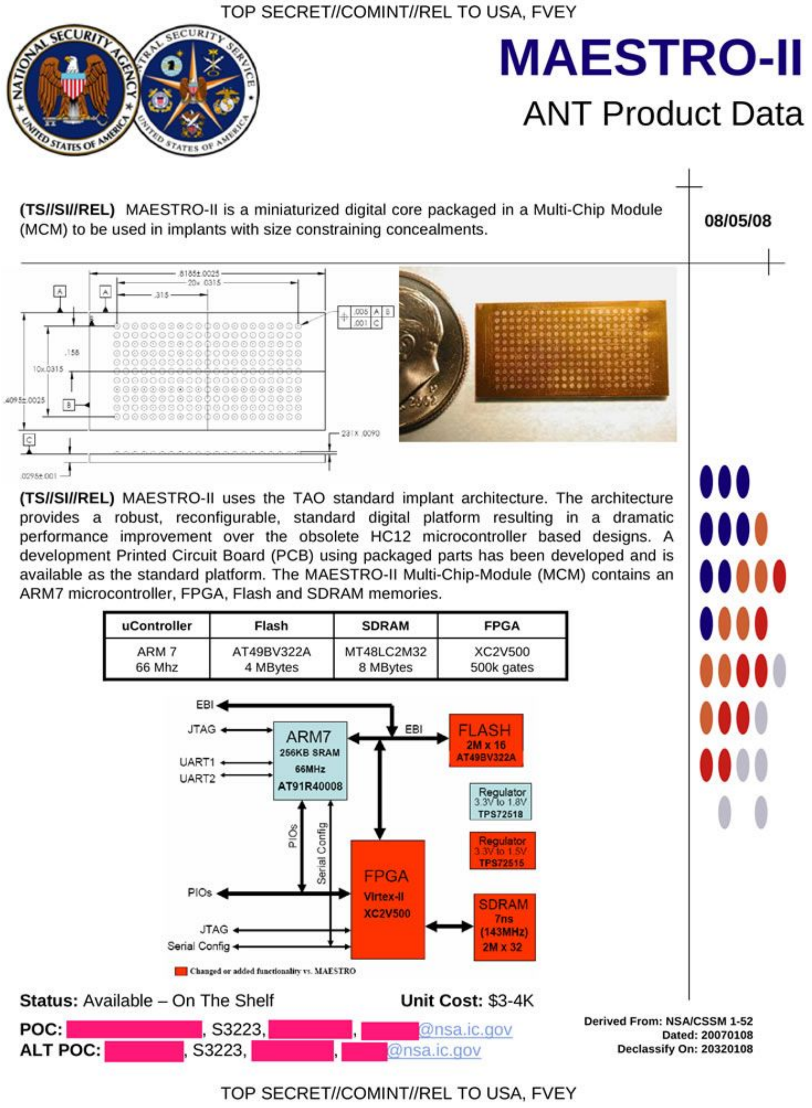
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.42 – JUNIORMINT. An ARM9 computer likely around the size of a United States penny similar to TRINITY as described in Section 4.6.45. JUNIORMINT has a 400 Mhz clock frequency. JUNIORMINT is used in tandem with other devices such as HOWLER-MONKEY in Section 4.6.41. JUNIORMINT is a replacement for HC12 microcontroller based designs and offers two sizes: Printed Circuit Board (PCB) or Flip Chip Module (FCM). The JUNIORMINT computer includes 128 MB of (Micron MT47H64M16) SDRAM, 32 MB of flash memory, and a 10752 Slice (Xilinx XC4VLX25) FPGA.

Figure 4.69: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

MAESTRO-II

ANT Product Data

08/05/08

(TS//SI//REL) MAESTRO-II is a miniaturized digital core packaged in a Multi-Chip Module (MCM) to be used in implants with size constraining concealments.

.8185±.0025
20x .0315
.315
A
A
.005 A B
.001 C
.158
10x.0315
.4095±.0025
B
C
.0295±.001
231X .0090

(TS//SI//REL) MAESTRO-II uses the TAO standard implant architecture. The architecture provides a robust, reconfigurable, standard digital platform resulting in a dramatic performance improvement over the obsolete HC12 microcontroller based designs. A development Printed Circuit Board (PCB) using packaged parts has been developed and is available as the standard platform. The MAESTRO-II Multi-Chip-Module (MCM) contains an ARM7 microcontroller, FPGA, Flash and SDRAM memories.

uController	Flash	SDRAM	FPGA
ARM 7 66 Mhz	AT49BV322A 4 MBytes	MT48LC2M32 8 MBytes	XC2V500 500k gates

EBI
JTAG
UART1
UART2
ARM7
256KB SRAM
66MHz
AT91R40008
EBI
FLASH
2M x 16
AT49BV322A
Regulator
3.3V to 1.8V
TPS72518
Regulator
3.3V to 1.5V
TPS72515
PIOs
Serial Config
PIOs
FPGA
Virtex-II
XC2V500
SDRAM
7ns
(143MHz)
2M x 32
JTAG
Serial Config
Changed or added functionality vs. MAESTRO

Status: Available – On The Shelf

Unit Cost: $3-4K

POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

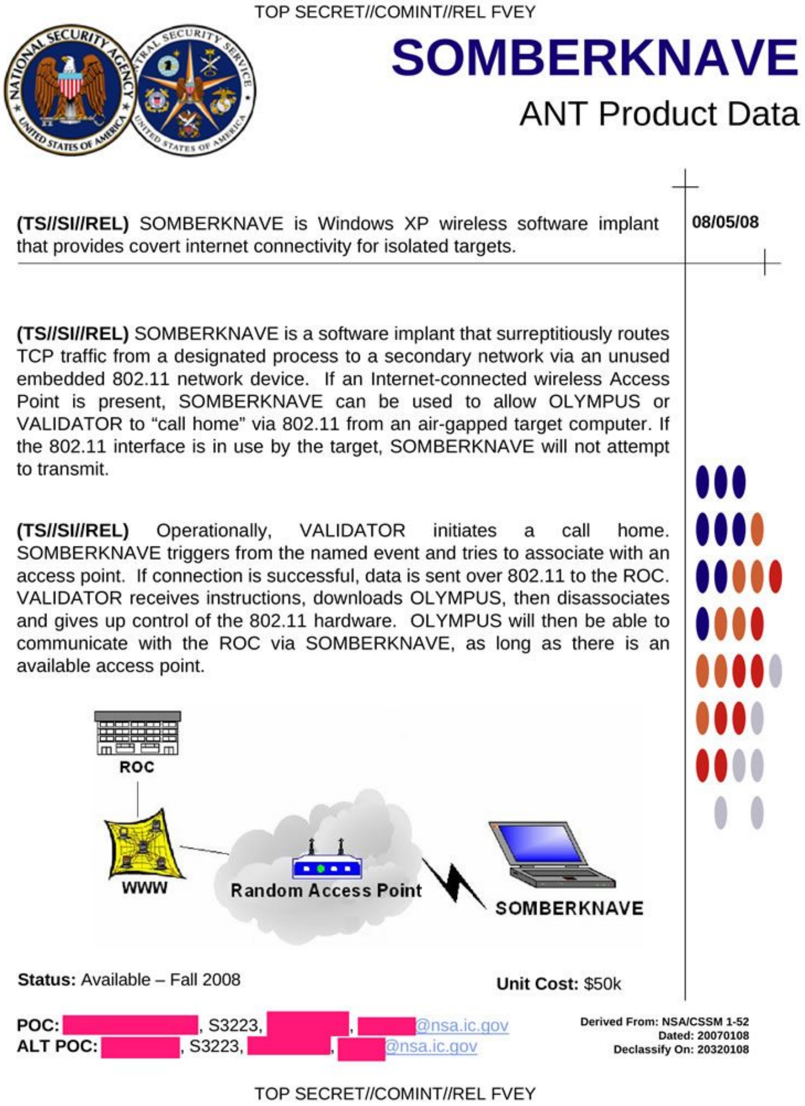
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.43 – MAESTRO-II. An ARM7 (Atmel AT91R40008) computer likely around the size of a United States dime coin similar to TRINITY as described in Section 4.6.45 and JUNIORMINT as described in Section 4.6.42. MAESTRO-II has a 66 Mhz clock frequency. Used in tandem with other devices such as HOWLERMONKEY as shown in Section 4.6.41. Replacement for HC12 microcontroller based designs. The MAESTRO-II computer includes 8 MB of (Micron MT48LCM32) SDRAM, 4 MB of flash memory (Atmel AT49BV322A), and a 500,000 gate (Xilinx XC2V500) Virtex-II FPGA.

Figure 4.70: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL FVEY

SOMBERKNAVE

ANT Product Data

(TS//SI//REL) SOMBERKNAVE is Windows XP wireless software implant that provides covert internet connectivity for isolated targets.

08/05/08

(TS//SI//REL) SOMBERKNAVE is a software implant that surreptitiously routes TCP traffic from a designated process to a secondary network via an unused embedded 802.11 network device. If an Internet-connected wireless Access Point is present, SOMBERKNAVE can be used to allow OLYMPUS or VALIDATOR to "call home" via 802.11 from an air-gapped target computer. If the 802.11 interface is in use by the target, SOMBERKNAVE will not attempt to transmit.

(TS//SI//REL) Operationally, VALIDATOR initiates a call home. SOMBERKNAVE triggers from the named event and tries to associate with an access point. If connection is successful, data is sent over 802.11 to the ROC. VALIDATOR receives instructions, downloads OLYMPUS, then disassociates and gives up control of the 802.11 hardware. OLYMPUS will then be able to communicate with the ROC via SOMBERKNAVE, as long as there is an available access point.

Status: Available – Fall 2008

Unit Cost: $50k

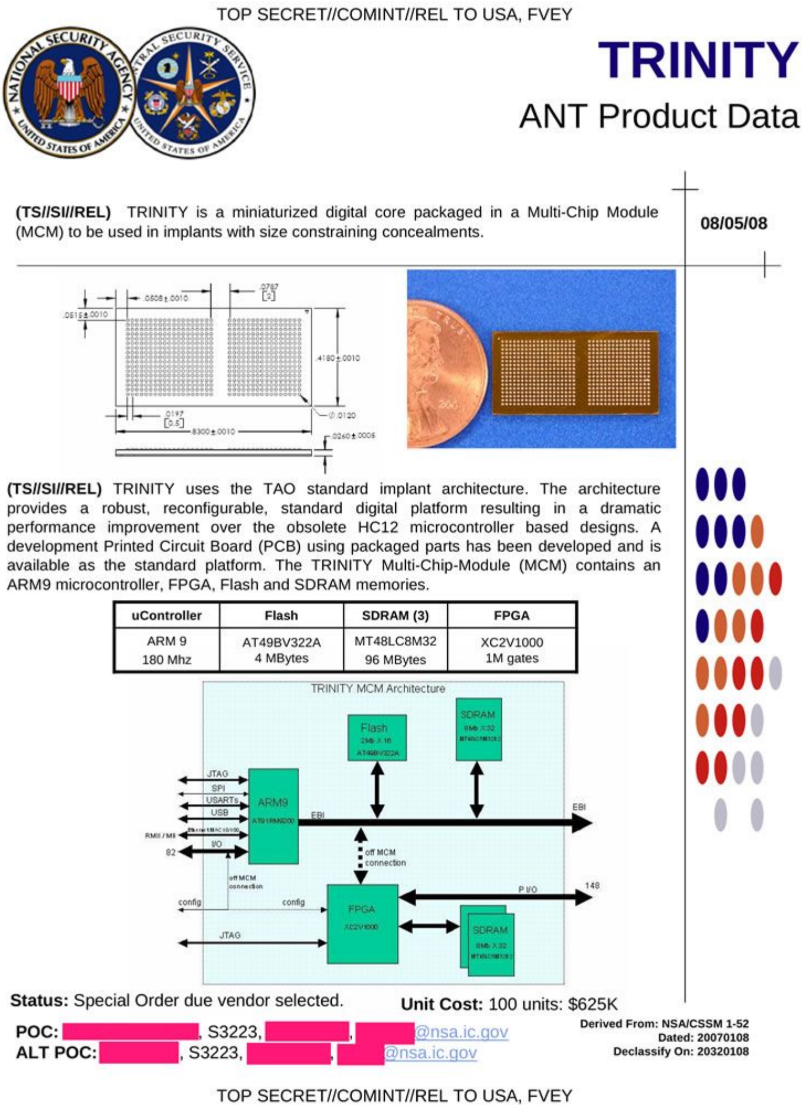
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL FVEY

4.6.44 – SOMBERKNAVE. A hardware implant for Windows XP that uses an unused wireless network device for covert exfiltration of data on an otherwise air-gapped computer. Used in conjunction with VALIDATOR in Section 4.6.50 to download OLYMPUS.

Figure 4.71: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

TRINITY

ANT Product Data

(TS//SI//REL) TRINITY is a miniaturized digital core packaged in a Multi-Chip Module (MCM) to be used in implants with size constraining concealments.

08/05/08

.0508±.0010
.0787
[2]
.0515±.0010
.4180±.0010
.0197
[0.5]
.8300±.0010
⌀.0120
.0260±.0005

(TS//SI//REL) TRINITY uses the TAO standard implant architecture. The architecture provides a robust, reconfigurable, standard digital platform resulting in a dramatic performance improvement over the obsolete HC12 microcontroller based designs. A development Printed Circuit Board (PCB) using packaged parts has been developed and is available as the standard platform. The TRINITY Multi-Chip-Module (MCM) contains an ARM9 microcontroller, FPGA, Flash and SDRAM memories.

uController	Flash	SDRAM (3)	FPGA
ARM 9 180 Mhz	AT49BV322A 4 MBytes	MT48LC8M32 96 MBytes	XC2V1000 1M gates

TRINITY MCM Architecture
Flash
2Mb X 16
AT49BV322A
SDRAM
8Mb X 32
JTAG
SPI
USARTs
USB
RMII / MII
I/O
82
ARM9
EBI
EBI
off MCM connection
off MCM connection
config
config
P I/O
148
FPGA
XC2V1000
SDRAM
8Mb X 32
JTAG

Status: Special Order due vendor selected.

Unit Cost: 100 units: $625K

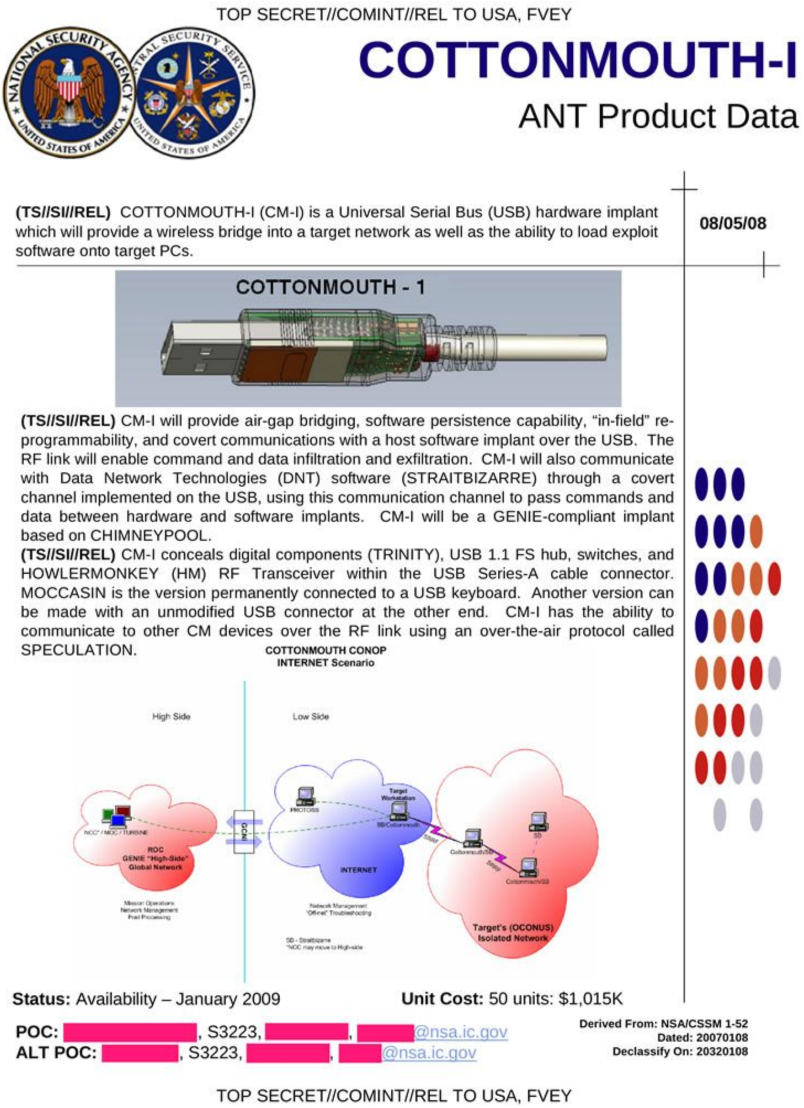
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.45 – TRINITY. An ARM9 computer roughly the size of a United States penny. TRINITY has a 180 Mhz clock frequency. TRINITY is used in tandem with other devices such as HOWLERMONKEY in Section 4.6.41. TRINITY is a replacement for HC12 microcontroller based designs. The TRINITY computer includes 96 MB of (Micron MT48LC8M32) SDRAM, 4 MB of (Atmel AT49BV322A) flash memory, and a one million gate (Xilinx XC2V1000) FPGA.

Figure 4.72: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

COTTONMOUTH-I

ANT Product Data

08/05/08

(TS//SI//REL) COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.

COTTONMOUTH - 1

(TS//SI//REL) CM-I will provide air-gap bridging, software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. The RF link will enable command and data infiltration and exfiltration. CM-I will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-I will be a GENIE-compliant implant based on CHIMNEYPOOL.

(TS//SI//REL) CM-I conceals digital components (TRINITY), USB 1.1 FS hub, switches, and HOWLERMONKEY (HM) RF Transceiver within the USB Series-A cable connector. MOCCASIN is the version permanently connected to a USB keyboard. Another version can be made with an unmodified USB connector at the other end. CM-I has the ability to communicate to other CM devices over the RF link using an over-the-air protocol called SPECULATION.

COTTONMOUTH CONOP
INTERNET Scenario

High Side
Low Side
NCC* / MOC / TURBINE
ROC
GENIE "High-Side" Global Network
Mission Operations
Network Management
Post Processing
GCNI
PROTOSS
Target Workstation
SB/Cottonmouth
INTERNET
Network Management
'Off-net' Troubleshooting
SD - Straitbizarre
*NOC may move to High-side
Cottonmouth/SB
SD
Cottonmouth/SD
Target's (OCONUS) Isolated Network

Status: Availability – January 2009

Unit Cost: 50 units: $1,015K

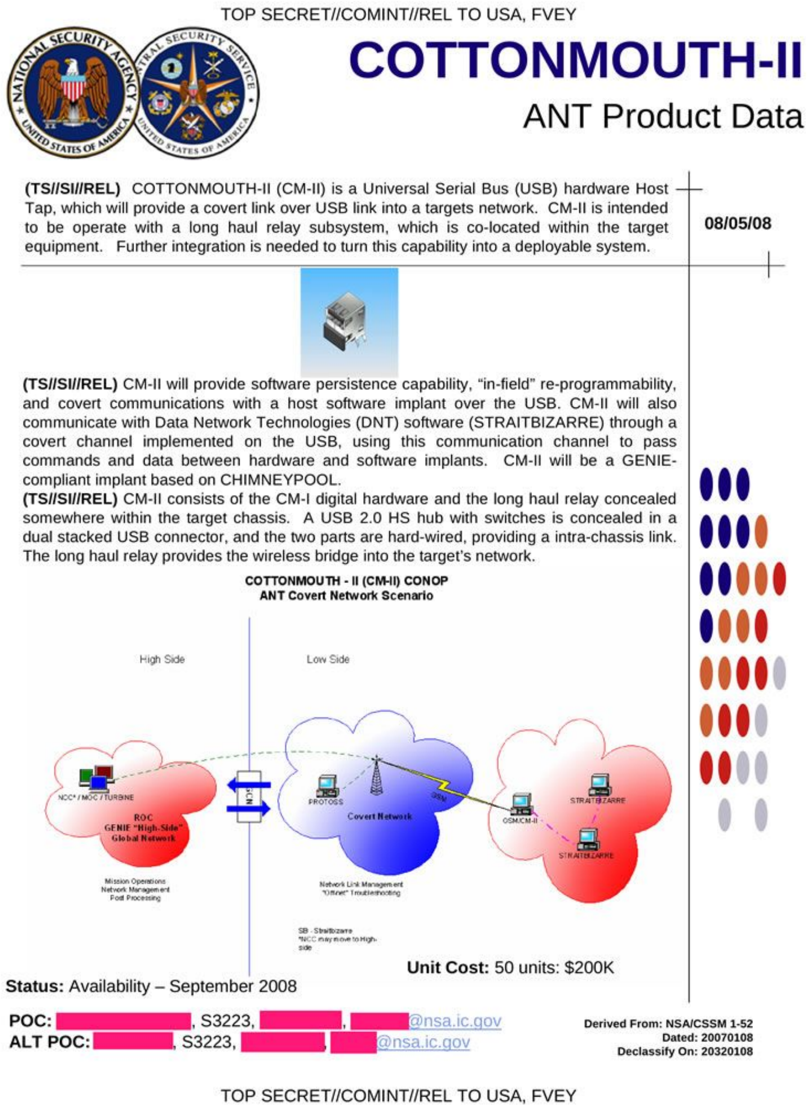
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.46 – COTTONMOUTH-I. An implant hidden inside a USB cable featuring a covert radio transceiver from the HOWLERMONKEY 4.6.41 family of implants. It uses an unknown digital radio protocol called SPECULATION. MOCCASIN is a variant for permanent connection to a USB keyboard. HOWLERMONKEY is a radio board that is paired with an embedded computer such as TRINITY in Section 4.6.45.

Figure 4.73: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

COTTONMOUTH-II

ANT Product Data

(TS//SI//REL) COTTONMOUTH-II (CM-II) is a Universal Serial Bus (USB) hardware Host Tap, which will provide a covert link over USB link into a targets network. CM-II is intended to be operate with a long haul relay subsystem, which is co-located within the target equipment. Further integration is needed to turn this capability into a deployable system.

08/05/08

(TS//SI//REL) CM-II will provide software persistence capability, “in-field” re-programmability, and covert communications with a host software implant over the USB. CM-II will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-II will be a GENIE-compliant implant based on CHIMNEYPOOL.

(TS//SI//REL) CM-II consists of the CM-I digital hardware and the long haul relay concealed somewhere within the target chassis. A USB 2.0 HS hub with switches is concealed in a dual stacked USB connector, and the two parts are hard-wired, providing a intra-chassis link. The long haul relay provides the wireless bridge into the target’s network.

COTTONMOUTH - II (CM-II) CONOP
ANT Covert Network Scenario

High Side
Low Side
NCC* / MOC / TURBINE
ROC
GENIE "High-Side" Global Network
Mission Operations
Network Management
Post Processing
GCN
PROTOSS
Covert Network
GSM
Network Link Management
"Offnet" Troubleshooting
OSM/CM-II
STRAITBIZARRE
STRAITBIZARRE
SB - Straitbizarre
*NCC may move to High-side

Unit Cost: 50 units: $200K

Status: Availability – September 2008

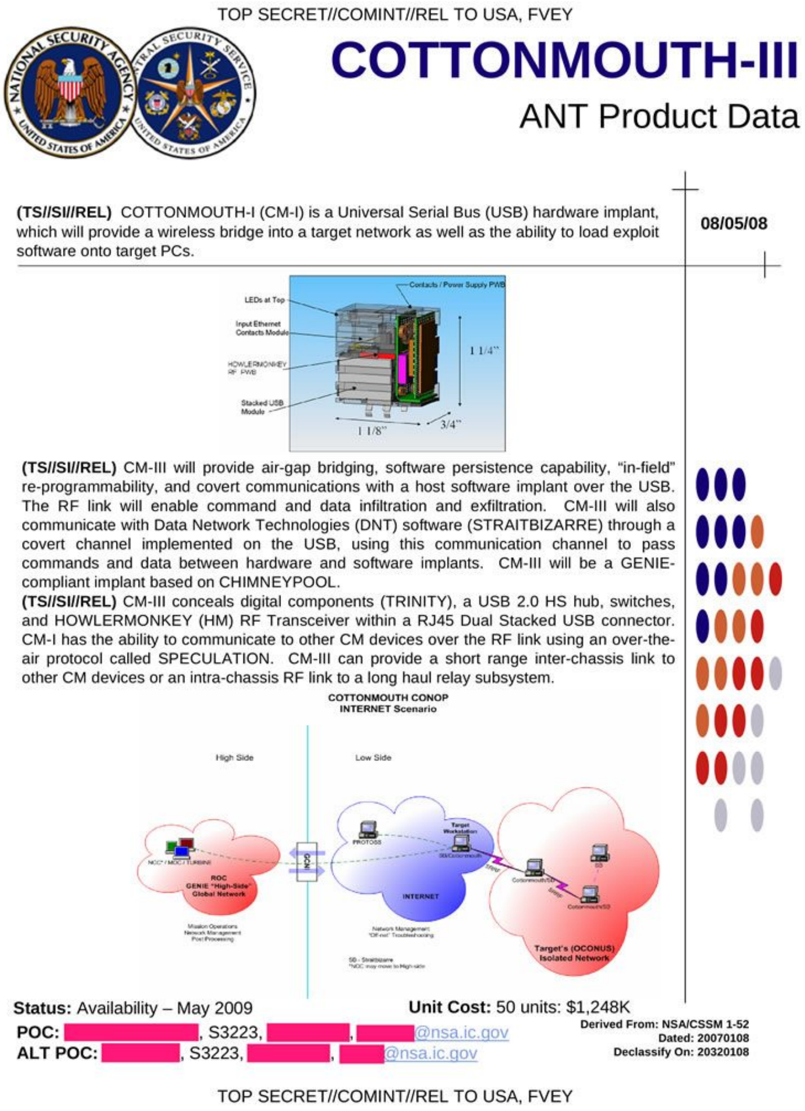
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.47 – COTTONMOUTH-II. An implant hidden inside a USB port featuring covert communications over the USB bus. Controllable through an unspecified covert long haul radio-relay subsystem, likely the SPECULATION protocol used by COTTONMOUTH-I and COTTONMOUTH-III.

Figure 4.74: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL TO USA, FVEY

COTTONMOUTH-III

ANT Product Data

08/05/08

(TS//SI//REL) COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant, which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.

Contacts / Power Supply PWB
LEDs at Top
Input Ethernet Contacts Module
HOWLERMONKEY RF PWB
Stacked USB Module
1 1/4"
1 1/8"
3/4"

(TS//SI//REL) CM-III will provide air-gap bridging, software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. The RF link will enable command and data infiltration and exfiltration. CM-III will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-III will be a GENIE-compliant implant based on CHIMNEYPOOL.

(TS//SI//REL) CM-III conceals digital components (TRINITY), a USB 2.0 HS hub, switches, and HOWLERMONKEY (HM) RF Transceiver within a RJ45 Dual Stacked USB connector. CM-I has the ability to communicate to other CM devices over the RF link using an over-the-air protocol called SPECULATION. CM-III can provide a short range inter-chassis link to other CM devices or an intra-chassis RF link to a long haul relay subsystem.

COTTONMOUTH CONOP
INTERNET Scenario

High Side
Low Side
NCC* / MOC / TURBINE
ROC
GENIE "High-Side" Global Network
Mission Operations
Network Management
Post Processing
GCN
PROTOSS
Target Workstation
SB/Cottonmouth
INTERNET
Network Management
"Off-net" Troubleshooting
SB - Straitbizarre
*NOC may move to High-side
Cottonmouth/SB
SB
Cottonmouth/SB
Target's (OCONUS) Isolated Network

Status: Availability – May 2009

Unit Cost: 50 units: $1,248K

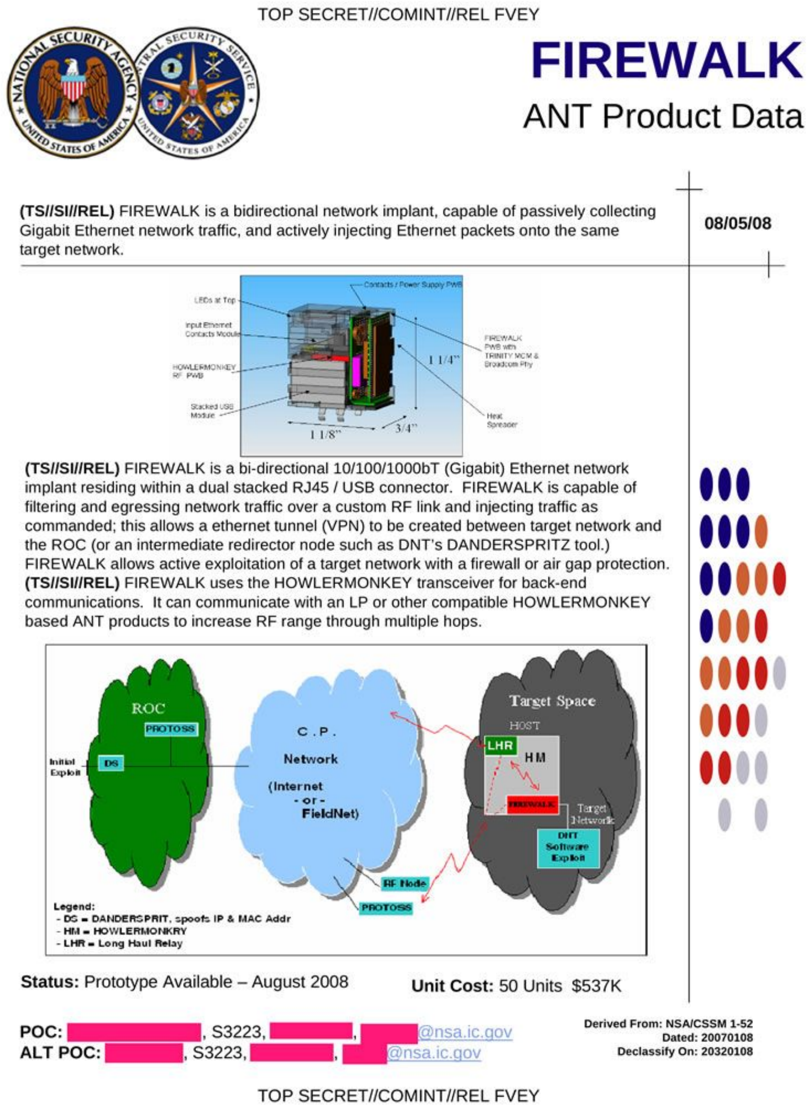
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

4.6.48 – COTTONMOUTH-III. A hardware implant hidden inside an RJ45 dual-stacked USB connector featuring a covert radio transceiver from the HOWLERMONKEY family of implants. It uses an unknown digital radio protocol called SPECULATION for short range (inter-chassis) or long range (intra-chassis) communication.

Figure 4.75: Courtesy of Der Spiegel [AHS13].



TOP SECRET//COMINT//REL FVEY

FIREWALK

ANT Product Data

(TS//SI//REL) FIREWALK is a bidirectional network implant, capable of passively collecting Gigabit Ethernet network traffic, and actively injecting Ethernet packets onto the same target network.

08/05/08

LEDs at Top
Input Ethernet Contacts Module
HOWLERMONKEY RF PWB
Stacked USB Module
Contacts / Power Supply PWB
FIREWALK PWB with TRINITY MCM & Broadcom Phy
Heat Spreader
1 1/4"
1 1/8"
3/4"

(TS//SI//REL) FIREWALK is a bi-directional 10/100/1000bT (Gigabit) Ethernet network implant residing within a dual stacked RJ45 / USB connector. FIREWALK is capable of filtering and egressing network traffic over a custom RF link and injecting traffic as commanded; this allows a ethernet tunnel (VPN) to be created between target network and the ROC (or an intermediate redirector node such as DNT's DANDERSPRITZ tool.) FIREWALK allows active exploitation of a target network with a firewall or air gap protection.

(TS//SI//REL) FIREWALK uses the HOWLERMONKEY transceiver for back-end communications. It can communicate with an LP or other compatible HOWLERMONKEY based ANT products to increase RF range through multiple hops.

ROC
PROTOSS
Initial Exploit
DS
C . P .
Network
(Internet - or - FieldNet)
RF Node
PROTOSS
Target Space
HOST
LHR
H M
FIREWALK
Target Network
DNT Software Exploit
Legend:
- DS = DANDERSPRIT, spoofs IP & MAC Addr
- HM = HOWLERMONKRY
- LHR = Long Haul Relay

Status: Prototype Available – August 2008

Unit Cost: 50 Units $537K

POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL FVEY

4.6.49 – FIREWALK. An Ethernet hardware backdoor hidden inside of an RJ45 dual-stacked USB connector that enables surveillance as well as injection of data into the connected Ethernet network. It uses HOWLERMONKEY as shown in Section 4.6.41 similarly to COTTONMOUTH-I as shown in Section 4.6.46, COTTONMOUTH-II as shown in Section 4.6.47, and COTTONMOUTH-III as shown in Section 4.6.48. It may be used directly through the HOWLERMONKEY radio interface or through relays.

4.6.50 – Related programs. The BANANAGLEE backdoor is used in tandem with the JETPLOW implant as described in Section 4.6.4. BANANAGLEE is also known by the other names such as BG, BARGLEE, BLATSTING, BUZZLIGHTYEAR, and BANALMONKEY. Relatedly, on Juniper Netscreen firewalls, a program named BANANALIAR may also be required for remote exploitation and/or interdiction. See also SCREAMINGPLOW in Section 4.6.50 as an alternative to JETPLOW that makes BANANAGLEE a persistent backdoor. Further information may be found in the documents from the Shadowbrokers publications.

The ZESTYLEAK backdoor is also used in tandem with JETPLOW, but rather than being deployed by the Data Network Technologies (DNT) section of TAO, it is a backdoor deployed by the Cryptographic Exploitation Services (CES) [AGG⁺14d] section of the NSA. This alone makes ZESTYLEAK an especially interesting Juniper backdoor as it strongly implies that this is a way that CES breaks the cryptography deployed by Juniper. Juniper was heavily impacted by NSA sabotage of cryptographic random number generators when another hacking group apparently changed the NSA backdoor parameters for the DUAL_EC_DRBG backdoor. This appears to be part of how the hack of the US Office of Personnel Management (OPM) was executed; there are many open questions. Executable programs related to ZESTYLEAK have been leaked but they have not been analyzed in any public manner known to the author of this thesis. HAMMERMILL is a DNT implant which is related to SECONDDATE. SCREAMINGPLOW is a bash shell script released by Shadowbrokers.

UNITEDRAKE is an NSA implant and backdoor targeting Microsoft Windows. The UNITEDRAKE manual was published [Acc17] by Shadowbrokers, who described UNITEDRAKE (UR) as: "*a fully extensible remote collection system designed for Windows targets.*" The manual details various aspects of UNITEDRAKE such as the System Management Interface (SMI), plug-ins to extend UNITEDRAKE for network transport obfuscation, self-destruct and covert implant features, and many other features.

STRAITBIZZARE is a malware implant for GNU/Linux, Microsoft Windows, and other operating systems which is part of the TAO CHIMNEYPOOL framework [Nex16]. Exfiltration and control is performed using the FRIEZERAMP and CHIMNEYPOOL programs for covert network communication. STRAITBIZZARE is implanted into target devices using computer network exploitation (CNE) techniques. CNE programs used for STRAITBIZZARE include the FOXACID, and QUANTUM families.

OLYMPUS is an implant and backdoor from the NSA.

VALIDATOR is an exploit payload that runs as a user space process and surveys the running OS. It looks for security software such as anti-virus software or other personal security products (PSP), and if nothing is found, additional payloads are downloaded and executed. It also appears to be referenced in some public security research under the name DoubleFantasy [31].

The QUANTUM [Unk14b,Fox15] or QUANTUMTHEORY [S3210] suite of tools including QUANTUMINSERT, QUANTUMHAND, QUANTUMSQUIRREL, QUANTUMCOOKIE, QUANTUMNATION, QUANTUMBOT QUANTUMBISQUIT, QUANTUMDNS, QUANTUMPHANTOM, QUANTUMSKY, QUANTUMSPIM, QUANTUMSMACKDOWN, QUANTUMCOPPER, and QFIRE are part of active CNE programs contained within the TURMOIL/TURBINE distributed attack infrastructure. The QUANTUM suite uses both passive and active injection techniques to perform man-on-the-side attacks. These man-on-the-side attacks are usually possible due to a complete lack of cryptographic security in the protocols which are being monitored, and then later reused for attacks. An example is that QUANTUMINSERT is used to inject network packets which redirect a web browser to a FOXACID server that serves exploit payloads and implants to the targeted client. The FOXACID server is written in Python and attempts to mimic other popular web servers. The QUANTUM suite of tools in 2010 was available for use at Menwith Hill Station, Misawa Airforce base, INCENSOR (GCHQ DS-300), NIPRNET, and additional sites SMOKEYSINK and SARATOGA were planned for the future. QUANTUM related programs are further discussed in Chapter 5.

---

[31] See `https://www.cs.bu.edu/~goldbe/teaching/HW55815/presos/eqngroup.pdf`

4.6.51 – Trickle-down effect. Cell phone surveillance with IMSI-catchers [PS14] has gone from rumor to open products that are commercially available. These are no longer [PS14] the exclusive domain of governments, as is true for many of the items in the ANT catalog. A number of public projects have attempted to clone the capabilities described in the ANT catalog as mentioned in Section 4.6. This underscores a primary criticism of NSA investing in attack tools *at the expense of defensive tools* in the first place: it is a race to the bottom where eventually there will be more than one adversary with such capabilities. The resulting race to the bottom is not accompanied by useful progress to secure the same systems that several adversaries are documented as targeting and attacking.

An example is the USBee [GME16] device which is a tool for covert exfiltration of data. The NSA Playset [Oss14] and other projects [FC14, Wei14, Fit16, VHM16, NYE17, Wak19] investigate recreating the contents of the ANT Catalog.

4.7 — Conclusions

A systematic analysis of surveillance capabilities linked to political geography would assist in understanding and constructing real-world threat models. The tools and other capabilities presented in this chapter are only the tip of the iceberg, and the leaked documents present only a minor glimpse into the behavior and capabilities of spies operating outside of the law. There is hope for cryptography to protect people when it is combined with reasonable operational security precautions and accurate threat modeling.

CHAPTER 5

The GNU name system

5.1 — Introduction

On the net, close to everything starts with a request to the Domain Name System (DNS), a core Internet protocol to allow users to access Internet services by names, such as `www.example.com`, instead of using numeric IP addresses, like 192.0.2.137 or even worse 2001:db8:4145::4242. Developed in the "Internet good old times" where privacy and security was not a concern, the contemporary DNS allows DNS operators to monitor user behavior and usage patterns, and exposes information about the existence and availability of most services on the Internet [Bor15]. Consequently, it now attracts not only all sorts of commercially-motivated surveillance and manipulation,[1] but–as new documents of the NSA spy program MORECOWBELL confirm–also the National Security Agency as well as other intelligence agencies.

DNS currently treats all information in the DNS database as public data. The content of queries and answers is typically not encrypted. This allows passive attackers to monitor the queries of users and see which services they are using and which websites they are visiting. For an active attacker, DNS facilitates locating potentially vulnerable services, which is the first step to their subsequent exploitation with commercially available 0-day attacks.

Given the design weaknesses of DNS, this begs the question if DNS can be secured and saved, or if it has to be replaced — at least for some use cases.

In the last two years, there has been a flurry of activity to address security and privacy in DNS at the Internet Engineering Task Force (IETF), the body that specifies Internet standards, including the DNS. The Internet Architecture Board, peer body of the IETF, called on the engineers to use encryption everywhere, possibly including DNS. [Boa14]

Despite the acknowledgment of the the DNS weaknesses and privacy implications in RFC 7626 [BDLP+15] experts are not expecting that existing industry solutions will change the situation anytime soon:

> "*It seems today that the possibility of massive encryption of DNS traffic is very remote.*" [Bor13b]

The discussions in the IETF now include proposals for "query name minimization", Confidential DNS, DNS over TLS, DNSCurve and more radical proposals for alternative name system designs to improve privacy. Additional work on encrypting traffic to the authoritative name servers high in the chain is ongoing. [Bor16b] All of these designs take different approaches in reducing the role of DNS as the ultimate source of metadata in the digital panopticon known as the Internet. Before we present the different approaches, we illustrate the security goals and the threat model, using NSA as an example of a highly capable attacker and explain what the benefits for the attacker and the risks for the DNS

---

[*] This work was published at the NDSS 2017 DNS Privacy Workshop DPRIV17 as joint paper [GWEA18b] entitled *Towards Secure Name Resolution on the Internet*. It is joint work with Christian Grothoff, Matthias Wachs, and Monika Ermert. This is a corrected version. We have corrected the characterization of NSEC5.

[1] For example, Google's public DNS service permanently logs a dozen items about each request, including the requested domain name, see `https://developers.google.com/speed/public-dns/privacy`. Also, Cisco-owned OpenDNS logs " *any statistical information related to the usage, traffic patterns and behavior of the users*", see `https://www.opendns.com/terms-of-service/`. Finally, there are ISPs manipulating DNS requests and responses, thereby achieving monetary benefits through advertisements, see `https://www.wired.com/2008/04/isps-error-page`. Security problems of these "wildcard" redirections of DNS traffic have been noted, but are ongoing. [fANN15]

user are. Note that, the NSA is only one of the potential attackers, as other state actors as well as criminals can use the same techniques, and some commercial entities mine data as well to feed their profiling databases. We present the NSA attack as an exemplary, because of their technical capabilities and the explanations of their DNS attack strategies published in recently published documents of the agency itself.

5.2 — Background: Domain Name System (DNS)

The Domain Name System (DNS) is an essential part of the Internet as it provides mappings from host names to IP addresses, providing memorable names for users. DNS is hierarchical and stores name-value mappings in so-called *records* in a distributed database. A record consists of a name, type, value and expiration time. Names consist of *labels* delimited by dots. The root of the hierarchy is the empty label, and the right-most label in a name is known as the top-level domain (TLD). Names with a common suffix are said to be in the same *domain*. The *record type* specifies what kind of value is associated with a name, and a name can have many records with various types. The most well-known record type is the “A” record, which maps names to IPv4 addresses.

The DNS database is partitioned into *zones*. A *zone* is a portion of the namespace where the administrative responsibility belongs to one particular authority. A zone has unrestricted autonomy to manage the records in one or more domains. Very importantly, an authority can delegate responsibility for particular *subdomains* to other authorities. This is achieved with an “NS” record, whose value is the name of a DNS server of the authority for the subdomain. The *root zone* is the zone corresponding to the empty label.

The root zone is managed by the Internet Assigned Numbers Authority (IANA), which is currently operated by the Internet Corporation for Assigned Names and Numbers (ICANN), which was depoliticized in 2016 and is since no longer under the control of the National Telecommunications and Information Administration (NTIA) but instead subject to a complex global multistakeholder oversight process where ordinary users will have a hard time being involved. [Erm16]

The root zone contains “NS” records which specify names for the authoritative DNS servers for all TLDs, such as “.de” or “.berlin”.

Names in DNS are resolved using *resolvers*. Many modern operating systems do not provide a full implementation of a DNS resolver but only so called *stub resolvers*. These stub resolvers do not resolve names directly but forward the request to another resolver. In general, we will refer to resolvers that merely forward requests (and possibly cache replies) as *forward resolvers*. After forwarding, the query eventually reaches a *recursive name server*, which is typically provided by the Internet Service Provider (ISP), as shown in Figure 5.1. These recursive name servers resolve the name by first querying the root servers for the required name and by way of recursion go down the DNS tree to fetch the information from the authoritative DNS server. The queried root servers provide the querying resolver with an “NS” record to the server authoritative for the TLD zone, the authoritative server for the zone provides the record for the authoritative server for the domain, subdomain and so on. This *iterative* process is repeated, and terminates for sure when the resolver queries the *authoritative name server* which is responsible for a particular domain.

DNS strongly benefits from caching of DNS information: many *caching resolvers* store information previously requested to improve lookup performance. They use cached record data to skip some or all of the iterations, and thus can return information more quickly to the client.

With the use of forwarding resolvers, the IP address of the client is hidden from authoritative name servers. This gives the user a certain degree of privacy as it prevents operators of authoritative name servers to monitor the source of DNS requests. Naturally, the operators of the forwarding resolvers can still trivially monitor and censor users' requests. Passive dragnet-monitoring with systems such as TURMOIL and XKEYSCORE are also able to see any part of the transaction that is available in the ingestion filter.

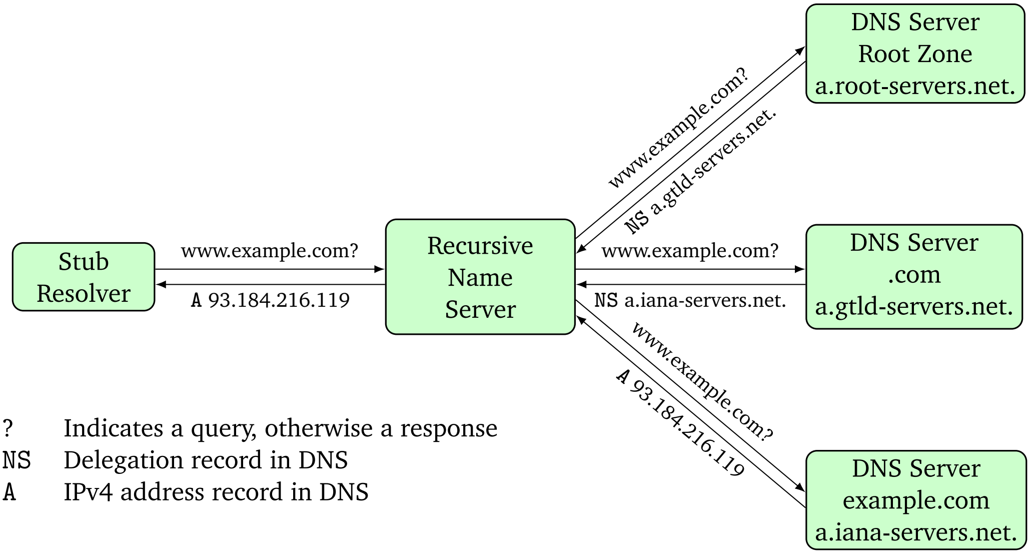

DNS Server
Root Zone
a.root-servers.net.
www.example.com?
NS a.gtld-servers.net.
Stub Resolver
www.example.com?
A 93.184.216.119
Recursive Name Server
www.example.com?
NS a.iana-servers.net.
DNS Server
.com
a.gtld-servers.net.
www.example.com?
A 93.184.216.119
DNS Server
example.com
a.iana-servers.net.
? Indicates a query, otherwise a response
NS Delegation record in DNS
A IPv4 address record in DNS

Figure 5.1: Resolving the name `www.example.com` with DNS. Many operating systems only provide minimal *stub resolvers* forwarding requests to full resolvers. To resolve a name, these resolvers start with querying the name servers of the root zone. If a server cannot provide the required information, it refers the resolver to the next server to query until the server *authoritative* for the respective zone is found.

5.3 — Security goals

When considering improving the security of DNS, there have been striking disagreements among designers as to what the security goals of the DNS system should be. What most designers do agree with is that for the public DNS service, anyone should be able to resolve domain names in it without prior authorization. This does not preclude the possibility of DNS servers returning sensitive records only for certain users, an approach commonly known as *split view*. However, generally speaking, the consensus is that DNS should answer queries without requiring origin authentication.

5.3.1 – Query origin anonymity. However, even if users of DNS do not have to authenticate, that does not mean that they are anonymous. In the original protocol, the IP addresses of the stub resolvers are hidden behind the recursive name servers, providing a thin veil of privacy. However, this may come at the expense of the origin having to trust the recursive name resolver. Furthermore, with the introduction of the client subnet extension [CvdGLK16], recursive name servers may be configured to expose most of a client's IP address to other DNS servers.

5.3.2 – Data origin authentication and integrity protection. Except for regional censors that today block domains by modifying DNS responses, most designers want to see the authenticity and integrity of DNS responses protected. Weak designs simply use secure communication channels between authenticated resolvers. This achieves integrity protection against adversaries in the network but does not help with data authenticity. Another possibility is to cryptographically sign responses with private keys held online; however, as a strong adversary may compromise authoritative name servers, the best protections are achieved by using offline keys for signing zone data to achieve "end-to-end" security including origin authenticity and integrity protection.

5.3.3 – Zone confidentiality. Before the DNS, all name resolution data was public. With DNS, the notion that zone data could be semi-private and only be exposed upon matching request became a possibility. Exposing full zone information provides useful information to attackers, as they can enumerate network services offered by the target, which with virtual hosting or IPv6 might otherwise not be feasible. Thus, it is desirable to minimize the adversary's ability to enumerate the names in a zone.

5.3.4 – Query and response privacy. Finally, the DNS query itself or the DNS response may include sensitive information. The design principle of data minimization dictates that participants should only learn as much as necessary, thus some proposals try to make DNS less chatty. In the most extreme case, a domain name may contain a password, and responses might contain key material, which both ought to be kept confidential from the recursive and (online) authoritative name servers.

5.3.5 – Censorship resistance. A special goal of some name systems is resistance against censorship. The goal is to make it impossible even for governments that have jurisdiction over any possible DNS operator to block name resolution using legal attacks. This is typically achieved by designs that are self-organizing and thus do not require the interaction with professional registrars.

5.4 — Exemplary Attacker: The NSA's MORECOWBELL and QUANTUMDNS programs

These security goals are critical, as the respective threats against the DNS and its users are not theoretical. As a set of top secret documents published by Le Monde [EGA+15] revealed, the American spy agency NSA monitors DNS as a source of information about the Internet (Figure 5.3). NSA's MORECOWBELL program uses a dedicated covert monitoring infrastructure to actively query DNS servers and perform HTTP requests to obtain meta information about services and to check their availability (Figure 5.2).

Despite the open nature of DNS, the NSA does so covertly (Figure 5.4) to ensure the thousands of DNS lookups every hour are not attributed to the US government (USG). In fact, the servers the NSA rented for the purpose of monitoring DNS and checking Web servers using HTTP are located in Malaysia, Germany and Denmark (Figure 5.5), allowing the NSA to perform the monitoring covertly and to get a more global view on DNS name resolution and service availability. While the NSA slides only list these three countries, the PACKAGEDGOODS non-attributable monitoring infrastructure that MORECOWBELL builds on is known to span machines in at least 13 other countries, as described previously by Der Spiegel in a set of slides describing the NSA's TREASUREMAP program. [Unk14a]

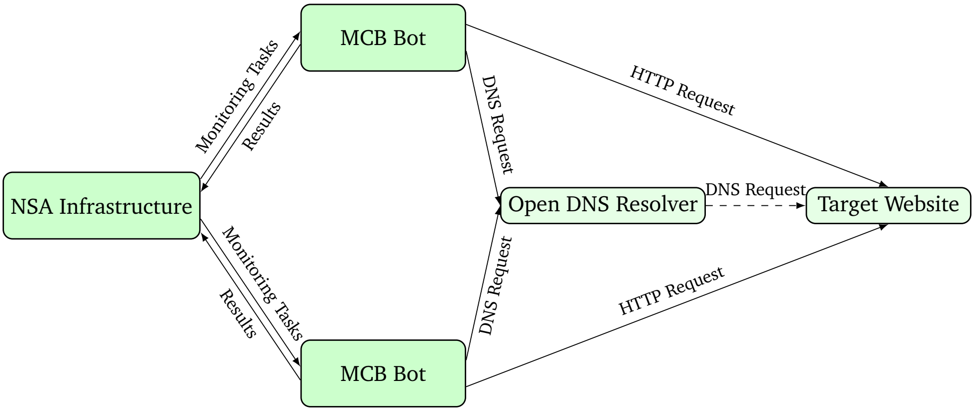
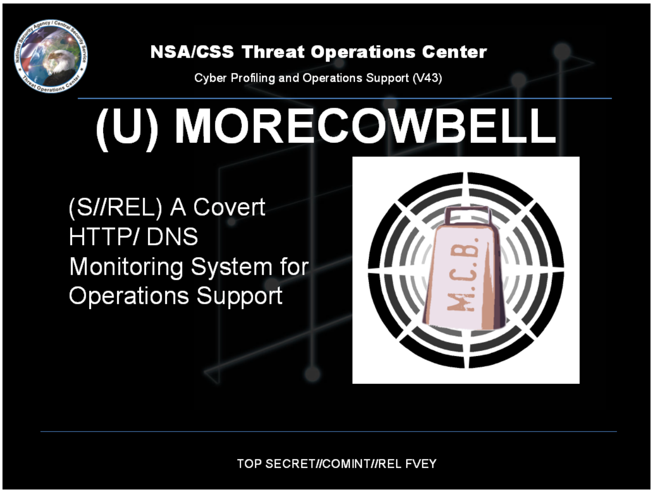


Figure 5.2: From [EGA+15]: NSA's MORECOWBELL infrastructure: a list of targets to monitor is deployed to geographically distributed bots performing DNS and HTTP requests against target websites to collect information about the availability of services. The resulting data are returned to the NSA in regular intervals.

What is interesting is that at the time, the NSA did not care much about the specific content of the Web servers or the DNS entries — as usual the NSA is after the metadata: The NSA wants to know if the DNS information has changed, and check on the availability of the service. The slides show that this simple check has some rather benign uses, for example it is used to monitor some of the US government's own websites.

A key justification for the need to make the active probing of DNS unattributable to the US government is most likely its use for "Battle Damage Indication" (Figure 5.6). Specifically, after "Computer Network Attacks (CNA)" are used against critical network infrastructure, the US may use such probes to confirm that its attacks have found their targets when the lights go out on the Internet systems, say of some foreign government. By monitoring for changes in the DNS, the attack could be repeated if the victim tries to shift its services to another system or network. By keeping the monitoring infrastructure covert and geographically distributed, the NSA gets a global view on the impact of an attack. This makes it harder for victims to identify the monitoring servers, which otherwise might enable victims to evade the attack by treating requests from monitors differently.

The various documents of the NSA relating to DNS show that existing covert attacks on DNS enable mass surveillance and active attacks. [Wea14] With the revelation about the NSA's QUANTUMTHEORY family of projects (Figure 5.7) with subprojects like QUANTUMDNS (Figure 5.8), we know that powerful attackers like nation states can not only eavesdrop DNS traffic but also inject DNS responses to modify the result of name resolution or make it even completely fail. [Red14] With DNS not providing confidentiality to protect a user's privacy, it is easy to create a profile of the users and their surfing behavior on the Web. [KM10] This information could then also be used to perform QUANTUMTHEORY attacks against the target. NSA programs like QUANTUMBOT have the purpose to monitor IRC botnets and detect computers operating as bots for a botnet and hijack the



NSA/CSS Threat Operations Center

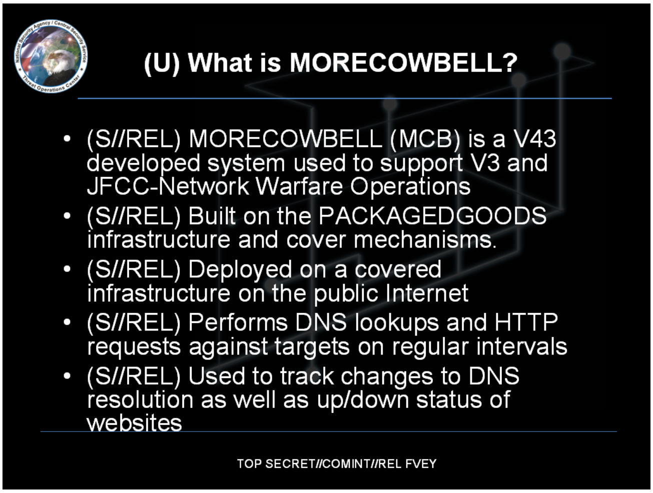
Cyber Profiling and Operations Support (V43)

(U) MORECOWBELL

(S//REL) A Covert HTTP/ DNS Monitoring System for Operations Support

TOP SECRET//COMINT//REL FVEY

Figure 5.3: From [EGA+15]: MORECOWBELL: A Covert HTTP/DNS Monitoring System



(U) What is MORECOWBELL?

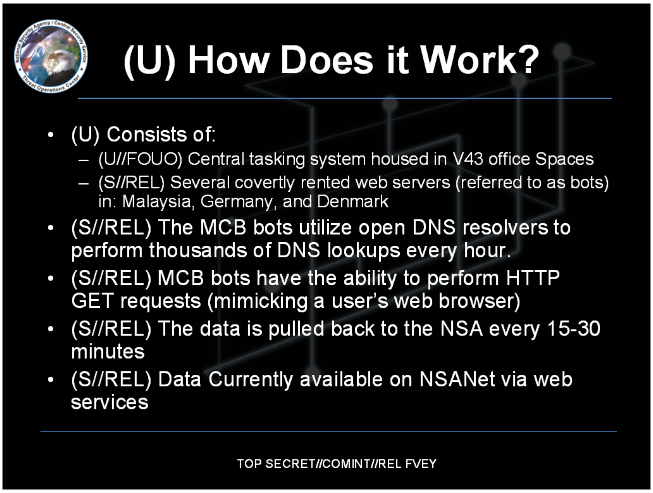
- (S//REL) MORECOWBELL (MCB) is a V43 developed system used to support V3 and JFCC-Network Warfare Operations
- (S//REL) Built on the PACKAGEDGOODS infrastructure and cover mechanisms.
- (S//REL) Deployed on a covered infrastructure on the public Internet
- (S//REL) Performs DNS lookups and HTTP requests against targets on regular intervals
- (S//REL) Used to track changes to DNS resolution as well as up/down status of websites

TOP SECRET//COMINT//REL FVEY

Figure 5.4: From [EGA+15]: What is MORECOWBELL.

command and control channel to manage the bots. These programs are evaluated by the NSA to be *highly successful* according to their documents. [Unk14c]

Thus, the Internet community needs to work towards resolving the privacy and security issues with name resolution and the current Domain Name System (DNS). In the next step, we will review a range of current proposals that have been made to improve the security of this critical Internet service.

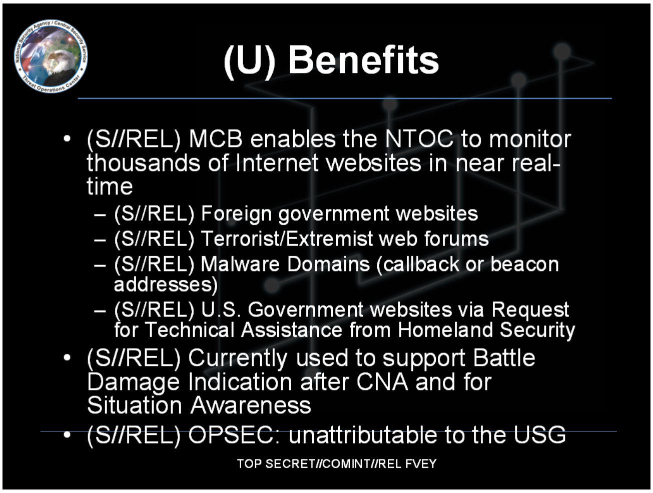


(U) How Does it Work?

- (U) Consists of:
  - (U//FOUO) Central tasking system housed in V43 office Spaces
  - (S//REL) Several covertly rented web servers (referred to as bots) in: Malaysia, Germany, and Denmark
- (S//REL) The MCB bots utilize open DNS resolvers to perform thousands of DNS lookups every hour.
- (S//REL) MCB bots have the ability to perform HTTP GET requests (mimicking a user's web browser)
- (S//REL) The data is pulled back to the NSA every 15-30 minutes
- (S//REL) Data Currently available on NSANet via web services

TOP SECRET//COMINT//REL FVEY

Figure 5.5: From [EGA+15]: How does MORECOWBELL work?



(U) Benefits

- (S//REL) MCB enables the NTOC to monitor thousands of Internet websites in near real-time
  - (S//REL) Foreign government websites
  - (S//REL) Terrorist/Extremist web forums
  - (S//REL) Malware Domains (callback or beacon addresses)
  - (S//REL) U.S. Government websites via Request for Technical Assistance from Homeland Security
- (S//REL) Currently used to support Battle Damage Indication after CNA and for Situation Awareness
- (S//REL) OPSEC: unattributable to the USG

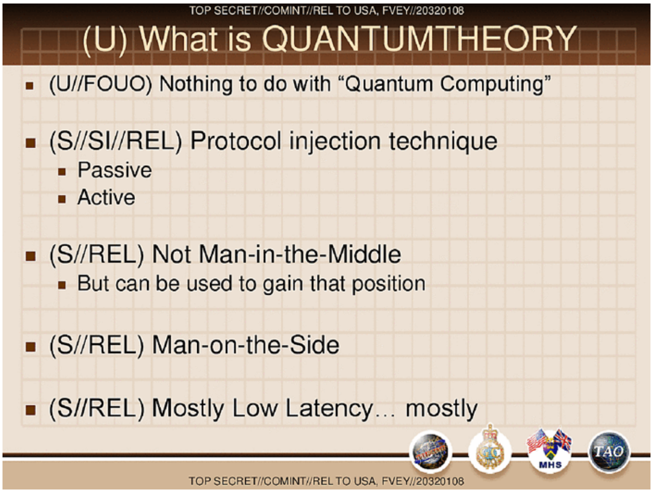
TOP SECRET//COMINT//REL FVEY

Figure 5.6: From [EGA+15]: "Benefits" of MORECOWBELL.

5.5 — Adversary Model

To evaluate existing approaches aiming to improve name resolution security and privacy, we employ two different adversaries:

On the one hand, we examine adversaries within the name system. This can be DNS infrastructure providers operating DNS relevant systems including DNS recursive or forward resolvers. Such adversaries can be honest-but-curious interested in users' usage patterns by monitoring name resolution. To counteract such an adversary query origin anonymity and query response privacy are relevant security goals. Besides being curious,



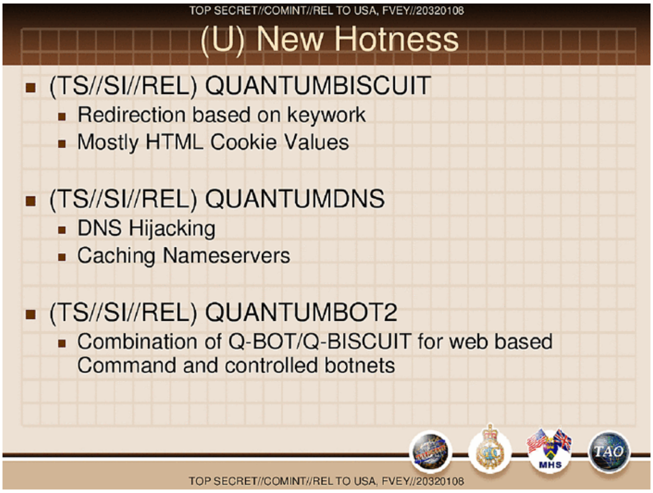
TOP SECRET//COMINT//REL TO USA, FVEY//20320108

(U) What is QUANTUMTHEORY

- (U//FOUO) Nothing to do with “Quantum Computing”
- (S//SI//REL) Protocol injection technique
  - Passive
  - Active
- (S//REL) Not Man-in-the-Middle
  - But can be used to gain that position
- (S//REL) Man-on-the-Side
- (S//REL) Mostly Low Latency… mostly

TOP SECRET//COMINT//REL TO USA, FVEY//20320108

Figure 5.7: NSA’s QUANTUMTHEORY: The man-on-the-side attack.



TOP SECRET//COMINT//REL TO USA, FVEY//20320108

(U) New Hotness

- (TS//SI//REL) QUANTUMBISCUIT
  - Redirection based on keywork
  - Mostly HTML Cookie Values
- (TS//SI//REL) QUANTUMDNS
  - DNS Hijacking
  - Caching Nameservers
- (TS//SI//REL) QUANTUMBOT2
  - Combination of Q-BOT/Q-BISCUIT for web based Command and controlled botnets

TOP SECRET//COMINT//REL TO USA, FVEY//20320108

Figure 5.8: NSA’s QUANTUMDNS: Attacks on DNS are not theoretical. Other slides from the NSA say that QUANTUMDNS is operational and has been successfully used.

such an adversary may be interested in modifying results or make name resolution fail, requiring integrity protection, data origin authentication, and censorship resistance as security goals to antagonize such an attacker.

On the other hand, we employ very powerful adversaries as introduced with the NSA and its MORECOWBELL and QUANTUMDNS programs. Such adversaries may be interested in monitoring users’ behavior monitoring DNS resolution by being able to eavesdrop network traffic, requiring query origin anonymity and query response privacy as a countermeasure. Besides monitoring, such adversaries may want to tamper with name resolution by modifying name resolution (requiring integrity protection and data origin authentication as security goals) or make name resolution fail using technical or legal means (requiring censorship resistance for name systems). Such adversaries may exploit name systems by obtaining zone information to learn about network services that they may subsequently target and exploit. Here, zone confidentiality and response confidentiality are important to avoid leaking knowledge about potential targets.

5.6 — Domain Name System Security Extensions (DNSSEC)

The Domain Name System Security Extensions (DNSSEC) [AAL+05] add integrity protection and data origin authentication for DNS records. DNSSEC does not attempt to improve privacy. It adds record types for public keys ("DNSKEY"), signer delegation ("DS"), for signatures on resource records ("RRSIG") and secure denial of existence ("NSEC"). Figure 5.9 illustrates the interactions among resolvers using DNSSEC. DNSSEC creates a hierarchical public-key infrastructure in which all DNSSEC operators must participate. It establishes a trust chain from a zone's authoritative server to the trust anchor, which is associated with the root zone. This association is achieved by distributing the root zone's public key out-of-band with, for example, operating systems. The trust chains established by DNSSEC mirror the zone delegations of DNS. With TLD operators typically subjected to the same jurisdiction as the domain operators in their zone, with respect to censorship resistance these trust chains are at risk of attacks using both legal and technical means.

Current DNSSEC deployment suffers from the use of the RSA cryptosystem, which thus must be supported by every DNSSEC-enabled resolver. The use of RSA leads to unnecessarily large keys and signatures, and the effect is amplified because response includes the signatures for all of the signature schemes supported by the authoritative server. This can result in message sizes that exceed traditional size restrictions on DNS packets, leading to additional vulnerabilities [Ber10, HS13]. While the IETF has started to add additional ciphers based on elliptic curves [HW12], deploying multiple ciphers further increases packet size and computational cost (if both ciphers are used to secure the same delegation), or reduces security to the weaker of the two ciphers if a mixture of ciphers is used on the resolution path.

DNSSEC also effectively lifts the few traditional limitations on bulk acquisition of zone data, practically reducing zone confidentiality. Before DNSSEC, DNS zone administrators could disallow zone transfers, making it difficult for an adversary to systematically enumerate all of the DNS records in a zone. However, as DNS allows for negative replies (NXDOMAIN), DNSSEC needed a way to create a signed statement that records did not exist. As DNSSEC was designed to keep the signing key offline, "NSEC" records were introduced to certify that an entire range of names was not in use. By looking at the boundaries of those ranges, an adversary can quickly enumerate all names in a zone that are in use. An attempt to fix this via the introduction of "NSEC3" records has been described as broken by security researchers[2]. Nevertheless, NSEC3 is now widely used.[3] As a result, DNSSEC makes it even easier for an adversary to discover vulnerable services and systems. [BM10] But above all, zone confidentiality remains a desideratum.

In the following section we describe the different approaches to add confidentiality to the DNS.

---

[2] `https://dnscurve.org/espionage2.html`

[3] `http://secspider.verisignlabs.com/stats.html`

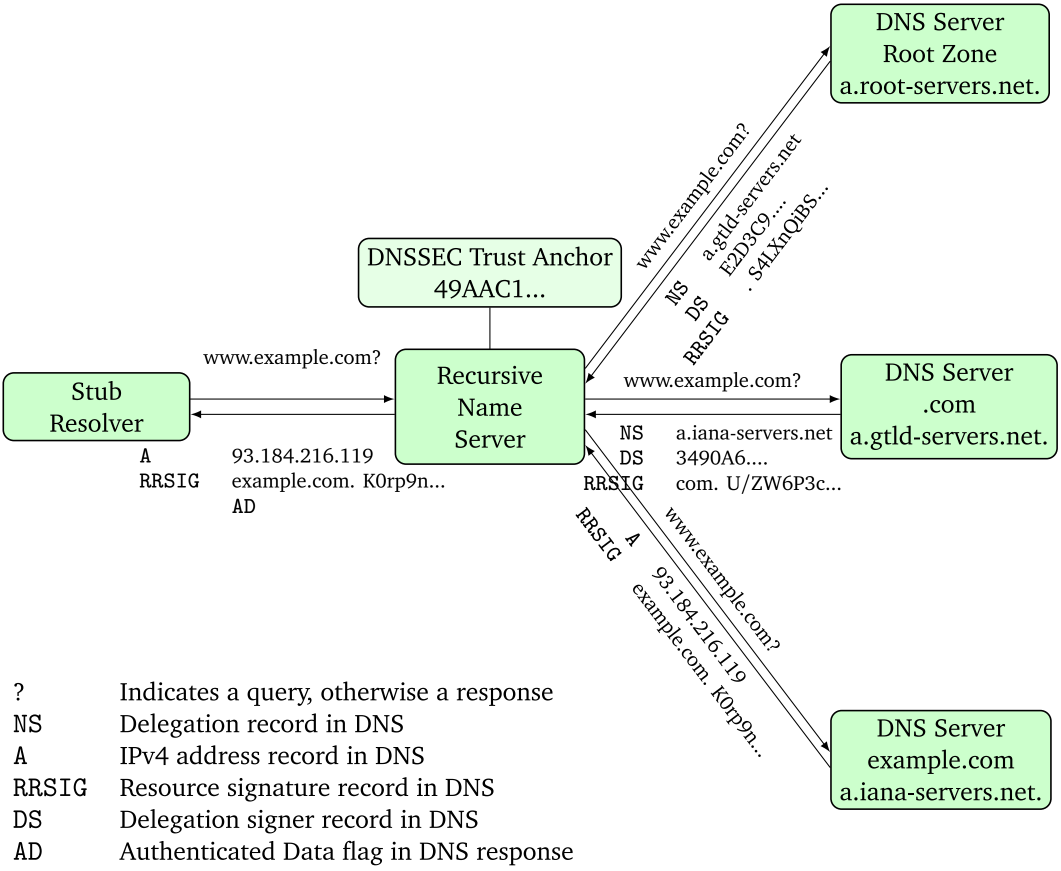

DNS Server
Root Zone
a.root-servers.net.
www.example.com?
a.gtld-servers.net
E2D3C9....
. S4LXnQiBS...
NS
DS
RRSIG
DNSSEC Trust Anchor
49AAC1...
www.example.com?
Stub
Resolver
Recursive
Name
Server
www.example.com?
DNS Server
.com
a.gtld-servers.net.
A 93.184.216.119
RRSIG example.com. K0rp9n...
AD
NS a.iana-servers.net
DS 3490A6....
RRSIG com. U/ZW6P3c...
RRSIG
A
93.184.216.119
example.com. K0rp9n...
www.example.com?
DNS Server
example.com
a.iana-servers.net.
? Indicates a query, otherwise a response
NS Delegation record in DNS
A IPv4 address record in DNS
RRSIG Resource signature record in DNS
DS Delegation signer record in DNS
AD Authenticated Data flag in DNS response

Figure 5.9: Resolving the name `www.example.com` with DNS and DNSSEC: information returned by name servers is cryptographically signed to ensure authenticity and integrity. This information is stored in “RRSIG” records and information about the parent zone stored in “DS” records. A resolver can verify a signature by following this trust chain and using the *trust anchor* shipped out-of-band. Stub resolvers cannot verify this chain and the resolver therefore indicates to the stub resolver that it checked authenticity by setting the `AD` bit in the response given to the client.

5.7 — Query name minimization

The recent discussions in the IETF to improve privacy in DNS (discussed in the DNSOP and DPRIVE working groups) include a standard for so-called *query name minimization* or *QNAME minimization* [Bor16a], which is easy to implement as it does not actually require changes to the DNS protocol. Query name minimization would slightly improve query privacy by having recursive name servers not send the full query to the DNS servers contacted in each resolution step. Instead, each DNS server only receives as much of the DNS name as is necessary for making progress in the resolution process (Figure 5.10). Consequently, the full name being queried is typically only exposed to the final authoritative DNS server.

Query name minimization can simply be implemented by changing how recursive name servers construct their iterative queries. Query name minimization may negatively impact performance, as at least in theory the full query may enable the DNS servers to respond faster with the ultimate answer, if cached information is available or they are the authoritative server for the queried fully qualified domain name. Even with query name minimization, the recursive name servers (at an ISP for example) still learn the full query and reply of a user.

Query name minimization has the advantage that its deployment only requires changes to the recursive name server, and the disadvantage that the change is entirely outside of user control. Query name minimization can be combined with the various approaches to encrypt DNS traffic presented in the next sections. Without query name minimization, simply encrypting DNS traffic— for example using TLS as described in the following section— continues to expose the full query to many DNS servers, in particular root servers and authoritative servers for the respective TLD. With query name minimization, it is possible that only the recursive name server and the authority for the full domain name learn the full name.

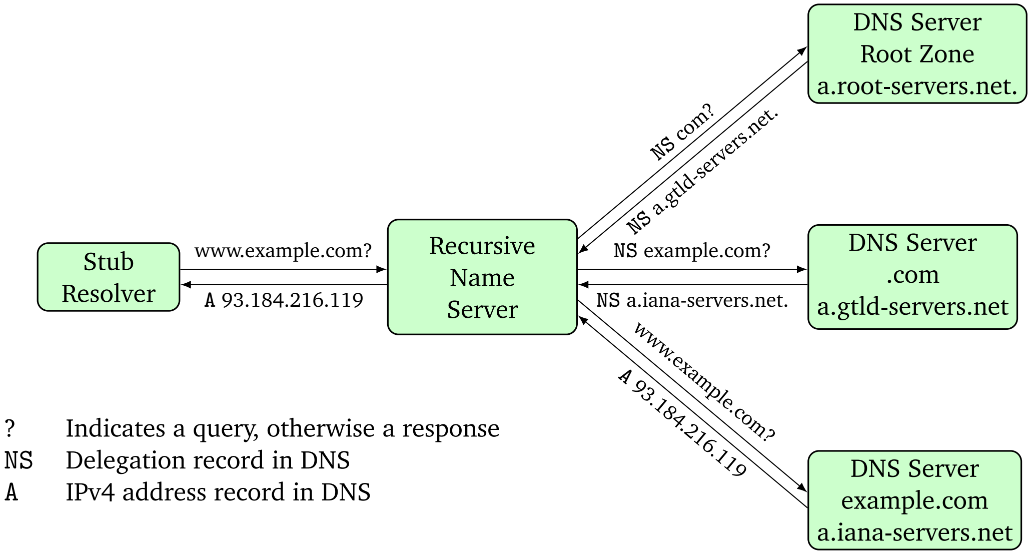

DNS Server
Root Zone
a.root-servers.net.
NS com?
NS a.gtld-servers.net.
Stub Resolver
www.example.com?
A 93.184.216.119
Recursive Name Server
NS example.com?
NS a.iana-servers.net.
DNS Server
.com
a.gtld-servers.net
www.example.com?
A 93.184.216.119
DNS Server
example.com
a.iana-servers.net
? Indicates a query, otherwise a response
NS Delegation record in DNS
A IPv4 address record in DNS

Figure 5.10: With query name minimization, resolving the name `www.example.com` no longer exposes the full name and query type to the root zone and the `.com` authority. Naturally, this scheme still leaks quite a bit of sensitive information to the TLD's DNS server, but less (no `www` in our example) than otherwise. Furthermore, the effect is even weaker in practice, as root zone is already often not contacted as information about TLD name servers is typically cached at forwarding resolvers.

5.8 — DNS-over-TLS

Discussions to use Transport Layer Security (TLS) for encrypting DNS traffic were previously often rejected because of the performance loss associated with such a change. In the discussions about DNS over TLS (standardized as RFC 7858 [HZH^{+}16]) it was pointed out that using TLS would not only be beneficial in supporting query and response privacy and hop-by-hop integrity protection, but by switching to TCP — and therefore from connectionless UDP to connection-oriented TCP — might also help mitigate against amplification attacks on (or by) DNS servers. [MWH+14]

By re-using a TCP connection for multiple DNS requests with moderate timeouts, pipelining requests and allowing out of order processing, the DNS-over-TLS proposal promises reasonable performance despite the overheads from TCP and TLS.

However, even if TLS were to be deployed for DNS, this would not improve query origin anonymity since it still leaks metadata, allowing third parties to easily determine which DNS data a user accesses: In the IETF proposal, TLS is used in combination with the traditional DNS lookup paths, which may involve the use of forward resolvers that assist endpoints performing DNS queries. The involvement of such forward resolvers can obscure the user's IP address from the other DNS servers; naturally, for this to be sufficient the forward resolvers themselves would have to be trusted to not spy on the user. Furthermore, TLS itself does not have the best security track record, with dozens of issues in recent years ranging from high-profile certificate authority compromises to broken implementations and insecure cipher modes. [Hol13] Ways for users to configure just how broken (or optimistic [Duk14]) TLS is allowed to be for their DNS-over-TLS requests continues to be the subject of a current IETF draft [DGR16]. Key problems in this context include the need for incremental deployment, and that TLS authentication itself can require DNS names [SAH11] or even use DNS records [Bar11], resulting in a bootstrap problem that needs to be mitigated.

TLS is not the only possible method for encrypting DNS queries and replies as they traverse the network to increase query and response privacy as well as integrity. DNSCurve and Confidential DNS are alternative proposals to protect the content of DNS queries and replies from network-level monitoring and modification.

DNS-over-TLS is available in the Unbound DNS server[4] and the Knot resolver[5] It is also possible to implement DNS-over-TLS using a TLS proxy in front of a nameserver. Several pilot public servers implementing DNS-over-TLS are currently set up[6] one for example at the Domain Name System Operations Analysis and Research Center.[7]

5.9 — DNSCurve

The first practical system that improves confidentiality with respect to DNS queries and responses was DNSCurve [Ber08b]. In DNSCurve, session keys are exchanged using Curve25519 [Ber06] and then used to provide authentication and encryption between caches and servers. DNSCurve improves the existing Domain Name System with query and response confidentiality and hop-by-hop integrity without the need to create expensive signatures or (D)TLS sessions. Specifically, DNSCurve achieves the same round trip time (RTT) as DNS by embedding the public key of the server in the "NS" record, conflating the DNS namespace with key information.

---

[4] `https://unbound.net/`.

[5] `https://www.knot-resolver.cz/`.

[6] `https://portal.sinodun.com/wiki/display/TDNS/DNS-over-TLS+test+servers` contains a list.

[7] `https://www.dns-oarc.net/oarc/services/dnsprivacy`.

DNSCurve creates an authenticated and encrypted association between a *DNSCurve server* and a *DNSCurve cache*, the latter being a caching recursive DNS resolver running at the endpoint instead of a DNS stub resolver (Figure 5.11). As DNSCurve does not use signatures, the DNSCurve cache cannot prove the authenticity of the cached records to other users, limiting the utility of each cache to the respective endpoint.

While in DNSCurve the user no longer has to trust a forward resolver, the endpoint's IP address is now directly exposed to the authoritative DNS servers: it is no longer obscured by recursive name servers operated by network service providers. Thus, DNSCurve can increase privacy against an adversary monitoring DNS traffic on intermediary systems or with other cable tapping, but reduces query origin anonymity with respect to authoritative DNS servers, as they learn both the full query and the identity (IP address) of the user. Another commonly voiced concern about DNSCurve is the need to keep private keys online. DNSCurve also cannot protect against censorship, as certain governments continue to effectively control the hierarchy of registrars and can thus make domains disappear. With respect to attacks from the NSA, DNSCurve only helps users against passive surveillance on the wire by protecting the confidentiality of at least the DNS payload.

With DNSCurve, DNS servers remain a juicy target for mass surveillance. Furthermore, as with DNS, the well-known and easily located DNS servers remain a target and confirmation vector for attacks on critical infrastructure. With DNSCurve, the need for online public key cryptography by the DNS authorities may open up an additional vulnerability to computational denial of service attacks if a small CPU is used to handle a high-speed link.

DNSCrypt. DNSCrypt is an unstandardized but documented protocol largely based on DNSCurve. It protects the end user's stub resolver queries from network surveillance and tampering hereby improving query and response privacy and integrity. As it is based on DNSCurve, it does not solve any of the major other privacy or security issues present in DNS. The largest known resolver to support DNSCrypt is OpenDNS. There are a number of open DNSCrypt resolvers run by the DNSCrypt community. Today, DNSCrypt remains the most widely deployed DNS encryption protocol designed to prevent surveillance of end users from the network. However, it only helps to solve half of the privacy problem, and it is not widely adopted or standardized.

5.10 — Confidential DNS

Another IETF draft which has been discussed in the IETF DPrive Working Group suggests an alternative method for adding encryption to DNS. It uses the main extension mechanism of DNS, the introduction of additional record types, to encrypt DNS traffic [Wij14], hereby achieving query and response privacy and integrity protection. With Confidential DNS, a new "ENCRYPT" record type is introduced to provide the necessary public key that would allow the recursive name server to encrypt the connection to the DNS server. This "ENCRYPT" record contains the public key of the DNS server to be used to encrypt communication initiated by the resolver. The "hack" used by DNSCurve where the public key was added into the "NS" response of the delegating zone is avoided.

The current draft supports two different operation modes: an *opportunistic* mode which is easier to realize since it does not require major changes to DNS infrastructure and an *authenticated* mode, where a domain's public keys are also stored in the respective

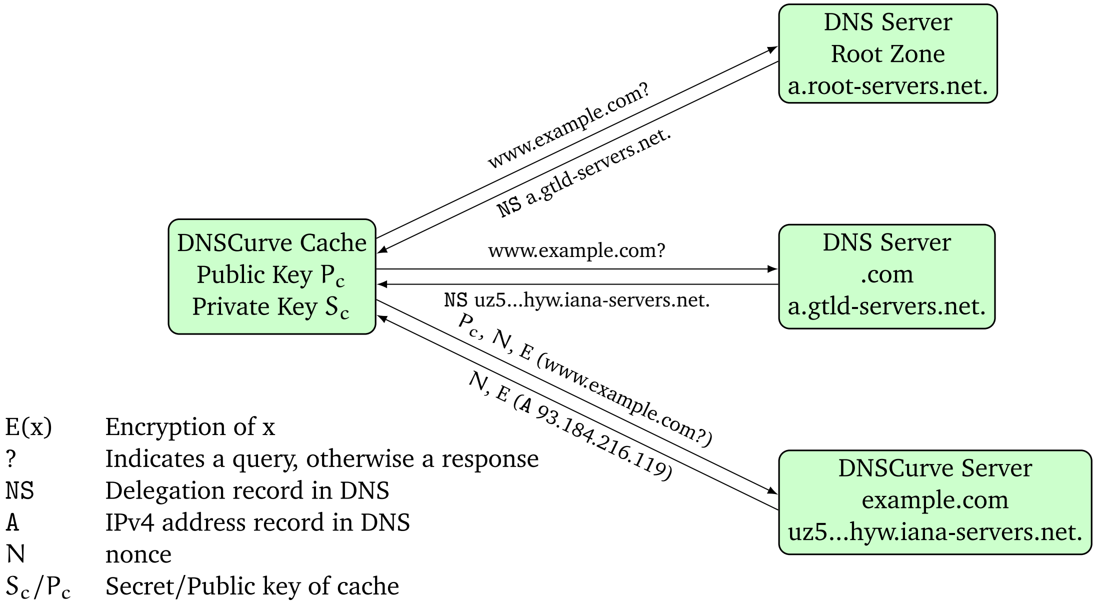

DNS Server
Root Zone
a.root-servers.net.
www.example.com?
NS a.gtld-servers.net.
DNSCurve Cache
Public Key P_c
Private Key S_c
www.example.com?
DNS Server
.com
a.gtld-servers.net.
NS uz5...hyw.iana-servers.net.
P_c, N, E (www.example.com?)
N, E (A 93.184.216.119)
DNSCurve Server
example.com
uz5...hyw.iana-servers.net.
E(x) Encryption of x
? Indicates a query, otherwise a response
NS Delegation record in DNS
A IPv4 address record in DNS
N nonce
S_c/P_c Secret/Public key of cache

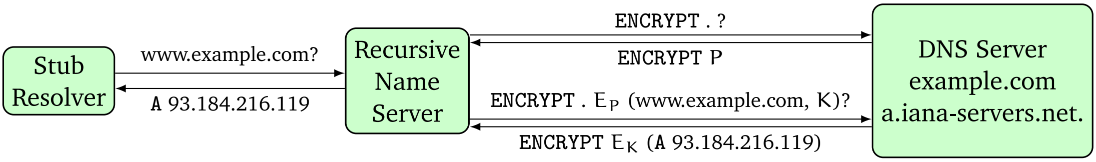
Figure 5.11: Resolving the name `www.example.com` with DNSCurve. With DNSCurve, the resolving cache and the DNSCurve server exchange a shared secret to encrypt their communication. The DNSCurve server's public key is encoded in the name of the name server itself using Base32. When a DNSCurve cache resolves a name and finds the name server to support DNSCurve, the cache creates a shared secret based on the server's public key, the cache's private key, and a one-time nonce. The cache sends its public key, the nonce and the query encrypted with the shared secret. The server will respond with the result of the query encrypted with the shared secret. The first two lookups to the root zone and the ".com" TLD do not use DNSCurve in the illustration as those currently do not support DNSCurve.

parent zone, thus requiring support from the parent zone's DNS infrastructure.

With the opportunistic mode, the public key is no longer associated with the parent zone and instead served separately in the clear and possibly without authentication as a record with the target zone. As a result, Confidential DNS using the "ENCRYPT" record only supports so-called *opportunistic encryption*, which is encryption that is trivially bypassed by a man-in-the-middle attack, as it uses unauthenticated keys for encryption.

The use of a new record type also creates the opportunity for the necessary complexity of a committee-engineered solution: Confidential DNS can use symmetric or asymmetric cryptography, and sports support for 512-bit RSA and AES in CBC mode (which was recently used to finally kill off SSL3 [MDK14]). The draft fails to set a strong minimum baseline and to ensure that this minimum will be updated to reflect new security considerations in due course.

The draft on Confidential DNS provides also a method to achieve "real" authenticated encryption by storing a domain's public key in the respective parent zone. To do so, Confidential DNS extends DNSSEC's Delegation Signer ("DS") resource records to provide the encryption key for the zone. This resembles the "NS" record used by DNSCurve. This approaches makes Confidential DNS susceptible to censorship attacks since it relies on DNS's hierarchical architecture.


Stub Resolver
www.example.com?
A 93.184.216.119
Recursive Name Server
ENCRYPT . ?
ENCRYPT P
ENCRYPT . E_P (www.example.com, K)?
ENCRYPT E_K (A 93.184.216.119)
DNS Server example.com a.iana-servers.net.

?	Indicates a query, otherwise a response
.	Query for the root zone
P	Public key of server
K	Encryption Key
E_P (x)	Encryption of x with P
E_K (x)	Encryption of x with K
A	IPv4 address record in DNS
ENCRYPT	Encrypt record in DNS

Figure 5.12: Resolving the name `www.example.com` with opportunistic Confidential DNS. The resolver retrieves the DNS server's public key querying for the new "ENCRYPT" record. This public key can then be used to encrypt the query to the server. The resolver sends the query encrypted with the server's public key containing the query and the key to encrypt the reply with.

The draft provides for a variety of failure modes, such as "fallback to insecure" allowing clients to relapse to insecure modes with "leaps of faith" even after secure connections used to be available. Confidential DNS allows implementations to "fallback to insecure" in case one side uses cryptographic algorithms that the other does not support. These various scenarios in which Confidential DNS simply falls back to unencrypted channels (without any indication to the user) highlight how much the design focuses on being easy to deploy at the expense of providing predictable security. Given the recent adoption of DNS-over-TLS and critiques that Confidential DNS introduces a DDoS vector, the Confidential DNS specification has not been updated in a while and remains unfinished IETF work.

5.11 — Namecoin

None of the approaches presented so far are designed to withstand legal attacks. Depending on their reach, governments, corporations and their lobbies can legally compel operators of DNS authorities to manipulate entries and certify the changes. Hence the above systems are vulnerable to censorship.

Alternative peer-to-peer name systems provide more radical solutions to secure name resolution. Timeline-based systems in the style of Bitcoin [Nak08a] have been proposed to create a global, secure and memorable name system [Swa11]. Here, the idea is to create a single, globally accessible timeline of name registrations that is append-only. Timeline-based systems rely on a peer-to-peer network to manage updates and store the timeline. In the Namecoin system [pro13], modifications to key-value mappings are attached to transactions which are committed to the timeline by mining. Mining is the use of brute-force methods to find (partial) hash collisions with a state summary (fingerprint) representing the complete global state — including the full history — of the timeline.

Given two timelines with possibly conflicting mappings, the network accepts the timeline with the longest chain as valid, as it represents the largest expense of computational power. This is supposed to make it computationally infeasible for an adversary to produce an alternative valid timeline. This assumes limited computational power and may not actually be binding for certain adversaries.

To perform a lookup for a name with Namecoin, the client has to check the timeline if it contains an entry for the desired name and check the timeline for correctness to ensure that the timeline is valid. To do so, the user has to possess a full copy of the timeline (Figure 5.13), which had a size of about 4.7 GB in November 2016.[8] Alternatively, users may use a trusted name server participating in the Namecoin network.

Namecoin can improve user privacy if the full blockchain is replicated at the user's end system. In this case, resolving a name does not involve the lookup and is thus perfectly private with respect to query origin anonymity and query and response privacy. However, replicating the full blockchain at each user may be impractical for some devices should Namecoin ever grow to be a serious competitor for DNS. Namecoin also does not protect the zone information from monitoring, and in particular zone enumeration is trivial. However, the decentralized nature of Namecoin does ensure that at least battle damage indication against a name server no longer makes sense.

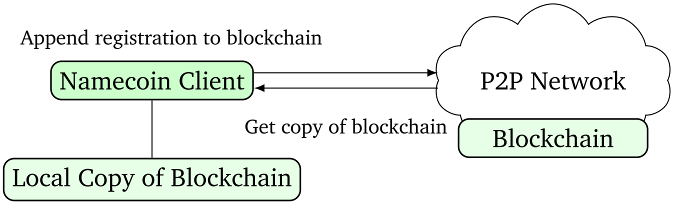


Figure 5.13: The Namecoin name system is decentralized and uses a peer-to-peer network. To achieve a consensus about names registered, Namecoin uses a *blockchain* stored in the peer-to-peer network. To register a name, clients have to pay a miner to perform some computational work to get their name appended to the chain. To resolve a name, clients have to possess a full copy of the blockchain and search for the name to resolve in the blockchain.

5.12 — The GNU name system

The authors of this chapter are working on the GNU Name System (GNS) [WSG14], which is a more radical proposal to address DNS privacy and security issues, and which like Namecoin significantly departs from DNS's name resolution process. The GNS resolution process does not use resolvers querying DNS authorities. Instead, GNS uses a peer-to-peer network and a distributed hash table (DHT) to enable resolvers to lookup key-value mappings. As a result, GNS will inherit the performance and availability characteristics of the underlying DHT. Various implications of such a transition on availability and performance have been analyzed previously in [PMTZ06]. However, in contrast to previous work that proposed to simply replicate information from DNS into a DHT to improve resilience and performance [RS04, CMM02], GNS provides a fully decentralized name system which is conceptually independent from DNS.

GNS is privacy-preserving since queries and responses are encrypted such that even an active and participating adversary can at best perform a confirmation attack, and otherwise only learn the expiration time of a response. Note that the queries and responses themselves are encrypted, not the connections between a resolver and some authority. As all replies are not just encrypted but also cryptographically signed, GNS provides integrity protection since peers in the DHT cannot tamper with the results without immediate detection and data origin authentication.

---

[8] `https://bitinfocharts.com/namecoin/`

Due to the use of a DHT, GNS avoids DNS complications such as glue records and out-of-bailiwick lookups. In GNS, the labels of a name correspond precisely to the lookup sequence, making the complete trust path obvious to the user. Finally, the use of a DHT to distribute records also makes it possible for GNS authorities to operate zones without visible, attributable critical infrastructure that could be used for battle damage indication.

GNS can securely resolve names to any kind of cryptographic token. Thus, it can be used for addressing, identity management and as an alternative for today's battered public key infrastructures.

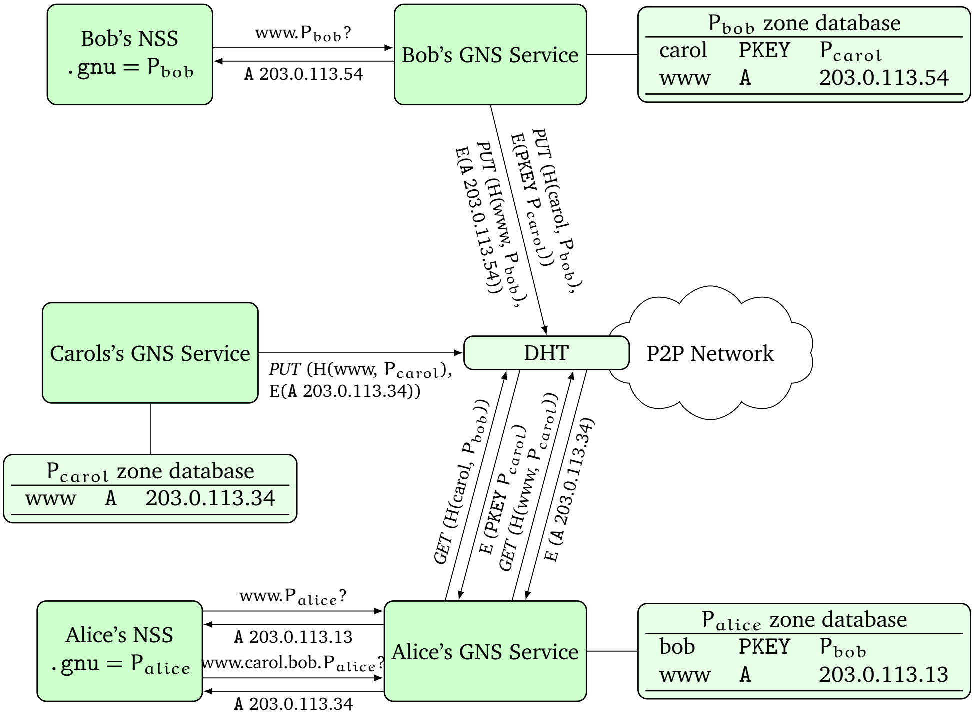

Bob's NSS
.gnu = P_{bob}
www.P_{bob}?
A 203.0.113.54
Bob's GNS Service
P_{bob} zone database
carol PKEY P_{carol}
www A 203.0.113.54
PUT (H(carol, P_{bob}), E(PKEY P_{carol}))
PUT (H(www, P_{bob}), E(A 203.0.113.54))
Carols's GNS Service
PUT (H(www, P_{carol}), E(A 203.0.113.34))
DHT
P2P Network
P_{carol} zone database
www A 203.0.113.34
GET (H(carol, P_{bob}))
E (PKEY P_{carol})
GET (H(www, P_{carol}))
E (A 203.0.113.34)
Alice's NSS
.gnu = P_{alice}
www.P_{alice}?
A 203.0.113.13
www.carol.bob.P_{alice}?
A 203.0.113.34
Alice's GNS Service
P_{alice} zone database
bob PKEY P_{bob}
www A 203.0.113.13

Figure 5.14: The GNU Name System: with GNS, every user maintains their own databases containing record sets under labels organized in zones. A zone is referenced by a public-key pair. Here Alice, Bob and Carol have web servers all reachable under `www.gnu`. For Alice `www.gnu` resolves to a different address than for Bob or Carol, as their respective local name service switches (NSS) associate a user-specific public key with `.gnu`. To allow other users to resolve the names, a user's public zone information is encrypted and published in a DHT under an obfuscated query key. A user can *delegate* to another user's namespace from their local namespace to resolve foreign names. Alice can access Bob's namespace by delegating control over the name `bob` to P_{bob} in her namespace using a GNS-specific "PKEY" record. This way Alice can access Carol's web server using the name `www.carol.bob.gnu`.

5.12.1 – Names, zones and delegations. A GNS zone is a public-private key pair and a set of associated records. The GNS name resolution process basically resolves a chain of public keys. In the absence of a widely recognized and operational *root zone*, but also as an inherent alternative to hierarchical addressing, GNS uses the pseudo-TLD ".gnu" to refer to the user's own zone, which is called the *master zone*. Each user can create any number of zones, but one must be designated as the master zone. Users can freely manage mappings for the labels in their zones. Most importantly, they can delegate control over a subdomain to any other zone (including those operated by other users) using a "PKEY" record, which simply specifies the public key of the target zone. "PKEY" records are used to establish the aforementioned delegation path. Due to the use of a DHT, it is not necessary to specify the address of some system that is responsible for operating the target zone. Record validity in the DHT is established using signatures and controlled using expiration values.

5.12.2 – Cryptography for privacy. To enable other users to look up records of a zone, all records for a given label are stored in a cryptographically signed block in the DHT. To maximize user privacy when using the DHT to look up records, both queries and replies are encrypted and replies are signed using a public key derived from the public key of the zone and the label (Figure 5.14). Any peer can easily validate the signature but not decrypt the reply without prior knowledge of the public key and label of the zone. Consequently, users can use passwords for labels or use public keys that are not publicly known to effectively restrict access to zone information to authorized parties.

Due to the use of a DHT, all GNS queries go to the same fully decentralized and shared global infrastructure instead of operator-specific servers. This provides censorship-resistance and makes it impossible to target a zone-specific server because all machines in the DHT are jointly responsible for all zones — in fact, the key-value pairs do not reveal which zone they belong to. At the same time, encryption and authentication of the records is critical as it helps protect the users from effective censorship or surveillance. However, unlike the other less radical proposals to overhaul DNS, deploying GNS will be a significant challenge: GNS requires more significant changes to software, as well as a community effort to operate a DHT as a new public infrastructure.

5.13 — Assessment

The technical approaches presented differ widely in their security goals. We summarize the key differences in Table 5.1. DNS basically assumes a trustworthy IP network, the other models (except for Confidential DNS) assume that the network cannot be trusted to protect the integrity of the data. Protecting the integrity of the responses has thus been the first order of business for all approaches to secure DNS, starting with DNSSEC.

DNSSEC's limited focus means that it does not consider privacy implications of exposing requests and responses and their origin to the network. Only Namecoin and GNS try to hide the nature of client requests from the operators of the network. Here, GNS is vulnerable to a confirmation attack, so Namecoin's protection is technically stronger in terms of client request privacy. The other approaches expose the contents of the queries and replies to the operators; query name minimization (not shown) can be used to limit which servers get to learn the full query. However, clients have not assurances that query name minimization is actually deployed.

DNSSEC did try (but failed) to protect zone information against zone walks. The situation is not easily remedied by the use of stronger cryptographic primitives, as NSEC5 [GNP+14] provides an impossibility result showing that online cryptography is necessary to support NXDOMAIN responses, and preventing bulk acquisition of zone data. The proposed scheme for NSEC5 uses two different public keys to separate the offline key used to sign zone data from the online key used to generate NXDOMAIN responses. This way, compromising the online key only enables zone enumeration, but does not impact integrity. In contrast, GNS does not use online cryptography or any direct interaction with the zone's authority. GNS can store even confidential data in the name system, effectively protect it from illicit observation by the network or service operators and use offline signing, but cannot support NXDOMAIN. Finally, Namecoin deliberately made the opposite design choice and exposes the full database to all participants.

Using unsolicited DNS replies by open resolvers for traffic amplification is a well-known vector for DDoS attacks. The increased size of DNSSEC responses makes the situation worse, while caching of NSEC replies could also help reduce traffic. Some of the new approaches are not based on UDP, thus making it significantly more difficult to abuse DNS for traffic amplification.

Only the alternative approaches, Namecoin and GNS, are resistant to censorship. Approaches using traditional DNS registrars are inherently vulnerable to legal attacks where influential entities force registrars to block names.

	Protection against						Ease of Migration / Compatibility
	Manipulation by MiTM	Zone walk	Client observation		Traffic Amplification	Censorship / Legal attacks	
			network	operator			
DNS	✗	✓	✗	✗	✗	✗	+++
DNSSEC	✓	failed	✗	✗	+/-	✗	+*
DNSCurve	✓	✓	✓	✗	✓	✗	+*
DNS-over-TLS	✓	n/a	✓	✗	✓	✗	+
Confidential DNS	✗	n/a	✓	✗	✗	✗	++
Namecoin	✓	✗	✓	✓	✓	✓	-
GNS	✓	✓	✓	✓	✓	✓	- -

*EDNS0

Table 5.1: Comparison of the defenses offered by the various designs and their relative deployment complexity.

Naturally, Table 5.1 falls short of considering the complete picture. For example, without padding, encrypted queries and responses may still leak information by exposing the size of the message. Also, traffic amplification is merely one vector for denial-of-service attacks, and there may be other ways to impact the fundamental security goal of availability. Our comparison also excludes practical issues, such as the propagation delay for updates, resolution latency, and general usability.

5.14 — Conclusions

In "*Culture Is Our Business*" [McL70] Marshall McLuhan stated presciently:

> "***World War III is a guerrilla information war with no division between military and civilian participation.***"

It appears that his prediction from 1970 remains relevant when we consider the Internet's architecture as it is woven through our everyday lives.

DNS was never designed with privacy or security as a design goal. In the battle of nation states for global dominance, any Internet infrastructure that serves a specific audience is a target for state attackers. Critical infrastructure needs to be logically decentralized and should ideally be shared globally to reduce the value of harming it. Merely encrypting DNS and Web traffic may not sufficiently reduce the effectiveness of targeted attacks against insecure designs.

Awareness exists in the DNS community that privacy is an issue, and ongoing work investigates the security, compatibility and performance implications of proposed alternatives [Shu14]. Nevertheless, the diverse interests in the community make it virtually impossible to quickly make significant progress by consensus. Modifications to a deployed system like DNS, following the general ossification trend of the Internet, are met with inertia and usually end up with death by committee, as any significant change could not only result in serious malfunctioning, but may also impact somebody's business model or nation state interest.

The currently proposed band aids from the IETF fail to address the scope of the problem: surveillance of users, commercial censorship and the danger that DNS systems and their administrators become legitimate targets for technical, political or military attacks must be addressed better in future designs.

CHAPTER 6

Tiny WireGuard Tweak

6.1 — Introduction

WireGuard [Don17b] is a recently introduced Virtual Private Network (VPN) protocol which is both simple and efficient. It aims to replace other protocols such as IPsec [Dun01] and OpenVPN [Yon] for point-to-point tunnels with a secure protocol design that rejects cryptographic agility. WireGuard uses a fixed set of sound cryptographic primitives and does not negotiate them – in stark contrast to nearly every other major VPN protocol. Unlike many protocols, WireGuard *requires* out-of-band peer configuration information to be exchanged before it may be used. All peers *must* exchange fixed pairwise-unique long-term static public keys as well as Internet host name or address information out-of-band. WireGuard optionally allows peers to fix a pairwise-unique static symmetric value known as a Pre-Shared Key (PSK). A well-known VPN provider, Mullvad, has a worldwide deployment [Mul17] of WireGuard that uses this PSK [Mul] as a method of adding post-quantum transitional security to the protocol. WireGuard does not require, nor use a PSK by default. A protocol is post-quantum transitionally secure when it is secure against a passive adversary with a quantum computer [SWZ16a]. If this transitionally secure protocol is used today, it is not possible for a quantum attacker to decrypt today's network traffic, *tomorrow*.

If a future adversary has access to a quantum computer, historic network traffic protected by WireGuard, and knowledge of *one* WireGuard user's long-term static public key, this threatens the security of the protocol for all related WireGuard users, as explained in Section 6.5. In this section of the thesis we propose a tiny tweak to the WireGuard protocol that makes WireGuard traffic flows secure against such an adversary; if our alteration is incorporated into the WireGuard protocol, a user's historic traffic will not be able to be decrypted by such an adversary if they do not release their long-term static public key to the network, as explained in Section 6.6. We accomplish this with both extremely minimal costs and minimal changes to the original protocol, as detailed in Section 6.6.1.

Note that our analysis applies to the current version of WireGuard [Don18b] as implemented in the Linux kernel [Don19d] as opposed to the older version described in the NDSS paper [Don17b]. A major difference exists in the application of the PSK during the handshake which results in two incompatible protocols. Additionally, we note that there is a migration process where currently in-use keys must be rotated before deploying and using the Tiny WireGuard Tweak.

6.2 — Realistic adversary concerns

It is well-documented and indisputable that a number of nation-state-sponsored adversaries are unilaterally conducting mass surveillance of the Internet as a whole. This has created new notions of realistic threat models [Pre15, Rog15, BSJ+15b] in the face of such pervasive surveillance adversaries. Some of these adversaries have an openly stated interest in *"collecting it all"* [Gre13c] and have directly stated that they use this data as actionable information, for example, for use in internationally contested drone strikes against unknown persons. The former director of the CIA, General Michael Hayden, famously said: *"We kill people based on metadata"* [Col14]. We additionally see that these adversaries target encrypted protocols and for example seek to exploit

---

[*] This work was published at Africacrypt 2019 as joint paper [AMW19] entitled *Tiny WireGuard Tweak* with Chloe Martindale and Peter Wu.

properties of handshakes, which may allow them to launch other subsequent attacks. These types of attacks are documented in the publication of numerous internal documents [Lan13, Lan14, Ada14] that show attacks, claims, and results against a number of VPNs and other important cryptographic protocols. Development of quantum computers for attacking cryptographic protocols is explicitly a budget line item [GM13a]. We consider it prudent to analyze WireGuard as a protocol that is, among others, of interest to these adversaries.

We consider nation-state mass surveillance adversaries (for example NSA [BPR14, Bie15] using XKeyscore [Gre13d]) as one of the primary adversaries to users of encrypted network tunnels, and we find that WireGuard will be vulnerable when these adversaries gain access to a quantum computer (see Section 6.5 for details). This is primarily due to the fact that large-scale [Hog15] surveillance data sets which contain logged encrypted traffic are explicitly kept for later attempts at decryption [Erw15].

We also consider less powerful adversaries which are directly coercive, oppressive, or political (COPs). These adversaries are able to take possession of any endpoint, such as through theft or other ill-gotten means, which includes a long-term public static cryptographic key pair. This type of attack is regularly carried out against VPN providers and is commonly understood as a kind of compulsion [DC05] attack.

6.3 — WireGuard overview

In this section we present an overview of the WireGuard protocol, briefly consider relevant implementations, and discuss traffic analysis considerations.

6.3.1 – WireGuard implementations. WireGuard is implemented in multiple languages and is easy to understand. The primary implementation is available as a patch to the Linux kernel and is written in C [Don19d]. Implementations targeting MacOS and iOS [Don19e], Android [Don19c], and Windows [Don19f] use the wireguard-go [Don19a] implementation which is written in the Go programming language. An experimental implementation in the Rust programming language is also available, wireguard-rs [Don19b].

We have implemented a user space Python implementation for experimentation using Scapy [Bio10] for use on GNU/Linux and a protocol dissector [Wu18] for WireGuard in Wireshark [Wir21], a software program that can capture and analyze network traffic. Our implementations are based on the published WireGuard paper [Don17b] and the evolving white paper [Don18b].

6.3.2 – WireGuard as a tunneling protocol. WireGuard is a point-to-point protocol for transporting IP packets. It uses the UDP protocol for transporting protocol messages. It is implemented as a device on common operating systems and users of WireGuard route IP packets into the WireGuard device to securely send those packets to their WireGuard peer. WireGuard does not have state for any IP packets that it transmits and it does not re-transmit packets if they are dropped by the network.

To start using the WireGuard protocol, a user must first generate a long-term static Curve25519 [Ber06] key pair and acquire the long-term static public key of their respective peer. This precondition for running the WireGuard protocol is different from common Internet protocols as users *must* exchange these keys out of band. This is in contrast to services such as OpenVPN which may only need to exchange a user name or password for access control reasons. Example methods of distributing WireGuard keys include using a camera on a smart phone to import the peer public keys with a QR code, or by manually entering the data. This *must* be done before attempting to run the WireGuard protocol and the would-be agents running the protocol are designed to not emit packets to parties which do not have possession of previously exchanged public keys. Users are also required to exchange a DNS name or an IP address along with a UDP port number for at least one of the two parties. To use the WireGuard tunnel, the peers additionally have to exchange the expected *internal* IP addressing information for their respective WireGuard tunnel endpoints. This again is in contrast to other VPN solutions which usually include some sort of automatic IP addressing scheme to ease automatic configuration of internal tunnel endpoint addresses.

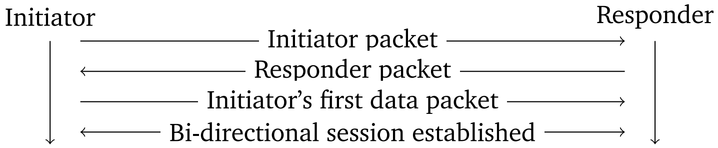

Initiator
Responder
Initiator packet
Responder packet
Initiator's first data packet
Bi-directional session established

Figure 6.1: Informal protocol narration of the 1.5 Round Trip Time (1.5-RTT) handshake valid for a ninety second session; parties may change roles in subsequent sessions; for additional information see Figure 6.7 and Algorithm 6.1

After configuring the endpoints with the respective public keys and IP addresses, peers will be able to create new cryptographic WireGuard sessions with each other as shown in Figure 6.1.

6.3.3 – WireGuard's cryptographic handshake. The Noise Protocol Framework [Per18] abstractly defines different Diffie-Hellman handshakes with different security, and privacy properties for use in cryptographic protocols. Protocol designers select a Noise Protocol pattern and then select the ideal abstract handshake properties. They must then select concrete objects such as an authenticated encryption scheme and a Diffie-Hellman primitive. WireGuard's cryptographic handshake [Don18b] is a variant of IKpsk2 pattern from the Noise Protocol [Per18, Section 9.4] framework. A WireGuard handshake consists of the initiator sending an initiation message (see Figure 6.3) and the responder replying with a corresponding responder message (see Figure 6.4).

WireGuard selected Curve25519 [Ber06] for Diffie-Hellman non-interactive key exchange messages, BLAKE2s [SA15a] for hashing operations, HKDF [KE10b] as the key derivation function (KDF), and ChaCha20Poly1305 [NL18] for authenticated encryption with additional data (AEAD).

WireGuard additionally augments the Noise protocol in certain areas that weaken conventional security assumptions relating to identity hiding; WireGuard reduces the identity hiding properties of the Noise IK protocol as part of a trade-off strategy to reduce computational costs and to resist detection by untargeted Internet-wide scanning. The popular Wireshark traffic analysis program displays a peer's identity and associates it with flows of traffic. We observe that preconditions of the protocol more closely resemble the Noise KK pattern; KK assumes that both parties know their peer's respective long-term static public key while IK assumes that only the responder's long-term static public key is known by the initiator. However, it is strictly weaker than the KK pattern in that the initiator always reveals their own long-term static public key identity to the responder, and thus to the network, encrypted to the responder's long-term public key. Unlike other protocols, the roles of initiator and responder do also reverse [Don18b]. This happens automatically when the responder attempts to send a data packet without a valid session.

6.3.4 – Handshake details. The initiator's long-term static public key is encrypted using the ChaCha20Poly1305 AEAD using a key derived from the responder's long-term static public key and a per-session ephemeral Curve25519 key pair generated by the initiator. The resulting ciphertext is decrypted, and the public key of the initiator is found, and matched to a corresponding data structure previously initialized for cryptographic operations on the responder side; see Algorithm 6.1 for details. In Section 6.5.2, we describe an attack based on the transmission of the encrypted long-term static public key.

Notes on Algorithm 6.1:

- As in the WireGuard protocol, we use the following notation for symmetric encryption with a nonce and additional authenticated data (AEAD): $\mathsf{ciphertext} = \mathsf{aead\text{-}enc}(\mathsf{key},\ \mathsf{nonce},\ \mathsf{message},\ \mathsf{associated\ data})$.
- Algorithm 6.1 gives a *simplified* version of the WireGuard key agreement process; the only fundamental simplifications that we have applied are:
  - We introduce Laura and Julian as parties in the role of Initiator and Responder.
  - Compressing the application of multiple hash function operations from $H(H(x)\|y)$ to a single $H(x\|y)$.
  - Omission of some constants in the initial hash and KDF salt.
  - Omission of details about construction of the 96-bit nonce. This value also serves as a counter for replay detection within a given session.
  - Compressing the application of multiple KDF's to a set of variables to the application of a single KDF to the set of variables.

6.4 — Traffic analysis

WireGuard traffic visible to a third party observer is subject to trivial fingerprinting and confirmation that the WireGuard protocol is in use. The protocol is not designed to resist traffic analysis: session identifiers, sequence numbers, and other values are visible. For any surveillance adversary, writing a comprehensive network protocol dissector is quick work as evidenced in our Wireshark and Scapy implementations. There are four message types. Three of these types have a fixed length and each has static values which act as distinguishers or network selectors [Pri14]. The fourth type has variable length, it additionally has static distinguishers and is linkable to other packets in any given flow. WireGuard does not attempt to hide that the WireGuard protocol is in use from a surveillance adversary, and it additionally does not attempt to hide information that allows sessions within network flows to be distinguished. WireGuard does attempt to resist active probing by requiring any initiating party to prove knowledge of the long-term static public key of the responder.

6.4.1 – Example WireGuard protocol run. To create a WireGuard session, the protocol is broken into several phases. The initiating party is called an *initiator*, and the receiving party which must be reachable, is called the *responder*. The first phase is a handshake protocol described in detail in Section 6.3.3, and the second phase is a time-limited data-

Algorithm 6.1 Simplified WireGuard key agreement process

Public Input: Curve25519 $E/\mathbb{F}_p$, base point $P \in E(\mathbb{F}_p)$, hash function H, an empty string ϵ, key derivation function KDF_n returning n derived values indexed by n, and a MAC function Poly1305.

Secret Input (Laura): secret key $\mathsf{sk}_L \in \mathbb{Z}$, public key $\mathsf{pk}_L = \mathsf{sk}_L \cdot P \in E(\mathbb{F}_p)$, Julian's pre-shared public key $\mathsf{pk}_J \in E(\mathbb{F}_p)$, shared secret $\mathsf{s} = \mathsf{DH}(\mathsf{sk}_L, \mathsf{pk}_J)$, message $time$, PSK $Q \in \{0,1\}^{256}$; $Q = 0^{256}$ by default.

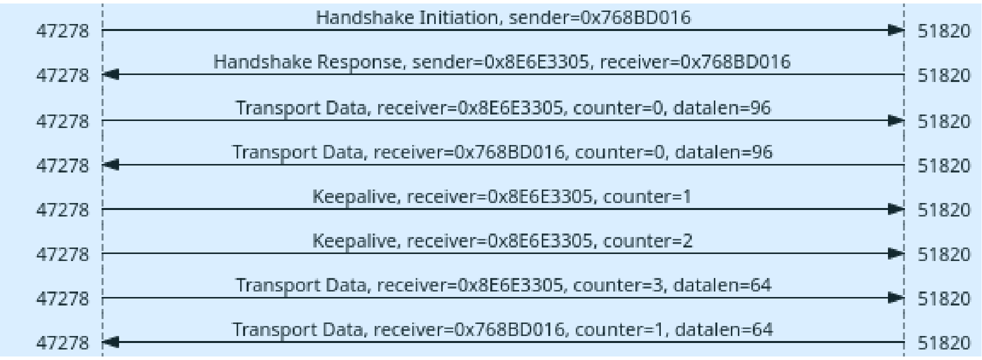
Secret Input (Julian): secret key $\mathsf{sk}_J \in \mathbb{Z}$, public key $\mathsf{pk}_J = \mathsf{sk}_J \cdot P \in E(\mathbb{F}_p)$, Laura's pre-shared public key $\mathsf{pk}_L \in E(\mathbb{F}_p)$, shared secret $\mathsf{s} = \mathsf{DH}(\mathsf{sk}_J, \mathsf{pk}_L)$, PSK $Q \in \{0,1\}^{256}$; $Q = 0^{256}$ by default.

Output: Session keys.

```
Both parties choose ephemeral secrets: $\mathsf{esk}_L \in \mathbb{Z}$ for Laura, $\mathsf{esk}_J \in \mathbb{Z}$ for Julian.
Laura publishes $\mathsf{epk}_L \leftarrow \mathsf{esk}_L \cdot P$.
Laura computes $\mathsf{se}_{JL} \leftarrow \mathsf{esk}_L \cdot \mathsf{pk}_J$; Julian computes $\mathsf{se}_{JL} \leftarrow \mathsf{sk}_J \cdot \mathsf{epk}_L$.
Both parties compute $(\mathsf{ck}_1, \mathsf{k}_1) \leftarrow \mathsf{KDF}_2(\mathsf{epk}_L, \mathsf{se}_{JL})$.
Laura computes $\mathsf{h}_1 \leftarrow H(\mathsf{pk}_J \| \mathsf{epk}_L)$.
Laura computes and transmits enc-id $\leftarrow$ aead-enc$(\mathsf{k}_1, 0, \mathsf{pk}_L, \mathsf{h}_1)$.
Julian decrypts enc-id with aead-dec$(\mathsf{k}_1, 0, \text{enc-id}, \mathsf{h}_1)$ and verifies that the resulting value $(\mathsf{pk}_L)$ is valid user's public key; aborts on failure.
Both parties compute $(\mathsf{ck}_2, \mathsf{k}_2) = \mathsf{KDF}_2(\mathsf{ck}_1, \mathsf{s})$.
Laura computes $\mathsf{h}_2 \leftarrow H(\mathsf{h}_1 \| \text{enc-id})$.
Laura computes and transmits enc-time $\leftarrow$ aead-enc$(\mathsf{k}_2, 0, time, \mathsf{h}_2)$.
Both parties compute $\mathsf{pkt} \leftarrow \mathsf{epk}_L \| \text{enc-id} \| \text{enc-time}$.
Laura computes and transmits $\mathsf{mac1} \leftarrow \mathsf{MAC}(\mathsf{pk}_J, \mathsf{pkt})$.
Julian verifies that $\mathsf{mac1} = \mathsf{MAC}(\mathsf{pk}_J, \mathsf{pkt})$; aborts on failure.
Julian computes $time$ = aead-dec$(\mathsf{k}_2, 0, \text{enc-time}, \mathsf{h}_2)$; aborts on failure.
Julian transmits $\mathsf{epk}_J \leftarrow \mathsf{esk}_J \cdot P$.
Laura computes $\mathsf{se}_{LJ} \leftarrow \mathsf{sk}_L \cdot \mathsf{epk}_J$; Julian computes $\mathsf{se}_{LJ} \leftarrow \mathsf{esk}_J \cdot \mathsf{pk}_L$.
Laura computes $\mathsf{e} \leftarrow \mathsf{esk}_L \cdot \mathsf{epk}_J$; Julian computes $\mathsf{e} \leftarrow \mathsf{esk}_J \cdot \mathsf{epk}_L$.
Both parties compute $(\mathsf{ck}_3, \mathsf{t}, \mathsf{k}_3) \leftarrow \mathsf{KDF}_3(\mathsf{ck}_2 \| \mathsf{epk}_J \| \mathsf{e} \| \mathsf{se}_{LJ}, Q)$.
Julian computes $\mathsf{h}_3 \leftarrow H(\mathsf{h}_2 \| \text{enc-time} \| \mathsf{epk}_J \| \mathsf{t})$.
Julian computes and transmits enc-e $\leftarrow$ aead-enc$(\mathsf{k}_3, 0, \epsilon, \mathsf{h}_3)$.
Laura verifies that $\epsilon$ = aead-dec$(\mathsf{k}_3, 0, \text{enc-e}, \mathsf{h}_3)$.
Both parties compute shared secrets $(T_i, T_r) \leftarrow \mathsf{KDF}_2(\mathsf{ck}_3, \epsilon)$.
return $(T_i, T_r)$.
```

transfer window. The third phase is reached when a time limit or a data-transfer limit is reached, at which point a new cryptographic session is established. Unlike other cryptographic protocols, the WireGuard protocol has no session renegotiation, peers simply start again as if they have never had a session in the first place.

After a successful handshake, once the initiator has received a responder message, it may proceed to send transport data messages (see Figure 6.6) which contain encrypted IP packets. The responder is only permitted to send data messages after successfully receiving and authenticating the transport data packet sent by the initiator. Data messages with an encrypted empty payload act as Keep-Alive messages. These are trivially distinguishable messages by their type and length as shown in Figure 6.2.



```
47278 Handshake Initiation, sender=0x768BD016 -> 51820
47278 <- Handshake Response, sender=0x8E6E3305, receiver=0x768BD016 51820
47278 Transport Data, receiver=0x8E6E3305, counter=0, datalen=96 -> 51820
47278 <- Transport Data, receiver=0x768BD016, counter=0, datalen=96 51820
47278 Keepalive, receiver=0x8E6E3305, counter=1 -> 51820
47278 Keepalive, receiver=0x8E6E3305, counter=2 -> 51820
47278 Transport Data, receiver=0x8E6E3305, counter=3, datalen=64 -> 51820
47278 <- Transport Data, receiver=0x768BD016, counter=1, datalen=64 51820
```

Figure 6.2: Flow graph between two WireGuard peers as seen in Wireshark

An example interaction taken from a packet capture between two WireGuard peers can be found in Figure 6.2, and an informal protocol narration in Figure 6.1. If either initiator or responder are under heavy computational load, they may send a Cookie message (see Figure 6.5) in response to an initiation or responder message without making further progress in completing the handshake. The recipient of a Cookie message should decrypt the cookie value and use it to calculate the MAC2 value for use in the next handshake attempt. It will not re-transmit the same handshake message under any circumstances. If a handshake is unsuccessful, the initiator will try to start a new handshake.

There is no explicit error or session-tear-down signaling. A session is invalidated after a fixed duration of time; session lifetimes are currently around ninety seconds.

6.4.2 – Packet formats. We display the four packet formats. The protocol includes only these four wire message formats, though there is an implied fifth type: an empty data message may be used as keep alive message. Each message is encapsulated entirely inside of an IP packet with UDP payload.

In Figure 6.3, the initiator message is shown. It is a fixed-size frame of 148 bytes. The MAC2 field is set to zero unless the sender has received a Cookie message before. This message is larger than the responder's message intentionally to prevent misuse such as amplification attacks using forged source addresses.

In Figure 6.4, the responder message is shown. It is a fixed-sized frame of 92 bytes. Unlike the initiator packet, it does not contain a long term static public key.

In Figure 6.5, the cookie message is shown. It is a fixed-sized frame of 64 bytes. This is not used for each run of the WireGuard protocol. This message is only sent by the

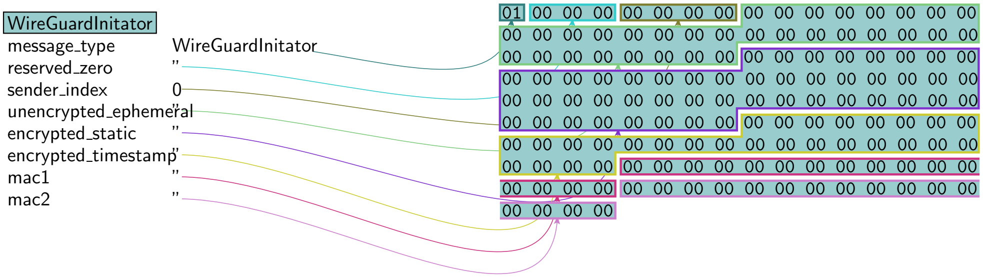

WireGuardInitator
message_type
reserved_zero
sender_index
unencrypted_ephemeral
encrypted_static
encrypted_timestamp
mac1
mac2
WireGuardInitator
"
0
"
"
"
"
"
01 00 00 00 00 00 00 00

Figure 6.3: 148 byte initiator packet payload

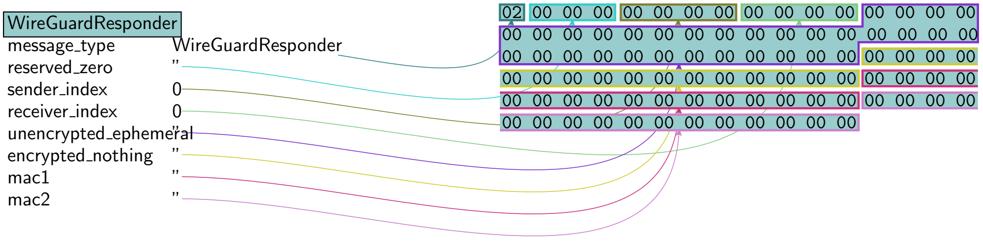

WireGuardResponder
message_type
reserved_zero
sender_index
receiver_index
unencrypted_ephemeral
encrypted_nothing
mac1
mac2
WireGuardResponder
"
0
0
"
"
"
"
02 00 00 00

Figure 6.4: 92 byte responder packet payload

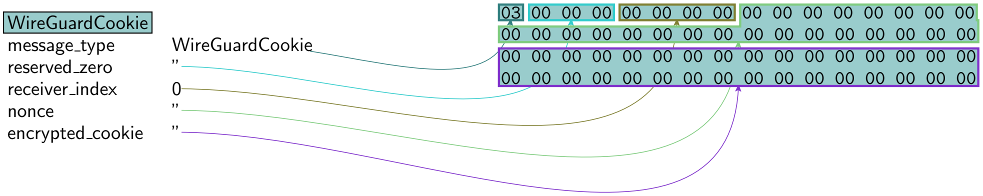

WireGuardCookie
message_type
reserved_zero
receiver_index
nonce
encrypted_cookie
WireGuardCookie
"
0
"
"
03 00 00 00

Figure 6.5: 64 byte Cookie packet payload

initiator or responder when they are “under load”. The recipient must decrypt the cookie value and store it for inclusion in future handshake messages.

While all handshake messages (Figure 6.3, Figure 6.4, Figure 6.5) have fixed lengths, the Transport Data message (Figure 6.6) has a variable length. At minimum it is 32 bytes in length. This includes the Transport Data message headers and the authentication tag for the encrypted payload. For any given WireGuard protocol run, the maximum size of a generated UDP packet depends on the maximum transmission unit (MTU) of the network interface. These are typically much smaller than the theoretical limits of an IP packet.

The UDP layer has a theoretical maximum length of $2^{16}-1$, this length also includes eight bytes of the UDP header so the actual maximum length for the UDP payload is $2^{16}-1-8$ bytes. The theoretical maximum length for Transport Data messages is shown in Table 6.1.

While WireGuard itself does not impose a maximum length, implementations on various platforms might be constrained by their environment. For example, the Linux kernel does not support IPv6 Jumbograms [Dum14] and FreeBSD currently does not support IPv6 Jumbograms with UDP due to the lack of a physical medium [Fre].

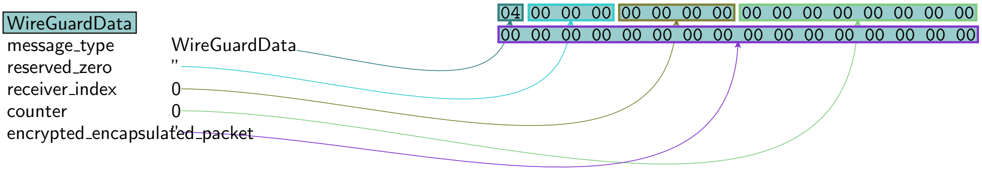

WireGuardData
message_type
reserved_zero
receiver_index
counter
encrypted_encapsulated_packet
WireGuardData
"
0
0
04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Figure 6.6: Variable length (32 up to $\infty + 16$) byte data packet payload; see Table 6.1 for implementation specific notes.

$2^{16} - 1 - 8$	IPv4 with fragmentation
$2^{16} - 1 - 20 - 8$	IPv4 without fragmentation nor IP options
$2^{16} - 1 - 40 - 8$	IPv6 without extension headers
$2^{32} - 1 - 40 - 8 - 8$	IPv6 with Jumbograms

Table 6.1: Theoretical maximum sizes for UDP payloads

6.5 — Security and privacy issues

We consider both the mass surveillance adversary and the less powerful local adversary conducting targeted attacks from Section 6.2.

Initial handshake message creation and processing

Laura **Julian**

$\mathsf{pk}_L, \mathsf{sk}_L, time$, secret key Q $\mathsf{pk}_J, \mathsf{sk}_J$

.... Out-of-band key exchange: $\mathsf{pk}_L, \mathsf{pk}_J$, PSK Q ....

$(\mathsf{epk}_L, \mathsf{esk}_L) = \mathsf{EphemeralKey}()$

Compute enc-id, enc-time, mac1

$$\xrightarrow[\text{Initiator packet}]{\mathsf{epk}_L,\ \text{enc-id, enc-time, mac1}}$$

.... Responder receives initiator packet ....

Compute pkt, verify mac1

Compute emphemeral DH

Decrypt enc-id to a known pk

Find session for resulting pk

Decrypt enc-time to get $time$

$\mathsf{VerifyAntiReplay}(time)$

.... Handshake continues ....

Figure 6.7: Informal protocol narration of sending and receiving an initiator packet. (For definitions of terms and details on how to compute, decrypt, and verify, see Algorithm 6.1)

6.5.1 – Identity hiding weakening. Throughout this section, suppose, as was justified in Section 6.2 to be a realistic situation, that a WireGuard user has released its long-term static public key. We analyze a handshake involving this user with this user in the role of responder.

The initiation packet contains the static public key of the initiator and it is encrypted as previously described with an ephemeral key pair used in conjunction with the responder's static key pair. The initiation packet is augmented with what WireGuard's design describes as a MAC. Under our assumptions, the input, which is an initiator or a responder packet, and the MAC key, which is the static public key of the receiving party, are both *public* values. Third party observers are able to passively confirm the identity of both peers when their public keys are known to the observer. This is strictly worse than NoiseIK's identity hiding properties and allows non-sophisticated attackers to link known static public keys to individual flows of traffic.

Ostensibly the additional MAC over the whole packet is done primarily as a verification step: to prevent arbitrary packets (e.g. from an adversary) from causing the responder to compute a Diffie-Hellman key-exchange. This is a known deficiency in OpenVPN [Don18a].

The MAC check also prevents practical Internet-wide scans from finding *unknown* WireGuard responders. While a verification step may be necessary to prevent unknown parties from exhausting resources or forcing a responder message, this additional MAC verification method is strongly divergent from the identity hiding properties of the Noise IK pattern; because of this identity hiding property, it is easier for a quantum adversary to attack, as we show below.

A simple shared secret value, set either on a per-site or per-peer basis would provide a similar protection without revealing the identity of one or both of the peers.

6.5.2 – Quantum attack. Consider an attacker capable [1] of running Shor's algorithm [Sho94]. Shor's algorithm breaks the discrete logarithm problem in any group in time polynomial in the size of the group; observe that this includes elliptic curve groups. Suppose that the long-term static public key of some WireGuard user U_0 is known to an adversary. We show in Algorithms 6.2 and 6.3 that in this situation, Shor's algorithm will apply to users of the WireGuard protocol, as given in Algorithm 6.1.

Recall from Section 6.4 that network traffic is visible to a third-party observer. In particular, an adversary can detect when a handshake takes place between U_0 and any other WireGuard user. We describe in Algorithm 6.2 how to extract the long-term static secret key of any initiator with a quantum computer when U_0 is the responder.

Of course after computing the ephemeral keys, an adversary who has access to the static secret and public keys of both the initiator and the responder of a WireGuard handshake can completely break the protocol (assuming the responder U_0 and the initiator use the default WireGuard settings, i.e. no PSK).

Now suppose an adversary wishes to attack some user U_n. Suppose also that there exists a *traceable path* from U_0 to U_n, that is, if by analyzing the traffic flow the adversary can find users $U_1, \ldots, U_{n-1}$ for which every pair of 'adjacent' users U_i and U_{i+1} have performed a WireGuard handshake. We show in Algorithm 6.3 how the adversary can then compute U_n's long-term static key pair. Recall from Section 6.4 that the information of which pairs of users have performed a WireGuard handshake is freely available; if such a path exists then an adversary can easily find it.

---

[1] See [RNSL17] for a recent estimate of the resources needed by an attacker to carry out such an attack using Shor's algorithm.

An important remark on this attack: if two WireGuard users do not publish their static public keys, and *both* users do not interact with any other WireGuard users, then this attack does not apply to those two users.

Algorithm 6.2 Extract Initiator's Long-term Static Key Pair

```
Input: Long-term static public key $\mathsf{pk}_J$ of the responder; Ephemeral public key $\mathsf{epk}_L$ of
    the initiator (transmitted over the wire in Step 2 of Algorithm 6.1); enc-id as sent
    over the wire by the initiator in Step 6 of Algorithm 6.1.
Output: Long-term static key pair $\mathsf{sk}_L, \mathsf{pk}_L$ of the initiator.
Using Shor's algorithm, compute $\mathsf{esk}_L$ from $\mathsf{epk}_L$.
Compute $k_1$ and $h_1$ as in Steps 4 and Steps 5 respectively of Algorithm 6.1.
Compute $\mathsf{pk}_L = \text{aead-dec}(k_1, 0, \text{enc-id}, h_1)$.
Compute $\mathsf{sk}_L$ from $\mathsf{pk}_L$ using Shor's algorithm.
    return $\mathsf{sk}_L, \mathsf{pk}_L$.
```

Algorithm 6.3 Extract User U_n's Long-term Static Key Pair

```
Input: Long-term static public key of some WireGuard User $U_0$; A traceable path from
    $U_0$ to WireGuard User of interest $U_n$.
Output: Long-term static key pair of WireGuard User $U_n$.
for $i := 0, \dots, n-1$ do
    $U_i \leftarrow$ Responder (without loss of generality, c.f. Section 6.3.3).
    $U_{i+1} \leftarrow$ Initiator (also without loss of generality).
    Compute long-term static key pair of $U_{i+1}$ using Algorithm 6.2.
end for
    return Long-term static key pair of $U_n$.
```

6.5.3 – A brief comment on extra security options. In Section 6.5.2 we analyzed the *default* use of the WireGuard protocol. There is an option open to WireGuard users to also preshare another secret key, i.e., to use a PSK Q as an additional input for the KDF in Step 18 of Algorithm 6.1. If the user does not configure a PSK, the default value ($Q = 0^{256}$) will be used.

Use of a secret PSK will not prevent a quantum adversary from computing $\mathsf{sk}_L, \mathsf{pk}_L$ using the method described in Section 6.5.2. It does however prevent compromise of session keys T_i and T_r in Step 22 of Algorithm 6.1 as the adversary no longer has enough information to compute ck_3 in Step 18 of Algorithm 6.1.

A prudent user may still be concerned about an adversary stealing their PSK; the tiny protocol tweak presented in Section 6.6 addresses this concern as well as protecting those who use the default mode of the WireGuard protocol.

Of course our tweak cannot protect against an adversary who steals the static long-term public key of both the initiator and the responder in a WireGuard handshake.

6.6 — Blinding flows against mass surveillance

We propose a tiny tweak to the WireGuard handshake which thwarts the quantum attack outlined in the previous section: In Step 6 and Step 7 of Algorithm 6.1, replace pk_L by $H(\mathsf{pk}_L)$. We suggest to use BLAKE2s as the hash function H as it is already used elsewhere in WireGuard. Naturally, the unhashed static public key pk_L of the initiator has still been exchanged out-of-band, so the responder can still perform Diffie-Hellman operations with the initiator's static public key pk_L, and is able to compute the hash $H(\mathsf{pk}_L)$. In Step 7 and Step 16 of Algorithm 6.1, the responder will use the decrypted value $H(\mathsf{pk}_L)$ to look up the corresponding key pk_L. The hashing process conceals the algebraic structure of the static public key of the initiator and replaces it with a deterministic, predictable identifier. This requires no extra configuration information for either of the peers. BLAKE2s is a one-way hashing function and a quantum adversary cannot easily [Wie04] deduce the initiator's static public or secret key from this hash value unless the hash function is broken.

An attacker as described in Section 6.5.2 may confirm a guess of a known long-term static public key. If the guess is correct, they may carry out the attack as in the unchanged WireGuard protocol. However, the tweak protects sessions where the public keys are not known.

We claim only *transitional security* with this alteration. That is, that a future quantum adversary will not be able to decrypt messages sent before the advent of practical quantum computers, if the messages are encrypted via an updated version of WireGuard that includes our proposed tweak. The tweaked protocol is not secure against active quantum attacks with knowledge of both long-term static public keys and a known PSK value. With knowledge of zero or only one long-term static public key, the protocol remains secure. A redesign of the WireGuard protocol to achieve full post-quantum security is still needed.

There are of course other choices of values to replace the static public key in Step 6 and Step 7 of Algorithm 6.1 to increase security. One alternative choice of value is an empty string, as in the case with the message sent in response to initiator packets by the responder. This would change the number of trial decryptions for the responder for initiator messages to $\mathcal{O}(n)$ where n is the number of configured peers. This change would allow any would-be attacker to force the responder to perform many more expensive calculations. It would improve identity hiding immensely but at a cost that simply suggests using a different Noise pattern in the first place. A second alternative choice of value is a random string which is mapped at configuration time, similar to a username or a numbered account, which is common in OpenVPN and similar deployments. This provides $\mathcal{O}(1)$ efficiency in lookups of session structures but with a major loss in ease of use and configuration. It would also add a second identifier for the peer which does not improve identity hiding. Both alternative choices have drawbacks. The first method would create an attack vector for unauthenticated consumption of responder resources and the second method would require additional configuration. Both weaken the channel binding property of Noise [Per18, Chapter 14] as the encrypted public key of the initiator is no longer hashed in the handshake hash. The major advantage of our proposed choice is that it does not complicate configuration, nor does it require a wire format change for the WireGuard protocol. Assuming collision-resistance of the hash function, the channel binding property is also preserved. Our proposal concretely improves the confidentiality of the protocol without increasing the computation in any handshake. It increases the computation for peer configuration by only a single hash function for each configured public key.

This change does not prevent linkability of flows as it exchanges one static identifier for another, and it does preclude sharing that identifier in a known vulnerable context.

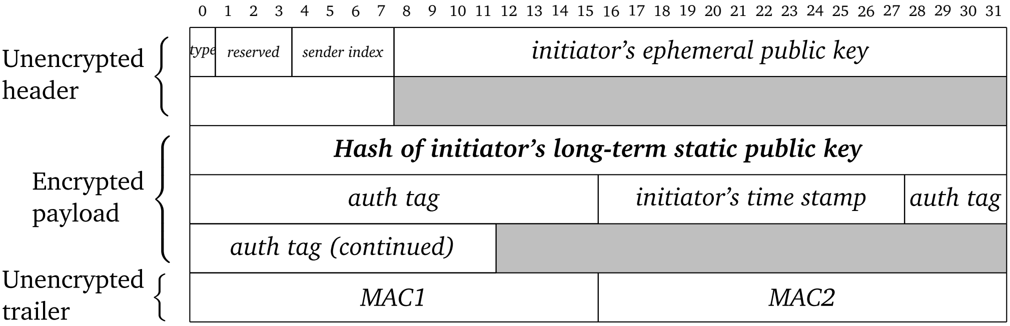

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Unencrypted header
type
reserved
sender index
initiator's ephemeral public key
Encrypted payload
Hash of initiator's long-term static public key
auth tag
initiator's time stamp
auth tag
auth tag (continued)
Unencrypted trailer
MAC1
MAC2

Figure 6.8: Tweaked initiator packet (in bytes)

6.6.1 – Modified protocol costs. Our modification obviously requires implementation changes. We study the effect on the proposed Linux kernel implementation as outlined in the WireGuard paper [Don18b] as well as the effect on the alternative implementations.

The hash function input of the initiator's static public key and the output value have an identical length, thus the wire format and internal message structure definitions do not need to change to accommodate the additional hash operation.

Initiators only have a single additional computational cost, calculation of the hash over their own static public key. This could be done during each handshake at no additional memory cost, or during device configuration which only requires an additional 32 bytes of memory in the device configuration data structure to store the hash of the peer's long-term static public key.

Responders must be able to find the peer configuration based on the initiation handshake message since it includes the peer's static public key, optional PSK, permitted addresses, and so on. In the unmodified protocol, a hash table could be used to enable efficient lookups using the static public key as table key. At insertion time, a hash would be computed over the table key. The Linux kernel implementation uses SipHash2-4 [AB12] as hash function for this table key [Don18b, Section 7.4]. Our modification increases the size of the per-peer data structure by 32 bytes and requires a single additional hash computation per long-term static public key at device configuration time. There are no additional memory or computational costs during the handshake.

The wireguard-go [Don19a, device/device.go] implementation uses a standard map data type using the static public key as map key. Again, a single additional hash computation is required at configuration time with no additional memory usage.

Recall that WireGuard is based on the Noise Protocol Framework. Our modification is not compatible with the current version of this framework, and thus implementations that rely on a Noise library to create and process handshake messages must be changed to use an alternative Noise implementation. This affects the Rust implementation [Don19b].

6.6.2 – Alternative designs and future work. In theory, an alternative WireGuard implementation could accept any initiator that connects to it and successfully completes the handshake. Additional authorization could then be performed after the handshake. Our modification would make it impossible to create such implementations as it ensures that the assumed pre-condition of requiring an out-of-band exchange of long-term static public key is not violated.

Our proposed modification is generic and also applies to other protocols based on the Noise IK pattern. A new *pattern modifier* could be defined in the Noise specification that enables new protocols to improve transitional post-quantum security in the case where static public keys have been exchanged before, and only an identity selector needs to be transmitted.

6.7 — Conclusions

We show that a future adversary with access to a quantum computer, historic network traffic protected by WireGuard, and knowledge of a WireGuard user's long-term static public key can likely decrypt many WireGuard users' historic messages when the optional PSK was not used or was compromised. We present a simple solution to this problem: hashing the long-term static public key before it is sent encrypted over the wire, resulting in the destruction of the algebraic structure of the elliptic-curve point which otherwise could be exploited by quantum computers via Shor's algorithm. The resulting hashed public key is the same size as the original public key and does not increase the size of any of the protocol messages. The required input for a quantum adversary to run Shor's algorithm would not be available from the network flow alone and it would thwart such an attacker from using a database of network flows to decrypt those very same flows. Targeted quantum attacks would still be possible in the case that the long-term keys of both parties, initiator and responder, are known. Active quantum attacks may still be possible, but our alteration provides transitional security. Our improvement requires zero extra bytes of data transmitted on the wire, potentially zero or 32 extra bytes for each peer data structure in memory, and completely negligible computational costs for cooperating honest parties.

CHAPTER 7

Vula

7.1 — Introduction

Mass surveillance [Eur18,Eur78,Eur06,Eur84,Eur10,Eur87,Eur15,Eur16] is not only a concern for backbone [PRS13b] networks; every network [Stö13] is potentially a target. Local Area Network (LAN) security in the form of protection against surveillance is generally lacking in home, small business, and enterprise networks. We propose Vula: a protocol for automatically securing the LAN against eavesdropping and traffic tampering by other users, and/or network infrastructure equipment. Vula combines a secure tunneling protocol for secure point-to-point communications between Vula peers, multicast DNS for publishing and discovery of Vula peer associated metadata, along with easy Vula peer verification. Vula automatically builds tunnels directly between participating Vula peers on the same local network segment. We have selected the WireGuard Virtual Private Network (VPN) protocol outlined in Section 7.4 for our Vula tunneling protocol. Unlike most deployments of WireGuard, Vula does not require the use of a third party located on another network. Users may additionally discover and/or verify Vula peers using user-friendly QR codes [BSI09] for protection against active adversaries.

We consider adversarial issues present in WPA [RS19, VR20] encrypted home networks, open public WiFi hotspots, and business networks. We designed Vula as a smaller scale starting point to deal with adversarial issues present in larger networks such as Internet Exchange Points (IXP) [WM15] in mind, which are known targets [ND16] of mass surveillance [1]. This is not the main target of Vula or this paper, but Figure 7.7 and additionally Figure 7.3 produced by the FLENT [HJGHB17,Fle17] tool shows that WireGuard performance keeps up with Gigabit line speed without any trouble even on constrained systems. We note that users of *any* wireless or Ethernet network will benefit from the use of Vula [2]. With Vula's ability to be gradually deployed, every host has a notion of cryptographic identity, and we think that with this improvement it will be clearer how to solve the problem of Internet-wide end-to-end encryption without resorting to sending unencrypted IP packets, encrypted but unauthenticated IP packets, or any of the various Single Points of Failure (SPOFs) as described in Section 7.2.

7.1.1 – Motivation. Public and private personal networks are commonly deployed using wired Ethernet (802.3) or wireless LAN (802.11) standards without comprehensive protection against surveillance adversaries. Ethernet networks are commonly deployed in consumer and commercial contexts without encryption of any kind. Authentication [AMC+14] of end-user's computers may be combined with a protocol such as MACsec [IEE06] or WPA for link encryption between an end-user's computer and their immediate upstream Ethernet or wireless link. This combination of end-user authentication and link encryption does not provide end-to-end encryption [Dif83] between hosts on the same Ethernet segment or the same IP multicast broadcast domain. It additionally adds a per-user administrative overhead, necessitating network equipment-specific

---

[*] This work was previously unpublished. It is joint work with Leif Ryge.

[1] Two examples in the IXP community are London's LINX [Cam17] and Frankfurt's DE-CIX [Mei15] [Lan18b]. LINX denies that they are under such an order and they promise to reveal it if they learn about it [Cam17] while DE-CIX has been to court to fight against such an order. In the IXP community, it is commonly understood in private discussions that both are under secret orders to export large volumes of traffic that their IXP carries to their local mass surveillance adversary, GCHQ and BND respectively.

[2] Any point-to-point traffic between participating systems will be protected by WireGuard, and the protection for point-to-point traffic is much stronger than the protection afforded by an unencrypted wireless network. It is also stronger in some ways than the protections afforded by WPA or WPA2 such as forward secrecy.

administration which must further be specialized to support the required protocols. This may also create additional logging and other user data management complexity. Logs often present themselves as a tempting target for attacks [Sav20].

7.1.2 – The Vula Proposal. We propose a system which provides automatic end-to-end encryption of IP packets with transitionally [SWZ16b] post-quantum [NIS21] forward secrecy [MvOV96], without requiring centralized administration or specialized equipment beyond a basic Ethernet hub, switch, and/or wireless LAN. We also experiment with other Ethernet devices such as USB and Thunderbolt [MRG⁺19, Ruy20, Ruy22] attached Ethernet devices.

The purpose of Vula is to enhance the security of LAN traffic between hosts on the same IP multicast broadcast domain. We have taken great care to avoid introducing new vectors for host compromise in the design and implementation of Vula. Overall, Vula reduces the attack capabilities of various adversaries by following the principle of least authority (POLA [Mil06]). We use but to the maximum extent possible we avoid trusting the local network infrastructure. For systems running an implementation of the Vula protocol, and which are participating in the Vula protocol on the local network segment, we reduce nearly all attack vectors against IP traffic interference between Vula peers to a denial-of-service vector. Vula maintains confidentiality for traffic of verified Vula peers, also known as participants, using the protocol, while maintaining backwards compatibility, and connectivity with non-participants. Traffic exchanged with non-participants of the Vula protocol remains unchanged, and may be optionally blocked, if desired.

The Vula protocol provides a number of properties:

0. No infrastructure required.
1. Functional across organizational boundaries.
2. Fully automatic: LAN traffic between hosts with Vula installed is protected without any configuration whatsoever.
3. Works on temporarily offline or airgapped networks (e.g.: link-local [CAG05] addressing on Ethernet, ad-hoc WiFi, Thunderbolt, etc).
4. Protects traffic using *existing* IP addresses (whether DHCP-assigned, link-local, or manually configured), so applications do not need to be reconfigured.
5. Protects entire IP packets regardless of sub-protocol (e.g.: UDP, TCP, ICMP).
6. Transitional post-quantum protection.

We consider relevant background and related work in Section 7.2. We specify our threat model in Section 7.3 in terms of two broad categories of network adversary capabilities: *passive adversaries* [MD05], which are those who can observe some or all of the network's traffic but who lack the ability or opportunity to inject packets into the network or to prevent packets from being delivered, and *active adversaries* [DY83, TNE08] who do not lack those abilities. We present Vula in Section 7.4, implementation details in Section 7.4.5 and in Section 7.4.9, with additional performance measurements in Section 7.5, security evaluation in Section 7.6, and finally conclusions in Section 7.7.

7.2 — Background and related work

Previous attempts to provide security for wired and wireless networks are myriad but in practice have largely failed to protect end-users from commonly understood surveillance adversaries such as corporate or government surveillance programs.

The general state of affairs for consumer or small business connections provided by Internet service providers is to require a modem of some kind. This modem usually acts as a media converter; examples such as DOCSIS [Ric15] or VDSL2 [EO06] convert their respective uplink technology into an Ethernet network or a wireless network, or both. The modem device generally either acts as an IP router, or as a router with network address translation (NAT), often with basic packet filtering capabilities. Often these modems provide both Ethernet and wireless LAN capabilities which are commonly configured as a single bridged network with a single multicast domain. The security of such networks from the perspective of a surveillance adversary often hinges on the strength of a wireless passphrase, with no protection of the Ethernet side beyond physical access restrictions. Worse, even with a strong passphrase, the ability to trivially predict factory-initialized or even user-chosen long term cryptographic keys from wireless routers has been available to unskilled adversaries for years [Gei16,LMV15,Vie11]. In a consumer wireless network deployment, one out of dozens of public vulnerabilities may be practically exploitable without extensive knowledge requirements.

Previous attempts to create automatic or opportunistic [Lan09] end-to-end encryption with IPsec [FK11] have generally foregone authentication, and attempt to solve a similar set of problems at Internet scale by simply attempting to build IPsec connections to every host or network block. Minimal work [FKMS20] has been done on post-quantum IPsec. Alternative authentication using DNS is also possible for highly technical users who have end to end reachability such as a routable IPv4 or IPv6 network address, and who are able to control their forward and reverse DNS. This is not a common situation for many Internet users who sit behind a carrier-grade NAT, or where their traffic is filtered to prevent running of services without permission from their ISP.

Our proposed protocol attempts to solve a similar and the related set of problems at a smaller scale without any trusted third parties, and without attempting to create trust relationships between people who are unable to meet and verify cryptographic keys.

7.2.1 – Related protocols: 802.1x and MACsec. Protocols primarily deployed in corporate and academic environments center around access-control in an attempt to address some security concerns posed by adversaries with or without permitted access to the LAN.

These networks generally provide authentication, authorization, and accounting (AAA) services. A popular example in academic environments is the Eduroam [WWW15] network which uses WPA2-Enterprise. In passive adversary models, Eduroam protects against a local surveillance adversary by shifting the risk of the user's authentication traffic with EAP [VCB$^+$04] to their home academic institution. In active adversary models with Eduroam, client software may or may not [WWW15, Section 7] be configured correctly to provide active adversary protection.

Wireless networks with AAA services as part of wireless WPA2-Enterprise or Ethernet networks protected by 802.1x are often used without any additional security measures against potential surveillance adversaries after authentication. Ethernet networks may also be deployed with MACsec [IEE06] in an attempt to thwart adversary access to the network infrastructure, not as a matter of protecting against surveillance adversaries. MACsec provides link layer security in the form of encryption, integrity, and authenticity between a given client's Ethernet *interface and usually only the immediate upstream switch port*. This scenario does not provide end-to-end security when used for access control [3].

---

[3] In principle layer-two MACSec security associations could be created between any given set of peers on

While GNU/Linux and some other operating systems do support MACsec and 802.1x, it is uncommon for consumer-grade switching equipment to support it and when enterprise switching equipment offers support it typically requires a paid license; these issues hinder general adoption. It may also be subject to export control, especially when specialized hardware is required for a given platform deployment of MACsec. MACSec is typically not end-to-end encrypted but host to switch-port and usually combined with 802.1x authentication. In such a setup traffic after the entry-switch is often completely unprotected.

Switching infrastructure with MACsec generally has access to encrypted and unencrypted Ethernet frames while a passive surveillance adversary is generally only able to intercept Ethernet frames through Ethernet cable tapping or by monitoring radio emissions for a related wireless network. An active adversary may compromise the wireless drivers of a client [APR+13a], an access point, switch, and/or router to gain access to key material. In the general case, end-to-end encryption is a much stronger and much more desirable protection than the partial protection offered by 802.1x networks even when deployed in tandem with MACsec.

Post-Quantum MACsec Key Agreement for Ethernet Networks [CS20] suffers from the same problems as MACsec in that it is not an end-to-end protocol, it is layer-two, and at this time it is an experimental protocol [Rep18] which has not been adopted by any MACsec vendors into the TCB [Kam20] of network equipment.

Wireless network security protocols attempt to tackle confidentiality, integrity, and access control in a manner which is generally not secure against surveillance adversaries as shown below. We consider the WPA1 and WPA2 personal protocols which are commonly used as they require no additional authentication servers or configuration beyond setting a passphrase. Long term monitoring of passphrase authenticated wireless networks with poor passphrase rotation policies is especially problematic. Given a password, a passive adversary is able to recover plaintext for each session for which they have recorded a successful authentication and association, in addition to the encrypted traffic that they wish to decrypt. Handshakes occur frequently, e.g. devices that enter a low power mode generally re-authenticate after waking from a sleep mode, so that adversaries arriving too late need not wait long. Many wireless networks do not support protected management frames [IEE09] and so adversaries commonly are able to force a disassociation without knowledge of the passphrase. Users' software will commonly automatically reconnect after an adversary has forced a disassociation. This extremely common issue gives an adversary the chance to force and then observe a fresh handshake and thus mount the above-mentioned attack. Furthermore, a recording of the handshake permits mounting offline password-guessing attacks.

With Vula, these attacks are mitigated with regard to traffic confidentiality concerns. For example, when Vula is deployed for users on a WiFi network, an attacker breaking the WPA/WPA-2 access controls and thus joining the network is restricted to performing only denial-of-service attacks instead of being able to mount a full on-path active Machine-In-The-Middle (MITM) attack with access to unencrypted IP packets.

7.2.2 – Comparison with other projects. There are a wide variety of tools which can be used to create end-to-end encrypted tunnels between hosts, or which share other su-

---

the same Ethernet segment. It would require additional research and development to create an equivalent layer-two protocol with a cryptographic handshake equivalent to WireGuard.

	0	1	2	3	4	5	6	7	8
Tailscale [Tai20]	✗	to specific IP addresses	✗	coordination server	✗	✗	✓	✓(client) + ✗(server)	WireGuard
Headscale [Fon20]	✗	to specific IP addresses	✗	coordination server	✗	✗	✓	✓	WireGuard
innernet [Ton21]	✗	to specific IP addresses	✗	coordination server	✗	✗	✓	✓	WireGuard
Nebula [Sla19]	✗	to specific IP addresses	✗	certificate authority	✗	✗	✓	✓	Custom protocol
MACsec [IEE06]	✗	Ethernet link w/host and switch	✓	RADIUS server	✗	✓	✗	✓(client) + ✗(switch)	MACsec
tcpcrypt [BGH+19]	✓	TCP traffic w/participating hosts	✓	none	✗	✓	✗	✓	tcpcrypt
IPsec OE [Wou13]	✗	w/participating hosts (LAN & WAN)	✓	DNS+DNSSEC and/or CA	✗	¡	✗	✓	IPsec cipher-suite
Vula	✓	w/participating hosts (LAN)	✓	none	†	✓	✓	✓	WireGuard

Table 7.1: **Comparison of properties:** ✗: no, ✓: yes, ¡: not default, †: transitional, 0: zero configuration, 1: encrypts, 2: works offline, 3: required infrastructure, 4: post-quantum, 5: protects traffic using existing IPs, 6: secure hostnames, 7: free software, 8: encrypted transport.

perficial similarities with Vula. To our knowledge, however, none of them achieve Vula's design goal of providing fully-automatic end-to-end encryption of local area network traffic. We present a comparison in Table 7.1.

Projects such as Tailscale [Tai20], Headscale [Fon20], and innernet [Ton21] are similar to Vula in that they can be used to encrypt traffic between hosts on a LAN using WireGuard tunnels, but they differ in some important respects: They only create tunnels between hosts that are logged in to the same account on a centralized coordination server. Tailscale outsources the operation of this component to Amazon, a surveillance actor. Headscale and innernet provide free software implementations which can be self-hosted, but the server remains a single point of failure. These systems use a different IP range inside and outside of the tunnels, so LAN-based applications need to be reconfigured to benefit from it. They do not provide any post-quantum protection. Furthermore, Tailscale requires Internet access, thus is unsuitable for offline, or airgapped networks. Tailscale also requires an additional trust relationship with at least one but likely more 3rd parties: Tailscale and one of Google, Amazon, Microsoft, or an email provider. Nebula uses a custom protocol that its authors claim is based on a Noise Protocol Framework [Per18] handshake and it has yet to receive the scrutiny of other instantiations such as WireGuard [Don17a]. Nebula, like Tailscale, is used to construct a similar organizational structure VPN mesh. Tailscale, Headscale, innernet, and Nebula are unsuitable for dynamically discovered peers, air-gapped network segments, and/or multi-organization protection, and these properties are not goals of the respective projects.

With the exception of tcpcrypt and IPsec OE, the other projects listed in Table 7.1 are all designed to protect traffic between hosts which are configured to be part of a single organization, whereas Vula provides automatic encryption of traffic between *all* locally-reachable hosts that are running the software. tcpcrypt is an outlier, in that it does provide opportunistic encryption between hosts without any configuration; however, it only protects TCP traffic, does not provide secure names, its key verification system requires application-specific support, and it appears to be an out-of-tree Linux kernel patch. These and other deployment impediments have prevented its adoption even after standardization [BGH+19]. For these reasons, we find tcpcrypt unsuitable for Vula's needs but we remark it is still an interesting design with important goals. IPsec OE is designed to provide opportunistic encryption, but has numerous [Wou13] shortcomings [Gil13], including vulnerability to quantum computers, and it has failed to gain adoption, partially because it requires manual configuration.

7.2.3 – Star network WireGuard deployments. While WireGuard's architecture is defined in terms of *peers*, deployments often use a hub-and-spoke network topology wherein multiple hosts which are commonly referred to as *clients* connect to one or more centralized hosts which are commonly referred to as *servers*. While this topology can be used with WireGuard or another VPN in the context of a local Ethernet segment, it presents a number of downsides related to the server(s) being SPOFs. Bandwidth SPOF: The total bandwidth available for clients to communicate with each other is limited to the bandwidth of the server(s) they are communicating through. When that bandwidth is exhausted, performance suffers for all clients. Confidentiality SPOF: Due to the absence of end-to-end encryption, a central server holds *excess authority* allowing it to capture and/or modify traffic from many clients which are routing through it. Availability SPOF: The ability of clients to communicate with each other is entirely dependent upon the availability of their centralized servers.

7.2.4 – Point-to-point VPN deployments. It is possible to manually configure WireGuard or another VPN in a point-to-point topology on a LAN to achieve some of the same properties that Vula provides. However, there are some shortcomings to a manual approach which also apply to the star network topology. Vula addresses these issues.

For example, Vula automatically securely computes and sets the pre-shared key (PSK) value in the WireGuard protocol for all peers. The use of an additional PSK is to add transitional post-quantum security to the WireGuard protocol, and Vula removes the need for manual configuration. We use CSIDH [CLM+18a] as described in Section 7.4.5 to compute shared symmetric keys between pairs of peers and the result is used as a PSK. Vula explicitly supports rotation of the CSIDH keypairs on a regular basis as this rotates the PSK shared between peers.

Management. Adding new hosts to a point-to-point WireGuard overlay network requires configuring each existing host with the new host's key and IP address, and configuring the new host with all existing hosts' keys and IPs. Vula performs this key distribution and routing configuration automatically. We explain the use of multicast in Section 7.4.2.

We make a distinction between Vula the protocol, Vula the Python implementation, and the `vula` device provided by the operating system. Vula the Python implementation uses a network interface called `vula`. The `vula` network interface is a standard WireGuard device with a custom string for a name rather than the typical `wg0` name. On some operating systems the device name is not customizable and for the sake of clarity, we call the WireGuard device configured by Vula the `vula` network interface or the `vula` device.

Addressing. In most point-to-point WireGuard configurations, the IP subnet used for VPN traffic is separate from the one used for other traffic. This means that traffic to and from typical LAN applications using mDNS [CK13b] hostnames will not be automatically encrypted without additional configuration of each application. Vula, in contrast, encrypts all connections between participating peers while applications continue using their existing LAN IP addresses and hostnames.

7.3 — Threat Model and design considerations

In the examples below, we require that Vula users have at least a single IPv4 address, and are connected to an IP network through an Ethernet switching fabric and/or a wireless LAN. To optionally protect upstream traffic, we additionally assume that any hypothetical user is on a LAN which has at least one IPv4 gateway with connectivity to the wider Internet. We choose the strongest adversaries to defend against, thus assume that the adversaries may record all IP packets. This would happen for unprotected WiFi, if an adversary has access to a switch mirroring port, or if the adversary has any WPA/WPA2 passphrases.

7.3.1 – Unilateral Surveillance Adversary. The possibility of **Unilateral Surveillance**, such as the wideband monitoring of all wireless networks in an area, is a well-understood attack vector. With commonly deployed consumer or so-called *prosumer* [RJ10] [Kot10] equipment, capture of association handshakes with a wireless access point will allow an attacker to guess a passphrase and later decrypt captured wireless traffic. Interception of wireless networks is so common that there are cloud-based services [Mar12] as well as GPU optimized key recovery tools [AKSE18] for attacking (WPA) cryptographic handshakes in service of decrypting intercepted data.

A simple and relatable example is a curious neighbor who lives in close physical proximity to a wireless network such that their basic interception equipment is within radio range. They may passively capture wireless traffic over long periods of time, decrypt it at a later date, and refrain from joining the wireless network lest their subterfuge be detected. An example of a tool that may be used by such a neighbor is the SPARROW II as seen in subsection 4.54. Their home devices may otherwise be compromised [AGG⁺14e] and used by an adversary.

Another passive example is the NSA program OVERHEAD [Gal16] which performs wireless network packet capture *in space using satellites* [Ryg16]; data from that program may be fed into systems such as XKeyscore [AGG⁺14a].

7.3.2 – End User. There are a variety of ways that **End Users** can attack each other on a Local Area Network, due to the reliance on vulnerable protocols such as the Dynamic Host Configuration Protocol (DHCP) [Dro97], the Address Resolution Protocol (ARP) [Plu82], and the Domain Name System (DNS) [Moc87]. By simply sending a few malicious ARP or DHCP packets, an end user can easily intercept other users' traffic on a switched network. Users may also attempt to use packet-in-packet [GBM⁺11] smuggling to interfere with other users.

If an End User on the network segment wants to pretend to be the canonical DHCP server for the network segment, they may be able to move a user to a completely different network segment. We call this the *DHCP attack*, though it is really many possible variations of different attacks which involve DHCP. This may include spoofing the real DHCP server, selectively filtering DHCP messages between client and server through ARP spoofing at layer two, or simply by exploiting race conditions present in how common operating systems join networks. Simply partitioning the network and offering a completely different DHCP server for that segment of the network may be enough. It is for this reason that we strongly encourage that users use manual IP addressing and set a static ARP entry for systems which are largely static in nature such as infrastructure that does not regularly relocate. For laptops, we recognize that using DHCP is a way of life for users, and we encourage those users to verify Vula peers manually, so that those peers are always present even when moving between networks.

An additional adversary to consider would be an active adversary using NIGHTSTAND as shown in Figure 4.53 which is an attack suite to compromise wireless (802.11) devices. Targeted devices may be the target themselves, or they may be connected to other networks where compromising the target allows for access to previously unreachable systems. NIGHTSTAND exploits device driver implementation bugs in the target wireless device drivers. Once a system is compromised, it may be used to intercept traffic that it sees from its vantage point. If the exploited device is connected to other systems such as a larger LAN, an attacker may use the targeted system for further lateral movement through the network.

7.3.3 – Network Operator. We presume that the **Network Operator** is able to enable port mirroring for an entire switching fabric. This means that they are able to passively collect every packet sent within the switching fabric, as well as enabling full packet capture on an upstream router which sends all user data to and from the Internet. We consider this adversary to be close to the Dolev-Yao [DY83] model for an attacker, in that they are able to arbitrarily disable user access, change passwords, capture packets, inject packets, delay delivery, and more. Usually this is only possible in the upstream equipment providing network access to the Internet. They are able to carry out all of the other attacks enumerated. In an ideal environment, at least one end user is actually the **Network Operator**. So while all powerful, we presume that a user will not attack themselves but rather consider what is possible if their own equipment is compromised [GWEA18a].

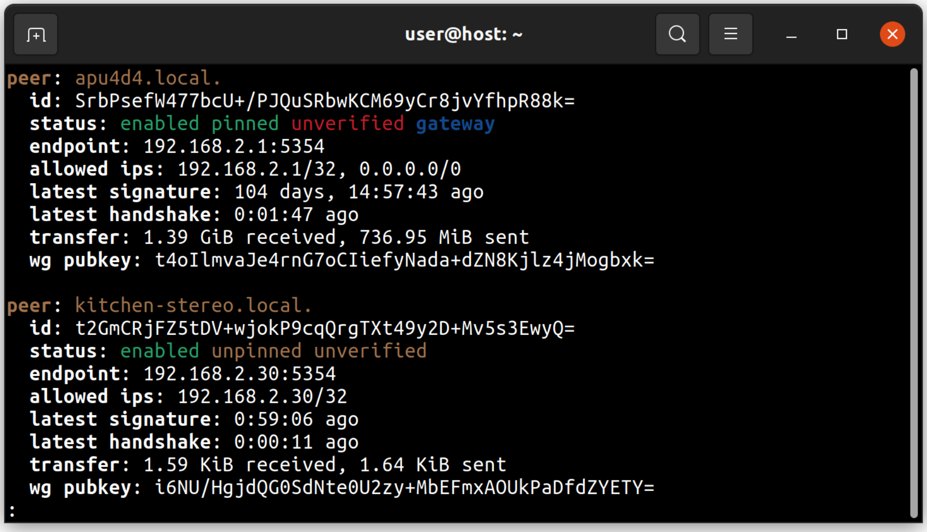
7.3.4 – Vula peer states. We distinguish peers by their *verification state* and their *pinned state*. By default, peers are either *unverified* and *unpinned*, or *unverified* and *pinned*. Unpinned should also be thought of as *replaceable* by another Vula peer, and pinned should be thought of as *permanent* where no other Vula peer may conflict with the peer's claimed resources.

In Vula each user has a cryptographic identity given by a long-term Ed25519 [BDL+11] key. These keys certify all other cryptographic keys used in the Vula protocol. Verification is an out-of-band process whereby Vula users compare these Ed25519 identity keys. An example of two peers and their verification state is presented in Figure 7.1. The identity keys are the only keys that do not change while all other cryptographic keys may be regularly rotated. The Ed25519 public key for signatures is known as the *verification key* (vk) and it is used to sign all Vula *descriptors* – either broadcast to other peers over the network or scanned as part of an optional QR code verification process. Descriptors can also be smuggled through other protocols; this is left as an exercise for advanced users.

In order to also protect against active attackers, continuity of vk public keys must be enforced with respect to both hostnames and IP addresses. This leads to a security-convenience trade-off: if continuity of vk public keys is enforced by default for all peers, naturally-occurring name or IP conflicts will sometimes lead to an inability to communicate. For this reason, we introduce a user-controlled boolean state for each peer called *pinned*. Continuity of vk public keys is only enforced for peers in the *pinned* state. Pinning peers allows users to have the benefit of protection against active adversaries at the cost of needing to manually resolve hostname or IP address conflicts.

Pinning a peer creates a binding from the peer's long-term *verification key* to lists of hostnames and IP addresses which that peer has been known to use. A single pinned peer may be associated with any number of hostnames and IP addresses, while a given hostname or IP address may never be associated with more than one peer.

A pinned peer is a permanent peer. A pinned peer has a permanent route and traffic



```
peer: apu4d4.local.
  id: SrbPsefW477bcU+/PJQuSRbwKCM69yCr8jvYfhpR88k=
  status: enabled pinned unverified gateway
  endpoint: 192.168.2.1:5354
  allowed ips: 192.168.2.1/32, 0.0.0.0/0
  latest signature: 104 days, 14:57:43 ago
  latest handshake: 0:01:47 ago
  transfer: 1.39 GiB received, 736.95 MiB sent
  wg pubkey: t4oIlmvaJe4rnG7oCIiefyNada+dZN8Kjlz4jMogbxk=

peer: kitchen-stereo.local.
  id: t2GmCRjFZ5tDV+wjokP9cqQrgTXt49y2D+Mv5s3EwyQ=
  status: enabled unpinned unverified
  endpoint: 192.168.2.30:5354
  allowed ips: 192.168.2.30/32
  latest signature: 0:59:06 ago
  latest handshake: 0:00:11 ago
  transfer: 1.59 KiB received, 1.64 KiB sent
  wg pubkey: i6NU/HgjdQG0SdNte0U2zy+MbEFmxAOUkPaDfdZYETY=
:
```

Figure 7.1: Vula peers as shown by `vula peer -show`

for that peer is always directed into the local `vula` network interface; pinned peers do not expire. If no WireGuard session exists between the user and their respective peer, the traffic is never emitted onto the network as the device will *fail closed* (See [RAKF12, Section 4.1] for the definition). Pinned peers cause a denial of service with non-participants or colliding Vula peers by design if other users obtain the same IP address or use the same host name; this is independent of those being Vula peers or non-participants. Consider the following scenario: Laura uses Vula and has pinned Glenn; she is hostname.local with 10.0.0.2 as her IP address. Glenn uses Vula; he is at otherhostname.local with 10.0.0.3 as his IP address. Glenn leaves the network. Carol arrives on the network. Carol does not use Vula. The DHCP server gives Carol 10.0.0.3 as their IP address. Laura knows that only Glenn is available at 10.0.0.3, but Glenn's WireGuard does not reply, so Laura cannot talk to Carol and any attempt at communication will fail closed. Unencrypted traffic will not leak out of the WireGuard tunnel. When Glenn returns, and the IP address is still in use he will obtain a new IP address which Laura will learn through a Vula broadcast. Laura will see that Glenn now has at least two IP addresses, and Laura will still be unable to reach Carol until Carol obtains an IP address not used by a pinned Vula peer. Carol is oblivious to all of this.

An unpinned peer is a temporary peer. Unpinned peers remain until the user's system moves to another network segment, until the peer descriptor expires, or until a new peer announces resources that conflict with this replaceable peer. To securely reach the peer, Vula adds specific routes for peer addresses to the `vula` device, and Vula additionally configures the same addresses as being associated with the cryptographic keys for the peer on the `vula` device. When the peer is replaced or expires, the routes are removed, and the cryptographic keys are removed from the `vula` device.

Often participants and non-participants are mixed on private network segments that use commonly allocated private [MKR+96] IP addresses. To prevent denial of service for potentially communicating hosts, unpinned peers *fail open* for the benefit of non-participant hosts.

7.3.5 – Cryptographic choices. A protocol is said to have *perfect forward secrecy* (PFS) [MvOV96] if compromise of long-term keys does not compromise past session keys, Vula brings this property to IP traffic for participating systems. From the perspective of public-key cryptography, the attack targets in Vula are reducible to a few specific problems. An attacker wishing to forge Vula peer descriptors must be able to forge Ed25519 signatures to break authenticity of the peer discovery and key exchange mechanism. If the authentication process is not broken, the attacker wishing to recover plaintext traffic must record traffic, and then they must break X25519 [Ber06] as used in WireGuard, and CSIDH-512 to recover the PSK.

7.3.6 – Automatic protection against passive adversaries. As Vula automatically encrypts traffic between hosts while they are connected to the same IP multicast domain, in the absence of an active attacker, it will always deny passive adversaries the opportunity to decrypt traffic that they capture.

7.3.7 – Automatic protection against active adversaries. Encryption relying on an unauthenticated key exchange is, of course, intrinsically vulnerable to key-substitution attacks by active adversaries who are present at the time of the initial key exchange. The concept of authentication, however, is meaningless in the absence of a notion of identity. In the LAN setting in which Vula operates, there are several notions of identity, such as hostnames, IP addresses, and MAC addresses, but none of these are intrinsically authenticatable. Therefore, without manual key verification or dependence on some sort of public key infrastructure, it is not possible to automatically authenticate the initial communication between two hosts on a LAN. However, it is possible to automatically provide protection against active adversaries who only become active after that point, by following the trust-on-first-use [WAP08] (TOFU) pattern often employed by users of SSH: Keys are implicitly assumed to be valid for hosts which have never been contacted before, and continuity of vk public keys is enforced for any subsequent communication. Unlike SSH, where users are prompted to explicitly make the TOFU decision, Vula has a configuration option called *pin_new_peers* which causes newly-discovered peers to be automatically marked as *pinned*. This is not the recommended default as it imposes user interface awareness requirements on users as explained in subsection 7.6.4 and shown in Figure 7.2.

Peers automatically *pinned* in the *pin_new_peers* state are vulnerable to an active attack *only at the time that they discover peers for the first time*. If their initial peer discovery was not compromised, Vula protects them against active attacks at any later time.

For full protection against active attackers, including those who could be present at the time of first contact, manual key verification is necessary. When a peer is manually verified, it is marked as *pinned* and is also marked as *verified* to allow the user to distinguish it from peers pinned automatically by the *pin_new_peers* state.

Vula provides a convenient-to use QR code-based tool for performing peer verification. We describe this verification process later in the description of the Vula protocol, specifically as an optional phase as explained of item 6 of Section 7.4.4. To protect against active adversaries who are present at the time of initial contact, it is necessary to manually verify fingerprints.

7.3.8 – Security-convenience trade-off. We consider the default behavior for Vula protocol implementations with regard to the usability and security outcomes.

pin_new_peers = true. As stated above, using the *pin_new_peers* mode has the advantage that unverified peers for whom the initial contact was not compromised are automatically protected against any subsequent active attacks. The disadvantage is that when an IP address which has been previously used by a Vula peer is later reassigned to a new host, Vula users who learned about the previously associated IP and are using *pin_new_peers* as their default mode will be unable to communicate with the new host, regardless of whether it runs Vula itself, until they explicitly remove the IP address as associated with the previously existing public key or the new host moves to a previously unassigned IP address. Pinned peers accumulate IP addresses and hostnames where manual removal may be necessary.

pin_new_peers = false. Marking new peers *unpinned* by default has the disadvantage that new peers will remain vulnerable to active attacks until they are explicitly marked as *pinned* or *verified*. It has the advantage that it will gracefully handle IP address reassignment and/or hostname collisions without requiring any user interaction, so Vula could conceivably be widely-deployed and enabled by default without causing significant inconvenience while also thwarting passive adversaries. Unpinned peers do not accumulate IP addresses and hostnames, they are replaced by any conflicting announcements, and they expire automatically.

7.3.9 – Summary of protections. Vula should always provide confidentiality with respect to passive adversaries. For peers that are *pinned*, it will also protect against active adversaries as long as those did not compromise the first contact. For peers that are manually verified, a successful verification ensures security against attackers which were active even at the time of the first contact as any key-substitution attack would make manual verification fail.

Although Vula protects the confidentiality of network traffic between verified peers against both passive and active attackers, we do not claim to be able to prevent all traffic analysis attacks which may be revealing. We also do not attempt to prevent packet delaying or Denial-of-Service attacks, and we admittedly do allow for some new minor avenues by which DoS attacks can potentially be executed as explained in subsection 7.6.4. However, these are not significantly different from the DoS vulnerabilities which are inherent in the LAN setting where DHCP and ARP are used. To reduce Denial-of-Service attacks against Vula, we note that a manually configured IP address removes dependence on the insecure DHCP protocol for IP address configuration and similarly setting static ARP entries for hosts removes dependence on the insecure ARP protocol for IP address to hardware Ethernet address resolution. However, we observe that future research is needed for securing both protocols. DHCP and ARP both need to be enhanced with cryptography. Vula cannot eliminate cross protocol attacks or make lower level protocols secure, it does however reduce the security issues of both protocols to a Denial-of-Service.

7.4 — Detailed Protocol Description

The Vula protocol does not rely on any infrastructure and is purely a peer-to-peer protocol. Every participant that wishes to use the protocol must install the Vula software on the computer system which has traffic it wishes to protect with the Vula protocol. As a concrete example, a router running the Vula implementation is able to provide a locally secured WireGuard tunnel to any downstream clients who also run Vula. Downstream clients may then communicate through the router to the Internet with all traffic protected between their respective systems and the router itself. If the router itself does not have a Vula peer upstream or another VPN tunnel, the traffic will be unencrypted as it traverses subsequent routers. The benefit of running this software on a router is that normally regular clients may intercept each other's traffic with minimal effort as explained in Section 7.6.4, and with Vula, they would need to violate some assumption of the protocol which is protected by strong cryptography.

7.4.1 – WireGuard. The Vula protocol relies on a secure tunneling protocol for protecting IP packets between participating systems. We have selected WireGuard [Don17a] as our encrypted tunneling protocol on the basis that it is well understood, peer-reviewed, extremely efficient, performs exceptionally fast packet transformation even under heavy system load, and is now a part of the Linux kernel shipping with a number of GNU/Linux distributions. Unlike IPsec, it is not suspected of being sabotaged by the NSA. IP traffic between any given pair of hosts participating in the Vula protocol is protected by WireGuard. WireGuard is modeled after the Noise Framework IK pattern [Per18, Section 7.5] which in turn has been updated to reflect some of the needs of WireGuard. The IK pattern optionally allows any pair of peers to use a symmetric pre-shared key (PSK) to make the WireGuard protocol transitionally post-quantum in addition to keys derived from both ephemeral and long term keys. We take advantage of this and generate a pair-wise shared secret with CSIDH. To a third party observer, the use of a PSK is indistinguishable from other WireGuard traffic which does not use a PSK. WireGuard presents an interesting constraint: the WireGuard user must configure WireGuard with a peer's public key before WireGuard can begin securely communicating with that peer. This raises a number of questions about efficient key exchange, as well as questions about rotation of keys used in the protocol. Session keys rotate every few minutes under normal usage conditions, though long term keys must be rotated manually. WireGuard leaves discovery of peer public keys, as well as configuration, as a problem for the user to solve. Vula automates everything that WireGuard has left for users to otherwise manually configure.

7.4.2 – mDNS/DNS-SD: decentralized Vula peer discovery. Each user's Vula descriptor contains their long term WireGuard public key, along with their CSIDH public key. We have chosen to automatically distribute Vula descriptors using multicast DNS (mDNS [CK13b]) and DNS Service Discovery (DNS-SD [CK13a, Cro19]), on the local network segment. DNS-SD specifies structure for DNS records which may be used to facilitate service discovery using DNS. When mDNS and DNS-SD are combined together, DNS queries for hosts under *.local.* should not leave the local network segment to be properly resolved. Each Vula peer must publish a *Service Name* under *_opabinia._udp.local.* [4] for their host to the local network. A query for the respective Service Name should return a TXT record containing a list of values representing a Vula *descriptor*. The values required for Vula are enumerated and briefly explained in Table 7.2.

---

[4] See Opabinia Regalis [Wik21m] from the Middle Cambrian.

key	example value	description
addrs	192.168.6.9	List of addresses for Vula peer
c	36e8...c764	CSIDH public key for deriving pair-wise PSKs
dt	86400	Seconds after vf that descriptor is valid
e	0	Flag indicating that a peer is ephemeral
hostname	seashepherd.local	Hostname
pk	cW8Ek...R0=	WireGuard Curve25519 public key
port	5354	WireGuard UDP port number
r	1	IP forwarding services are available to peers
vf	1601388653	Starting validity of descriptor in seconds since 1970
vk	ptKKc...0M=	Ed25519 public key used to sign Vula descriptor
s	adsLEe...a=	Ed25519 signature over Vula descriptor, except s

Table 7.2: mDNS/DNS-SD Vula descriptor key, value examples

We presume that the generally insecure nature [GWE+15] of DNS, even in the local LAN context with mDNS and DNS-SD, is understood. As an alternative to DNSSEC [RLM+05] or other proposals [GWEA18a], Vula enhances the security of DNS-SD service records with cryptographic signing of the service descriptors. All computers which wish to deploy Vula must be able to send IP packets to and receive from **224.0.0.251:5353** or **[FF02::FB]:5353** with the correct multicast MAC addresses for any IP packet they send or receive respectively. These packets should only contain properly formatted mDNS queries or answers.

When a peer publishes its descriptor, all of the values besides the signature s but including the verification key vk are ordered and serialized into a string. A signature over the string is computed and its value is stored in the final item, s; the resulting list of name-value pairs is then added to the DNS-SD Service record.

The verification key (vk) is used for authenticating Vula protocol messages; currently the messages are used for peer discovery and peer consistency, stateless rotation of IP addresses such as when a DHCP server gives a DHCP client a new IP address, for rotation of the CSIDH public key and derived PSKs, and for rotation of the WireGuard Curve25519 public key used by the `vula` device. This device appears as a normal network interface with the name *vula* in various system configuration tools.

7.4.3 – Vula Protocol logic. In this subsection of the paper, we walk step-by-step through a full protocol run for two peers on the same LAN segment.

The same protocol scales to n possible peers. One peer queries for the DNS-SD service and n devices may answer. The only limit to the number of peers is the number of addresses on the local segment, any internal limit on peers that the WireGuard implementation may impose on configuring the network interface, and on system memory.

When a peer receives a new descriptor, it evaluates it according to a policy engine which considers the peer's current state and enforces various constraints. We define the policy engine as a pure function which computes the next policy state from the current state and some event: `ProcessEvent(PolicyEngineState, Event)` $\Longrightarrow$

`PolicyEngineState'`. Events include incoming descriptors, changes in the system state such as an IP address being configured or unconfigured by a DHCP client, or some user action. The `Event` objects contain a timestamp indicating when the `Event` occurred, which allows `ProcessEvent()` to make decisions which include time despite being a pure function. A flowchart showing an overview of the policy engine's handling of the *incoming descriptor* event is shown in Figure 7.2.

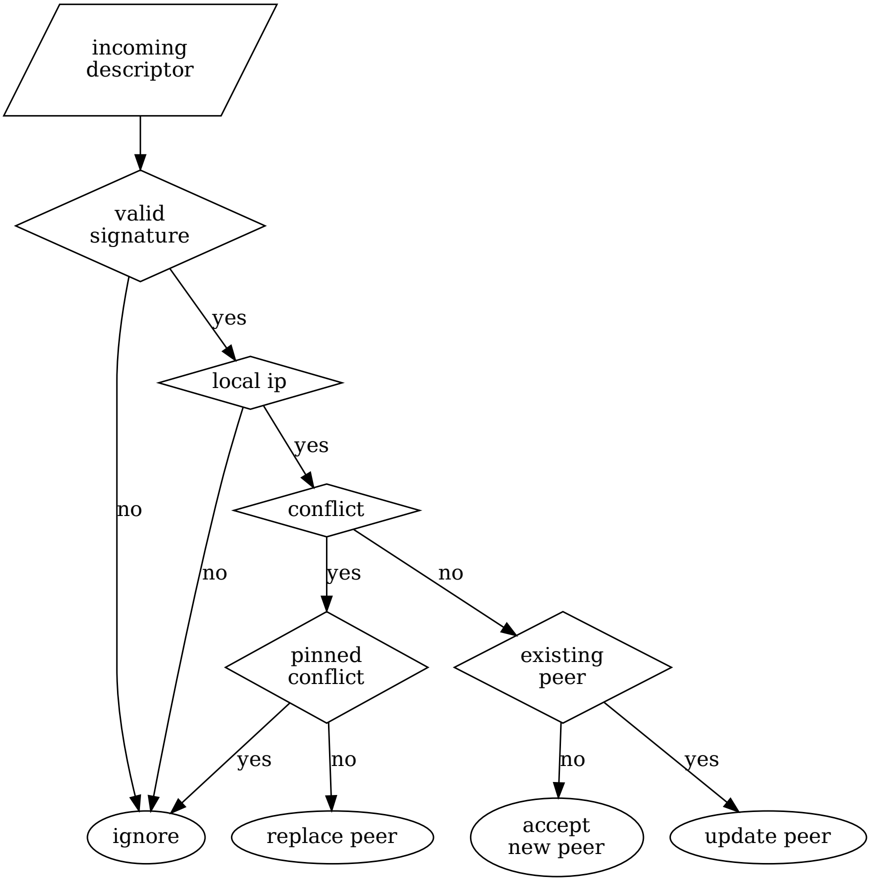

incoming descriptor
valid signature
yes
no
local ip
yes
no
conflict
yes
no
pinned conflict
existing peer
yes
no
no
yes
ignore
replace peer
accept new peer
update peer

Figure 7.2: Incoming descriptor processing state engine

If the *vk* in a descriptor corresponds to a known peer and the descriptor is different from the latest descriptor previously seen from that peer, then the peer state is updated to reflect the new descriptor; otherwise, a new peer entry is created. The peer state includes a list of all hostnames and IP addresses which each peer has ever announced, along with an *enabled* flag for each. Newly announced IP addresses are marked as *enabled* only if they are both in an acceptable subnet and they do not collide with any *pinned* peers' enabled IPs. Acceptable subnets are those which are both on the list of *allowed_subnets* and where the evaluating peer also has an IP bound at the time that the descriptor is first seen. Hostnames are likewise protected against collisions, and accepted only if they end with an allowed suffix on the *local_domains* list which by default is set to *.local*.

Collisions with unpinned peers' hostnames and IPs are governed by the *overwrite_unpinned* policy option; if it is set, then unpinned peers can have their hostnames and IPs immediately disabled and reassigned to newly discovered peers. Unpinned peers are automatically removed from the database when they have not made an announcement in at least *expire_time* seconds, which defaults to 3600. Descriptors must have a valid from (*vf*) value that is smaller than the current time, and for already-known peers the value must be greater than the previous descriptor from that peer. If the descriptor sets the IP router flag, and the receiving peer processing it sees that the announced IP address matches the current default route, and the *accept_default_route* policy option is enabled, then the peer's *use_as_gateway* flag is set, which will cause *vula organize* to configure the remote peer as the receiving system's default route and adjust the peer's AllowedIPs value accordingly.

7.4.4 – Protocol steps. The Vula protocol requires that a Vula implementation will regularly repeat the query and response phases. The Vula implementation currently uses NetLink events to know when the network has possibly changed, and to ensure new descriptors are being regularly created, sent, received, and processed.

0. Phase 0: Laura and Glenn both start with three keypairs each. The Curve25519 keypair is used for the WireGuard peer identity for the Vula device, the CSIDH keypair is used to establish PSKs with other peers, and the Ed25519 keypair is used for signing Vula protocol messages.
1. Phase 1: Laura creates a protocol message $\aleph$ which contains a cryptographic signature over the contained list of values as shown in Table 7.2.
2. Phase 2: The $\aleph$ value is used to construct a TXT record for the mDNS DNS-SD service associated with Laura's hostname and all records (A, SRV, and TXT) are published by Laura upon request.
3. Phase 3: Glenn sends a query to the multicast address for the network and queries for the Vula service *_opabinia._udp.local*.
4. Phase 4: Glenn receives Laura's $\aleph$ protocol message and any other messages of participating hosts for *_opabinia._udp.local*.
   Glenn verifies the $\aleph$ descriptor is properly formatted and that the signature is valid. Glenn will then process the $\aleph$ protocol message according to a set of constraints. Any failure to meet the constraints as enumerated in Section 7.4.3 will result in rejecting the Vula descriptor $\aleph$.
5. Phase 5: If the descriptor is not rejected, the Vula device will also be reconfigured and system routes added as needed. If there was a new IP address announced for an existing peer, it will become the new endpoint for the peer. Traffic to the new IP address will now be routed via the Vula device. If the peer is in the *pinned* state, traffic to its previous address or addresses will also continue to be routed via the Vula device.
6. Phase 6, optionally verify peers: The final step of the protocol is optional and highly recommended. To complete this phase of the protocol, the end-user may verify a

peer's *vk* either manually or with a convenient-to-use QR code. When the user has verified a peer, the peer's state is mutated to reflect that the peer is now both *pinned* and *verified*.

7.4.5 – Implementation. We have implemented Vula in Python 3 for GNU/Linux. All of our software and changes to related software are Free Software, and are available at `https://vula.link/`.

Vula is separated into three distinct services: a publishing daemon, a discovery daemon, and a configuration daemon. We have implemented each of these daemons to have minimal attack surface. Each participating host must run all of the services listed here to properly use the Vula protocol with other hosts on their LAN segment.

Our initial implementation of Vula made use of three different CSIDH implementations depending on the platform where it would be used. We first started with the reference implementation [CLM+18b], it is a generic C program which runs on systems with little-endian architecture and word size of 64 bits. We found it lacking in portability and in side-channel protections which is to be expected for a proof of concept implementation. The x86_64 implementation [CCC+19] claims to be constant-time and extremely fast, while the ARM64 [JAKJ19] implementation is constant-time and extremely slow computing key derivation. The performance of each C implementation relied on specific CPU features which made portability extremely difficult. Additionally, each implementation had its own serialization formats which were incompatible. We later adopted a pure Python CSIDH implementation [ACDR21] [ACDR20] which is constant-time. While significantly slower than the reference or other implementations in C as mentioned in subsection 7.4.9, the Python implementation allowed for supporting any CPU architecture where Python is available with a single implementation. Python does not require separate builds for each of our systems' CPUs (RISC-V, AMD64, POWER9, ARM64, ARM32) and it provides memory safety. All available CSIDH implementations in C use unportable Intel or ARM assembly. Vula's handshake is not performance-sensitive and key derivation is cached for previously seen public keys. We additionally implemented serialization formats for CSIDH keypairs which should allow for greater interoperability. Bulk encryption of IP packets is handled efficiently by in-kernel WireGuard.

Slow key derivation may be a denial of service vector for embedded devices which decide to deploy a constant-time implementation over the reference implementation. We have extended each of the previously mentioned CSIDH implementations to include a basic tool for key generation and key derivation as well as shared secret generation. These tools are not currently used as Vula has chosen portability over performance at this time. After a shared secret is generated, we use a standard HKDF [KE10a] construction to hash the secret value before use in any cryptographic context.

7.4.6 – Multi-daemon systemd integration or monolithic mode. The Vula implementation operates by default in multi-daemon mode with `vula organize`, `vula discover`, and `vula publish` daemons. Multi-daemon mode includes systemd configuration files to run as several systemd services at install time. Each of the daemon services is run as a systemd service with minimal privileges, i.e.: as an unprivileged user which has minimal access to the overall system. The services are grouped in a systemd slice called `vula.slice`. Each daemon follows the principle of least authority: each service has the minimum set of capabilities and permissions required to accomplish the specific tasks of the daemon. Further details about Vula systemd integration are available in subsection 7.4.12.

For systems that do not support systemd or for systems where only a single daemon is desired, the Vula implementation can also run all required services as a single process. The monolithic mode combines the `vula organize`, `vula discover`, and `vula publish` daemons into a single daemon, `vula`, which retains the superset of all other required daemon privileges which are normally compartmentalized away.

7.4.7 – Vula peer tunnel considerations. During the `vula organize` daemon startup the local peer cache is loaded before new configuration information is accepted from the `discovery` daemon. After configuration of previously known peers, the `organize` daemon sits idle until a new descriptor is sent by the `discover` daemon or until another network event changes the system state. Key changes, IP address information, route updates, and interprocess communication from the command line interface are handled by this daemon.

The `vula` device is a normal WireGuard network interface which is entirely managed by the Vula `organize` process. This device has a single long-term identity which corresponds to the Curve25519 public key in the Vula descriptor announcements. Unlike normal usage of WireGuard, this key may be rotated at any time as long as the newly generated public key is announced to the local network or the descriptor is otherwise shared with Vula peers. WireGuard peers on the `vula` device always have a pre-shared key set. This key is derived from the CSIDH public key of the peer, and the CSIDH private key of the device owner. This key may also be rotated at any time as long as the new public key is also announced to the local network.

IP packet marking. IP packet marking is required to ensure that unencrypted packets are encrypted by the `vula` WireGuard device when appropriate as well as to mitigate routing loops of already encrypted packets. An important corner case with any point-to-point tunnel is to guarantee that packets which should be encrypted are encrypted. When a failure to encrypt happens and an unencrypted packet is sent over a device other than the VPN device, it is generally called a *bypass* or a *leak*. IP packet marking allows Vula to use WireGuard in a way that prevents this class of catastrophic failures that are common with point-to-point VPN software. Other VPN software that does not use IP packet marking suffers from catastrophic traffic bypass issues [RAKF12] which may be exploited by an adversary. One example where a bypass may occur is that WireGuard devices are configured with a peer at a given endpoint IP address, UDP port, and a list of AllowedIPs. Without IP packet marking, the endpoint address cannot be inside of any IP range in the AllowedIPs list unless AllowedIPs is 0.0.0.0/0, and with marking, desired traffic always traverses the Vula device, and it does not leak unencrypted IP packets.

7.4.8 – Memorable and Secure: petnames. Vula's network based discovery and publication is built on top of the trivially insecure mDNS protocol. Local active attackers are able to trivially forge responses to queries broadcast to the local network segment. It is for this reason that we turn Vula peer hostnames under the existing .local namespace into a secure petname [Sti05] system.

Vula learns hostnames automatically as part of peer discovery. As currently implemented the Vula descriptor includes a .local hostname in its signed mDNS descriptor announcements. The default .local top level domain name is user configurable. The signed

.local hostnames in announcements from permanent peers are accumulated in a similar fashion as IP addresses already are: if a name is not already claimed, it will be added to the list of previously accepted names which that key has announced, all of the key's names resolve to the latest IP announced. By default, Vula scopes the name to only allow for claiming names under .local, or by a user setting a specific policy. This prevents an attacker from claiming a popular hostname while allowing them to claim a locally relevant hostname [5].

The Vula hosts file is used by a Name Service Switch (NSS) module [GNU20] which requires reconfiguration of `/etc/nsswitch.conf`; our Vula implementation provides packages that perform this configuration automatically at package install time. Therefore, Vula provides protection of the authenticity of mDNS hostnames of participating Vula systems. The Vula vk is currently scoped to the hostname and only one vk may be the claimant of any single hostname, though in principle many hostnames is fine, none may conflict amongst all peers.

7.4.9 – Post-Quantum considerations by the CSIDH. Several approaches have been proposed for enhancing WireGuard with regard to attacks from quantum computers. In the Tiny WireGuard Tweak Chapter 6, we explain that to gain resistance to attacks by quantum computers, the Curve25519 public key used by WireGuard peers must be further concealed. The suggested enhancement is incompatible with Vula as the WireGuard public keys must be published with mDNS. We considered privacy improvements to mDNS and think this area is worth exploring in the future. However, absent privacy protections for mDNS service publications, we found the hiding of public keys to be impractical at this time.

In Post-quantum WireGuard [HNS+21], the authors proposed a post-quantum enhancement which effectively replaces the current WireGuard protocol with a post-quantum WireGuard protocol. Adopting this underlying protocol would add post-quantum protections for IP packets from attacks posed by universal quantum computers. The Post-Quantum WireGuard protocol has a great deal of promise. It additionally has practical implementation drawbacks for our envisioned deployment of Vula. Like WireGuard, it requires pre-configuration of peers by their public keys, and unlike WireGuard, it uses much larger public keys that do not easily fit in a single IP packet. The current implementation [HNS+20] is only available as a Linux kernel patch and it is incompatible with all other WireGuard implementations which makes cross platform support impractical.

One promising method to achieve post-quantum protection for traffic protected by the current WireGuard protocol is to set a per-peer pre-shared key. We had the idea to derive the PSKs by computation, using a different cryptosystem, rather than simply setting a symmetric key. If that system is a post-quantum key exchange, IP traffic will be further protected. We chose to use CSIDH for Vula to achieve transitional post-quantum security.

Each Vula peer announces their CSIDH-512 public key. This only adds 93 bytes to the mDNS service announcement when encoded as base64 and while accounting for DNS-SD overhead. All Vula descriptor data continues to fit into a single packet. Vula could rely on an architecture specific C implementation where computing a PSK for a peer with CSIDH would take roughly 111 milliseconds on x86_64 [CCC+19]. However, we have chosen portability over performance. This choice results in significantly longer computation time with a pure Python CSIDH implementation as shown in Table 7.3 and Table 7.5. Implementation details and platform specifics may dictate other constraints. With the pure Python CSIDH, we find it to be an acceptable but high computational cost for potential protection against an adversary with a quantum computer.

---

[5] After peer processing, the `vula organize` daemon writes a hosts file to disk in `/var/lib/vula-organize/hosts` which contains the current list of known hostnames and their respective IPv4 endpoints in classic `/etc/hosts` format.

Regarding the selection of CSIDH-512, CSIDH adds to the X25519 security already built into WireGuard. The purpose of this addition is to protect against the risk of quantum computers being built that are large enough to break X25519. Most post-quantum encryption options [NIS21] are Key Encapsulation Mechanisms (KEMs), which need point-to-point communication, leaving CSIDH as the only practically deployable non-interactive key exchange (NIKE) choice compatible with broadcast channels. Current estimates to break CSIDH-512 with a quantum computer take around 2^{60} qubit operations, each of those costing as much as roughly 2^{40} bit operations.

The use of a post-quantum signature system such as SPHINCS+-128s [BHK+19] could replace the use of Ed25519 in the Vula protocol as long as the mDNS record size does not exceed 9000 bytes [CK13b] split over multiple 1500 byte Ethernet frames if necessary. The SPHINCS+-128s signatures are 7856 bytes for 128-bit post-quantum security levels and the typical record size of a Vula key, value descriptor is around 300 bytes. We found that while the standard does allow larger record size, the underlying mDNS libraries we use did not. Furthermore, by using larger signatures we would move from a single 1500 byte packet for informing the entire local multicast group of a systems' descriptor to between five and six packets. Vula currently prioritizes solving the immediate problem of encrypting everything, but it will be important to integrate post-quantum signatures before an active attacker possesses a universal quantum computer.

7.4.10 – Verifpal verification. The following listing models the Vula protocol and shows our security queries. For an executable version see our anonymous page [Vul21].

```
attacker[active]

principal Laura[
        knows public _hkdf_salt
        knows public _hkdf_info
        generates time_stamp_a_0
        knows private vk_a
        vk_a_pk = G^vk_a
        generates csidh_a
        csidh_a_pk = G^csidh_a
        descriptor_a_pt0 = CONCAT(time_stamp_a_0,
           csidh_a_pk)
        ha_0 = HASH(descriptor_a_pt0)
        sig_a_0 = SIGN(vk_a, ha_0)
]

principal Glenn[
        knows public _hkdf_salt
        knows public _hkdf_info
        generates time_stamp_b_0
        knows private vk_b
        vk_b_pk = G^vk_b
        generates csidh_b
        csidh_b_pk = G^csidh_b
        descriptor_b_pt0 = CONCAT(time_stamp_b_0,
           csidh_b_pk)
        hb_0 = HASH(descriptor_b_pt0)
        sig_b_0 = SIGN(vk_b, hb_0)
]

Laura -> Glenn: [vk_a_pk], time_stamp_a_0, csidh_a_pk,
    sig_a_0

Glenn -> Laura: [vk_b_pk], time_stamp_b_0, csidh_b_pk,
    sig_b_0

principal Laura[
        x_0 = SIGNVERIF(vk_b_pk, HASH(CONCAT(
           time_stamp_b_0, csidh_b_pk)), sig_b_0)?
        ss_a = HKDF(_hkdf_salt, HASH(csidh_b_pk^
           csidh_a), _hkdf_info)
]

principal Glenn[
        y_0 = SIGNVERIF(vk_a_pk, HASH(CONCAT(
           time_stamp_a_0, csidh_a_pk)), sig_a_0)?
        ss_b = HKDF(_hkdf_salt, HASH(csidh_a_pk^
           csidh_b), _hkdf_info)
]

queries[
        freshness? sig_a_0
        freshness? sig_b_0
        freshness? time_stamp_a_0
        freshness? time_stamp_b_0
        authentication? Glenn -> Laura: sig_b_0
        authentication? Laura -> Glenn: sig_a_0
        confidentiality? ss_a
        confidentiality? ss_b
]
```

Listing 7.1: "Verifpal Vula model protocol"

7.4.11 – Additional FLENT performance graphs. The following graphs show some of the performance characteristics with different CPU architectures, and microarchitectures. The solid green and solid orange lines represent the upload and download performance for IP traffic processed by WireGuard. The dotted green and dotted orange lines represent the upload and download performance for IP traffic without any protection from WireGuard on the same system. The latency of IP traffic is represented by the solid purple line for WireGuard and the dotted purple line is without any protection from WireGuard.

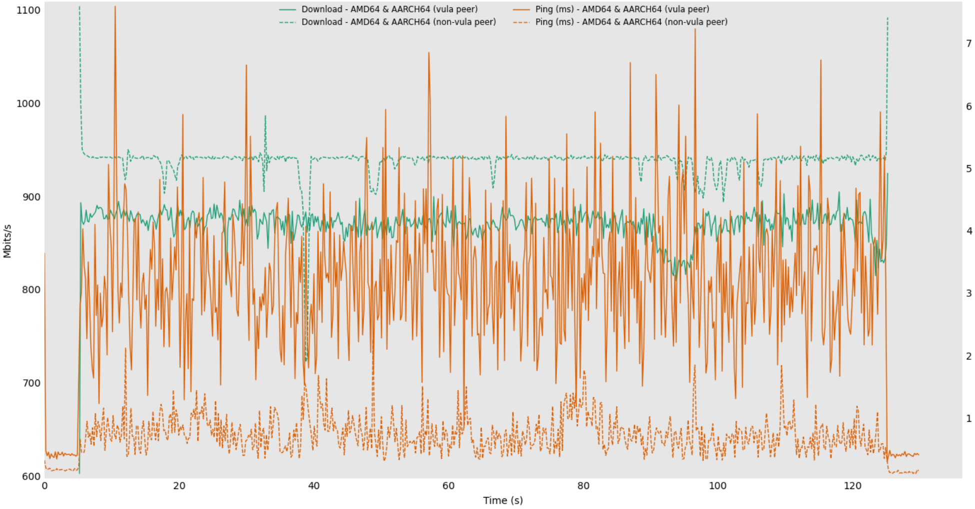


Figure 7.3: FLENT 12 stream down with ping; graph with and without WireGuard. AMD64 and AARCH64.

In Figure 7.3 we see the performance of a twelve stream iperf3 test with and without WireGuard between an AMD64 machine and an AARCH64 (ARM64) machine. The performance for gigabit traffic is as expected and fills roughly all available bandwidth modulo measurement noise.

In Figure 7.4 we see the performance of a twelve stream iperf3 test with and without WireGuard between an AMD64 machine and an RISC-V machine. The RISC-V machine has performance issues. It is not even able to sustain a full gigabit of traffic without WireGuard. Adding WireGuard shows a steady 200Mb/s which indicates that the RISC-V platform would greatly benefit from an optimized WireGuard implementation. That WireGuard works everywhere that Linux works helps with deployment and performance improvements may be made as needed for each CPU architecture.

In Figure 7.5 we see the performance of a twelve stream iperf3 test with and without WireGuard between an AMD64 (Intel i7) machine and an AMD64 (zen) machine. The performance difference between these two micro-architectures is nominal, and while improvements may be useful for speeds in excess of one gigabit, they are suitable for full gigabit saturation. The AMD64 architecture is extremely common and many home users likely have only AMD64 machines as their laptop or desktop endpoints.

In Figure 7.6 we see the performance of a twelve stream iperf3 test with and without WireGuard between a POWER9 machine and an AMD64 machine. The performance for gigabit traffic is as expected and fills roughly all available bandwidth modulo measurement noise.

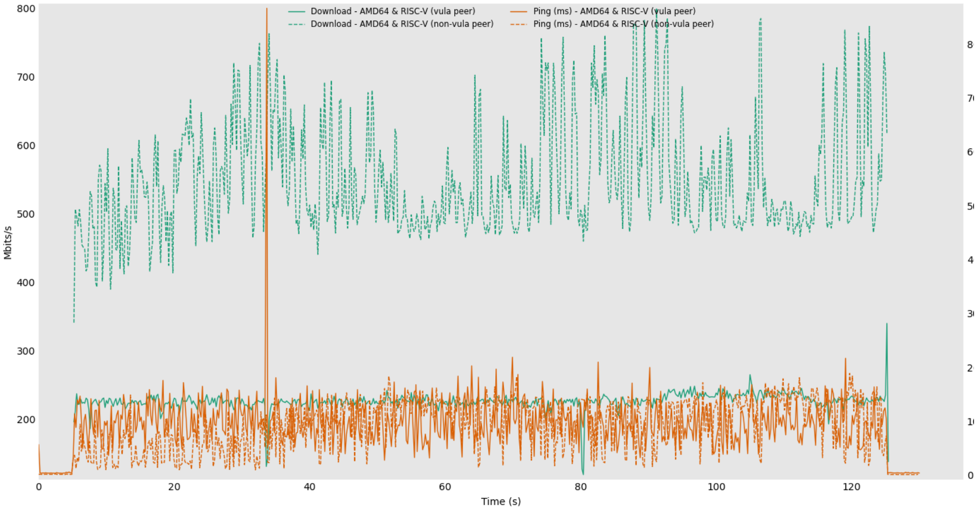


Figure 7.4: FLENT 12 stream down with ping; graph with and without WireGuard. AMD64 and RISC-V.

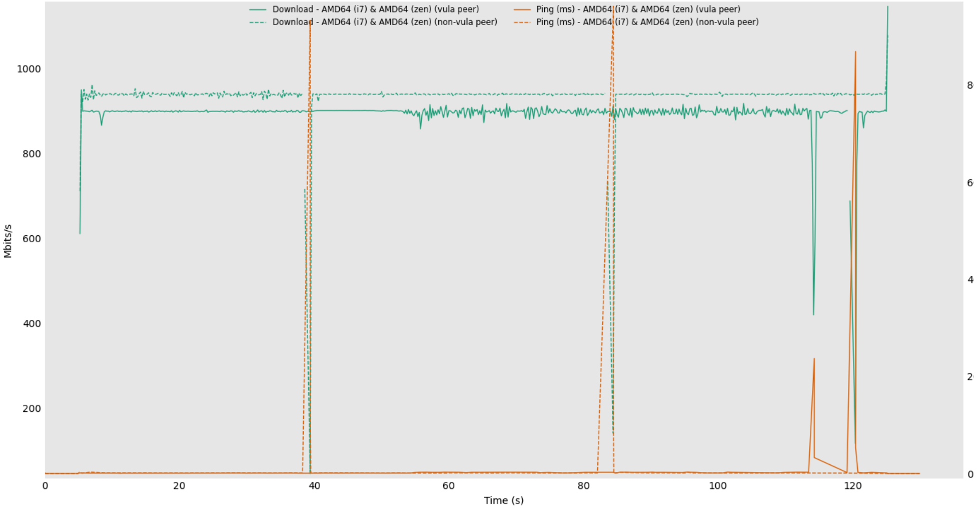


Figure 7.5: FLENT 12 stream down with ping; graph with and without WireGuard. AMD64 (i7) and AMD64 (zen).

The performance characteristics clearly show the benefits of architecture specific optimization. WireGuard is able to saturate gigabit Ethernet connections bidirectionally when two peers use modern AMD64 CPUs. WireGuard performance on more esoteric or otherwise new CPU architectures leaves something to be desired by comparison to optimized versions of itself on other platforms.

7.4.12 – systemd integration details. Vula is integrated into the system as multiple

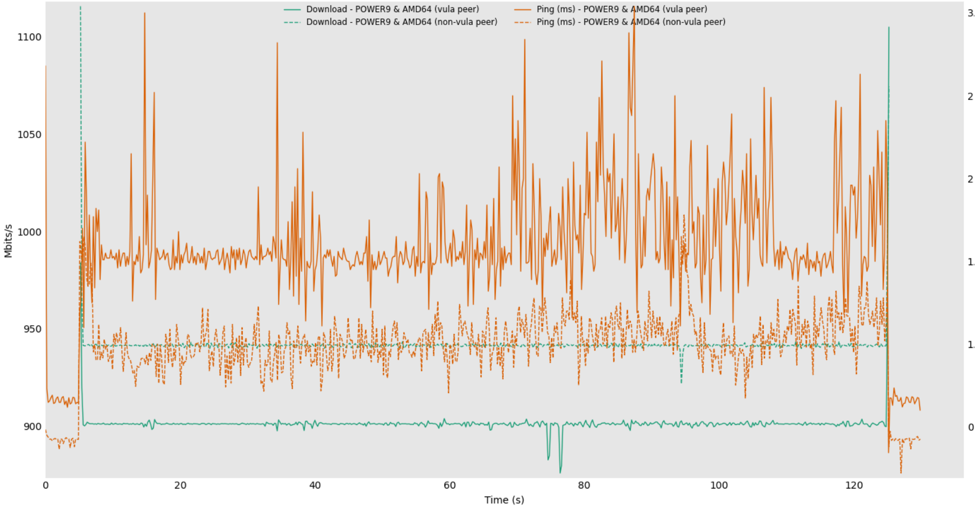


Figure 7.6: FLENT 12 stream down with ping; graph with and without WireGuard. POWER9 and AMD64.

daemons managed by systemd.

vula.slice. The `vula.slice` limits memory and other resources to ensure that none of the daemons run with systemd are able to consume excessive resources. All Vula daemons are a part of the `vula.slice`.

Discovery daemon. `vula-discover.service` runs the `vula discover` daemon which monitors for Vula publications on the local network. It runs as user `vula-discover` and as group `vula`. It requires access to the local network segment.

`vula discover` listens for mDNS service announcements under the DNS-SD label of `_opabinia._udp.local.` and it outputs each discovered mDNS WireGuard service. The output is a peer descriptor contained in a single line for each discovered WireGuard service. The peer descriptor contains all the information needed to reach and configure the newly discovered WireGuard peer. For each Vula service detected, it constructs a descriptor which is then sent to the `vula organize` daemon.

Publish daemon. `vula-publish.service` runs the `vula publish` daemon and publishes the mDNS Service record on the local network. It runs as user `vula-publish` and as group `vula`.

`vula publish` is a standalone mDNS service announcer which does not conflict with other mDNS programs commonly found on GNU/Linux systems such as avahi-daemon. It receives instructions from the `vula organize` daemon via `d-bus`, or via a python function call in monolithic mode, and publishes service records containing specifically formatted data signed under a Vula specific Ed25519 private key.

Configuration daemon. `vula-organize.service` runs the `vula organize` daemon which reads peer descriptors from a systemd managed socket. It runs as user `vula-organize` and as group `vula`. It does not access the network and its primary purpose is to configure the local `vula device` WireGuard interface. It retains the capability [Lin20] `CAP_NET_ADMIN` to ensure it has the relevant authority and permission to modify the interface.

Additional implementation details. `vula organize` will generate cryptographic keys and write out data to the following files:

0. `/var/lib/vula-organize/keys.yaml`
   CSIDH secret key, Curve25519 private key, and the Ed25519 private key for the `vula organize` daemon.
1. `/var/lib/vula-organize/vula-organize.yaml`
   Configuration file containing relevant Vula state for the `vula organize` daemon.
2. `/etc/systemd/system/vula-organize.service`
   A systemd daemon configuration file.
3. `/etc/systemd/system/vula-publish.service`
   A systemd daemon configuration file.
4. `/etc/systemd/system/vula-discover.service`
   A systemd daemon configuration file.
5. `/etc/systemd/system/vula.slice`
   A systemd slice to contain and constrain the aforementioned systemd services.

`vula configure` will add a firewall rule using `ufw` to allow traffic to the `vula` interface (`ufw allow 5354/udp`):

```
To                         Action      From
--                         ------      ----
5354/udp                   ALLOW IN    Anywhere
5354/udp (v6)              ALLOW IN    Anywhere
```

7.4.13 – Adversary realities. Often when writing about adversaries it is difficult to point to specific tools that may motivate specific design goals. Thanks to some very special whistleblowers, we have evidence from inside one of the largest, and well funded state level adversaries on the planet. We know that cryptography is a hard barrier [AGG+14e] for successful surveillance by such adversaries. It is reasonable to expect that other large state adversaries have similar limitations, similar tools, or even access to the same tools based on geopolitical agreements.

We consider two of the products from the ANT catalog in Chapter 4.6. NIGHTSTAND as seen in Figure 4.53 is a so-called close access operation tool for attacking wireless devices. SPARROW II as seen in Figure 4.54 is a so-called Airborne Operations tool for monitoring wireless networks. These two devices represent tools which exemplify the passive adversary (SPARROW II), and the active adversary (NIGHTSTAND).

7.5 — Performance

In this section we consider the performance of Vula's cryptographic choices. Post-quantum protections provided by CSIDH is explored in Section 7.5.1. Bulk encryption is handled by WireGuard in-kernel, and we consider its performance in Section 7.5.2. Additional measurements are available in Section 7.4.11.

7.5.1 – CSIDH performance evaluation. At peer discovery time, Vula uses CSIDH to generate a pairwise PSK. Each peer computes a shared key computation using its own secret key and the respective public key of each other peer. We show the performance time in Table 7.3, Table 7.4, and Table 7.5 for six popular CPU architectures.

Table 7.3: Python CSIDH-512 shared key computation execution time in seconds averaged over 128 runs

Arch	amd64	amd64
CPU	i7-9750H	Zen R1606G
Frequency	3.4Ghz	1.39Ghz
CSIDH	6.25	9.38

Arch	aarch64	armv7l
CPU	Cortex-A72	Exynos5422
Frequency	1.5Ghz	2.0GHz
CSIDH	26.16	53.28

Table 7.4: Python CSIDH-512 shared key computation execution time in seconds averaged over 128 runs

Arch	ppc64le	riscv64
CPU	POWER9	rv64imafdc
Frequency	3.2GHz	1.5Ghz
CSIDH	20.01	130.324

Table 7.5: Python CSIDH-512 shared key computation execution time in seconds averaged over 128 runs

The performance of the pure python CSIDH leaves much to be desired and is an area in need of architecture-specific performance improvements. A pure C implementation that has speed as the only goal is around two orders of magnitude faster than the pure Python approach. A hybrid approach of extending the Supersingular Isogeny-Based Cryptographic constructions (SIBC) Python module with C for architecture specific operations is almost certainly an ideal compromise. Caching of CSIDH derived symmetric keys as well as background computation for the shared key computation additionally improve performance. We consider that CSIDH computations present a possible denial of service vector to Vula as multi-millisecond computations still leave an easy denial-of-service attack vector.

7.5.2 – Network performance evaluation. We examined the performance in both ideal lab conditions and in an actual home network deployment. Performance greatly varies by CPU architecture as shown in Tables 7.3, 7.4, and 7.5.

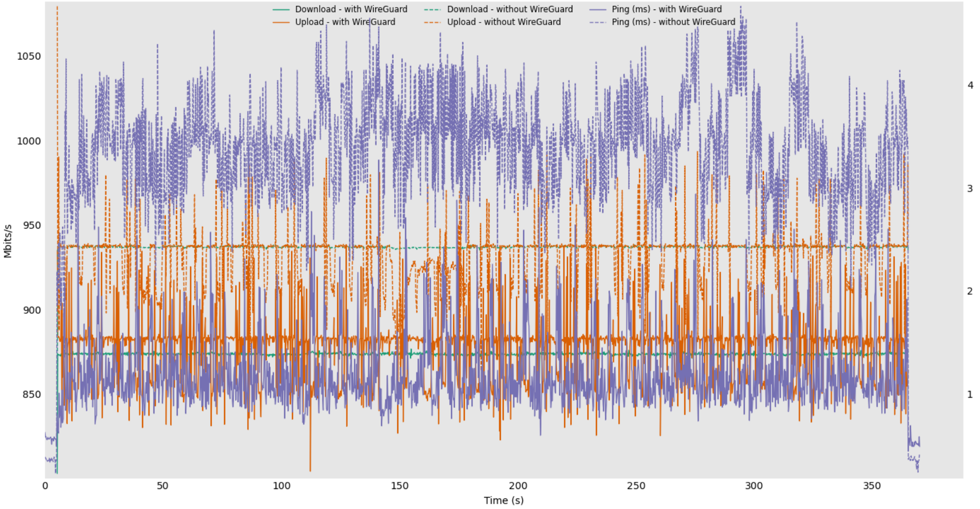


Figure 7.7: FLENT throughput and latency measurement

We observed that performance is not an issue with the underlying WireGuard transport. We found that WireGuard was able to sustain a consistent 1Gb/s in each direction using full duplex Ethernet devices as seen in Figure 7.7 using a FLENT [HJGHB17,Fle17] TCP bidirectional measurement test. The solid green and solid orange lines represent the upload and download performance for IP traffic processed by WireGuard. The dotted green and dotted orange lines represent the upload and download performance for IP traffic without any protection from WireGuard on the same system. The latency of IP traffic is represented by the solid purple line for WireGuard and the dotted purple line is without any protection from WireGuard.

Notice that Figure 7.7 shows that IP traffic latency performance is sometimes *better* when IP packets are encrypted with WireGuard. This is surprising as we would expect packet processing to take a constant amount of time and for WireGuard encryption to incur an extra cost in addition; this is true and due to kernel scheduling, WireGuard packets appear to be processed faster in many cases under load.

Our primary test systems for this evaluation were an Intel NUC running Ubuntu 20.04 with an Intel i7-8705G CPU and an AMD Ryzen Embedded R1606G with Ubuntu 20.10. The NUC has an Intel I219-LM Gigabit Ethernet device and the AMD system uses an Intel I211 Gigabit Ethernet device. The switching fabric used is the prosumer Unifi Gigabit Ethernet by Ubiquiti. Latency is naturally increased as a side effect of sustained 2Gb/s traffic over time. When not under extreme network load, the latency is nearly indistinguishable.

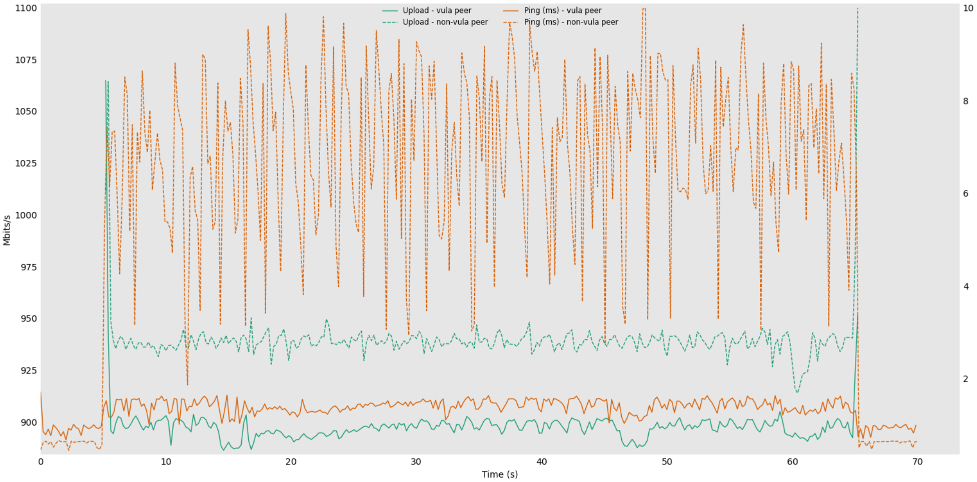


Figure 7.8: FLENT throughput and latency measurement

In Figure 7.8 we examine transmission of multiple flows with high performance switching equipment from Allies Telesis (x930 series; 48 1Gb/s ports) using two Dell PowerEdge R240 systems (Intel(R) Xeon(R) E-2124 CPU @ 3.30GHz with BCM5720 gigabit network card). We see that the throughput and latency for transmitted packets are again consistently lower than when WireGuard is not used, and with significantly less variability. The solid green line represents traffic to another Vula peer and this traffic is protected by WireGuard. The dotted green line represents traffic to a non-Vula peer. The solid orange line shows latency with a Vula peer and the dotted orange line shows latency to a non-Vula system. The difference in total bytes of payload sent remains to be investigated, and may be related to maximum transmission unit (MTU) of the underlying Ethernet network.

7.6 — Security Evaluation

The security of IP traffic protected by Vula is provided by the WireGuard protocol as outlined in the NDSS 2017 paper [Don17a], which relies on Curve25519, ChaCha20, Poly1305, and BLAKE2. It is further enhanced by setting the optional WireGuard peer-wise pre-shared symmetric key which the Vula protocol generates using CSIDH-512.

Vula's protection against active adversaries on descriptor announcements as described in Section 7.4.3 is dependent on the security of Ed25519 signatures, and a specific order of operations as outlined in section 7.6.1 through section 7.6.5.

7.6.1 – Vula Security Goals. The Vula protocol aims to automatically protect IP traffic for the local Ethernet segment with end-to-end encryption. When Laura wishes to transmit an IP packet to Glenn, and this traffic would otherwise be sent directly on the local segment, Vula will configure the local system to automatically upgrade the security of the IP packets by sending them over the *Vula* WireGuard interface. The Vula interface does not have an IP address assigned to it; the IP packets have the same IP addresses inside and outside of the tunnel.

7.6.2 – Data and Metadata. Users of wireless and wired Ethernet networks leave behind a number of unique data points. The Vula protocol seeks to reduce the data and metadata sent unencrypted overall. However, broadcast traffic such as mDNS data, including Vula announcements, any layer-two traffic such as ARP traffic with the MAC address [6] or addresses of each system, and traffic to hosts which are not using Vula, will be unencrypted as usual. Users of the Vula system will additionally generate a signed descriptor that may be verified by any third party.

7.6.3 – Formal verification. We have proven that our model of the Vula protocol in Listing 7.1 is secure against both passive and active adversaries using the Verifpal [Kob19, KNT20] symbolic formal verification tool. Verifpal has cryptographic constructions that make modeling protocols a straightforward, easy to read, easy to understand exercise. Verifpal models form a basis for security and privacy-property-centric queries and thus proofs of protocol properties. Our Verifpal model captures the conditions and constraints expressed in Section 7.3.4, and the Verifpal analysis confirms that the Vula protocol is secure in the passive attacker model without any public key verification, and that it is secure in the active adversary model if the long term vk public keys are verified. The queries show that descriptor updates are fresh, signed, authenticated, and that long term secret keys stay confidential:

```
queries[
  freshness? sig_a_0
  freshness?  time_stamp_a_0
  authentication? Glenn -> Laura: sig_b_0
  confidentiality? ss_a
]
```

The first asks about the freshness of the signature from Laura. The second asks about the freshness of the *vf* timestamp. The third asks about the authentication of the signature from Laura. The last query asks about the confidentiality of the shared secret computed by Laura. Verifpal confirms all these (and similar queries for Glenn) pass. Verifpal outputs: `Verifpal * All queries pass.`

---

[6] It has also been revealed that MAC addresses are used as a kind of covert communication channel about a system's cryptographic state. An example is that the MAC address of a common router platform may be used as a lookup for its initial cryptographic state. (This is a BULLRUN-style [AGG+14d, BBG13, PLS13, Lar13] cryptographic "*enabling*" in ARM CPU configuration for a popular router brand. Source: Private correspondence.) It also is understood that NSA and related adversaries collect MAC address information from drones and even from satellites in outer space [Ryg16] for geolocation reasons. As these MAC addresses are collected broadly, we encourage users to change the default MAC address at least once in the lifetime of their computer. This change should be inconsequential to everyone except saboteurs [BLN16]. As this kind of sabotage has become known as SIGINT *enabling*, we consider counter actions to be SIGINT *disabling*.

As always, formal analyses have limitations. Major risks that would not be ruled out by this analysis include the following: breaks in the WireGuard integration, cryptographic breaks in any of the assumed perfect cryptographic primitives, and/or any possible issues with Verifpal itself.

We have taken steps to secure the Vula traffic against active adversaries with our use of Ed25519. We have also taken steps to secure the Vula traffic against passive quantum adversaries who are able to record traffic and then later attack the recorded data's cryptography with their quantum computer. The protocol will need to be revised when quantum computers become available as Vula does not currently resist an active quantum adversary. Such an adversary should be able to forge Ed25519 signatures and would be able to publish new Vula descriptors to their advantage which would completely break Vula.

7.6.4 – Active attacks against Vula. Here we discuss some attacks against systems on a local area network with and without Vula.

Address Resolution Protocol. Unrelated to the Vula protocol, the Address Resolution Protocol (ARP) allows for selectively targeting users by carrying out an ARP poisoning [RN05] attack. An attacker able to successfully ARP poison a target is able to place themselves into an on-path position. This attacker may delay, drop, or modify traffic depending on the protection available for any given IP packet.

With Vula in place, an on-path attacker is no longer able to modify traffic protected by Vula, adversaries will only be able to delay or drop encrypted packets. They may be able to interfere with mDNS and other unencrypted broadcast traffic.

Dynamic Host Configuration Protocol. Users who use the Dynamic Host Configuration Protocol (DHCP) to obtain an IP address automatically from the network are susceptible to DHCP related attacks. Attackers acting as a DHCP server may assign a targeted user any address on any subnet, and they may change the lease of any system on the network segment to a new IP address. References to previous IP address assignments would then be stale until a new Vula descriptor containing the new IP address is broadcast to the network. We describe this as the DHCP attack in as mentioned in subsection 7.3.2.

MAC address vs IP address vs hostname security. Vula's use of petnames is important to security and participants should use the local names to address peers. To see the importance, consider either the ARP attack vector or the DHCP attack vector. Our petname system from Section 7.4.8 is a required part of ensuring Vula's security claims in either case. Hostnames for Vula peers are cached locally into hostname, IP address pairs based on atomic processing of Vula peer descriptors. Additionally, and most critically: some IP addresses are routed via the Vula WireGuard device and some are not. If users use the Vula protected hostname, they will receive the latest IP that is routed through WireGuard for that respective Vula peer. This ensures that unless there is a valid WireGuard session, the packets will buffer, or drop before being sent or if there is a valid session but the peer has moved, they will be sent encrypted but no replies are expected.

The Vula petname system from Section 7.4.8 does not prevent the DHCP attack from subsection 7.3.2 where a rogue DHCP server is attempting to trick users into using a new subnet entirely. Consider the case where Laura is connecting to Glenn's actual IP. Mallory published her key before she tricked Glenn into moving there. However, if users understand that they should rely on names to connect to Glenn's system rather than IP addresses when referring to Vula peers, then from Laura's perspective bob.local will continue to resolve to the last announcement from Glenn which she considered valid. As Glenn unbound his old IP when Mallory tricked him, Laura will not be able to reach him there, but the use of the name system has downgraded Mallory's attack from a confidentiality break to a mere Denial-of-Service. The same applies to the simpler ARP attack: as long as the packets first pass through the WireGuard device, they will be protected, and only by using hostnames is this guaranteed. This is why using secure names and having Vula perform the resolution is mandatory for Vula's security.

Traffic analysis. An attacker monitoring traffic as either an on-path or off-path attacker has the ability to perform traffic analysis such as website fingerprinting [HWF09] or other traffic classification [AGG$^+$14b]. This capability may allow for targeted on-path selective blocking even when traffic is protected. WireGuard, and thus Vula, does not attempt to resist traffic analysis through timing obfuscation, padding, or other schemes such as generating dummy traffic or mixing.

Selective blocking. Attackers can selectively prevent certain packets, such as Vula announcements, from being delivered. This can prevent new peers from being automatically discovered, and can prevent existing peers from learning about each other's address changes, which can cause Denial-of-Service. Such an attack does not allow a breach of confidentiality between peers that are already *pinned* at the time of the attack because of the secure petname system.

Continuity of Verification Keys. We learn about peers and index them by their verification key. All other keys are considered peer-specific state and we allow rotation of those values. This means that the WireGuard and CSIDH public key may be rotated by issuing a new descriptor signed by the vk keypair. It is for this reason that the Verification Key must not change as it is the root of trust, and additionally, the hostname list, and the IP address list, are indexed under the Verification Key. No two peers may have overlapping hostnames or IP addresses.

Key substitution. Absent pinning as introduced in Section 7.3.4, peers are *replaceable*, and active adversaries may announce their own descriptors, with conflicting resources, and with their own keys for two or more peers, enabling an active MITM attack. Peer substitution is possible against Vula peers which are not *pinned*, and new peers when they are making first contact. Using this attack, an active adversary can observe and bidirectionally forward IP packets between pairs of victims, or even in a single direction. This attack can be performed by an adversary with the capability to drop or replace targeted Vula peer mDNS announcements, and may be combined with other attacks such as ARP spoofing as mentioned in Section 7.3.2. Adversaries are not able to breach confidentiality for *pinned* peers. Peers in the replaceable state are only secure against passive adversaries, and peers in the pinned state are secure against active adversaries from peer replacement attacks.

Further protection against active adversaries. Vula makes some trade-offs by default which may be adjusted according to a specific deployment's desired security properties. All options are implemented in our Python reference implementation. Users of Vula concerned about active adversaries must configure Vula for their use case. By default, Vula does not require users to be aware of the software or any configuration option after installation. As mentioned in the peer replacement attack section, Vula peers are replaceable by default and this is only safe against a completely passive adversary. In the unpinned, replaceable state, peers expire, and any conflicting peer descriptor will simply replace the original peer entirely. Expiration of peers ensures that non-Vula systems will retain connectivity in the event of IP address reuse. In the pinned, permanent state, a user must resolve any resource conflicts manually, and the first peer descriptor to arrive and be pinned will always remain during automatic descriptor conflict resolution. To accommodate users who are unable or unwilling to manually resolve IP address conflicts, Vula defaults to all peers starting in a replaceable state. Changing the default is straightforward at install or run time. The *pin_new_peers* configuration option default state is *false*. The benefit is that normally-occurring IP address or hostname collisions will be handled normally as if Vula were not in use; the disadvantage is that active attackers are able to replace peers, and are *not* thwarted even while being detectable. This default configuration is intended to be suitable for all deployments without requiring any user awareness of Vula at all. For protection against active attackers who are not present before first contact, a knowledgeable user can choose pin automatically by setting *pin_new_peers* to *true*. This setting means that naturally-occurring IP address or hostname collisions will sometimes lead to an inability to communicate with affected Vula or non-Vula hosts. Pinned hosts may always update their own resources, and their descriptors must not conflict with any other pinned Vula peers or it will be rejected entirely. In any case, a user may always pin or verify a peer regardless of defaults, and they may set their own defaults at install time.

7.6.5 – Adversary evaluation. We consider the adversary definitions from Section 7.3 and their respective attack vectors. As described in Section 7.3.2 all layer-two traffic remains unprotected such as ARP as well as layer-three IP broadcast traffic. Any active adversary as described in Section 7.3 may delay, drop, and/or store any traffic where they have successfully performed an *ARP poisoning* attack. When operating with Vula, nearly all intranet traffic to participating systems will be protected in a forward-secret manner which defeats passive adversaries automatically. Using a trusted third party would allow for automatic trust decisions, but there is no suitable trusted third party in the context of every LAN, and across organizational boundaries.

Unilateral Surveillance. Vula completely defeats the **Unilateral Surveillance Adversary** in a forward-secret manner that is not dependent on a wireless passphrase. Thus, regardless if there is wireless encryption or wireless passphrase rotation policies, Vula successfully defeats the **Unilateral Surveillance Adversary** for the types of traffic which are protected by Vula. When the router deploys Vula, *all* traffic to the Internet may be protected from interception in the local LAN context.

End User. Vula *partially* defeats the **End User** adversary for the types of traffic which are protected by Vula. It reduces the adversary capabilities to denial-of-service as they are only able to delay or drop Vula protected traffic. Recording of the traffic is now largely useless thanks to the forward secrecy provided by the underlying WireGuard transport. The protection is bound by time of adversary arrival, default peer pinning status, and by peer verification status. If the **End User** adversary arrives before some other users, the adversary is able to claim possession of any currently unclaimed IP addresses, including addresses which may conflict with DHCP leases of newly arriving users. This would allow the adversary to be a Vula peer for an IP address assigned by DHCP to another user, thus impacting all subsequent users on the network. Active attacks by long-term adversaries can only be detected and defeated by manual key verification.

Network Operator. Vula successfully defeats the **Network Operator** adversary for the majority of the traffic which is protected by Vula. Intra-network traffic is protected between any set of systems deploying Vula, and peers should be pinned, as well as verified.

If the router also deploys Vula, *all* traffic to the Internet is protected from interception in the local LAN context. The **Network Operator** is still able to monitor it on the gateway itself. Using an additional protection mechanism such as a layered VPN may reduce the adversary capabilities to delaying or dropping traffic which is destined for the Internet.

7.7 — Conclusions

Vula enhances the confidentiality, integrity, and authenticity of IP traffic which is routed to or through another Vula peer. The cryptographic overhead with regard to performance for Gigabit networks is acceptable. Without any configuration, or even any user awareness that Vula exists on their system, the protection provided by Vula completely defeats a passive adversary, and active adversaries arriving after the first contact with automatic pinning. With user awareness and peer verification, all active adversaries are also defeated until such a time in which they have the ability to forge Ed25519 signatures, e.g. with access to an universal quantum computer. Our implementation has been released anonymously [Vul21] while under peer review as Free Software available for deployment, and is now deployed in experimental contexts including in laboratory, home, and as well as public networks.

CHAPTER 8

REUNION

8.1 — Introduction

In our era of targeted and mass surveillance [Rog15], encrypting data is often seen as a privacy panacea. Even as the newest protocols standardize [RE17] on privacy-by-design [LM13] patterns such as encryption and data minimization, we observe that metadata or content retention is inevitable in many systems. Modern protocol proposals in which unencrypted data *may* be intercepted should assume that it *will* be logged by a variety of systems, often unintentionally, and that it *will* later be analyzed by adversaries of unknown capability and with unpredictable consequences. The surest defense against this threat is to render the data valueless.

When a secure digital communication channel is in use, one party's computer may disclose information for reasons such as unsafe default preferences, simple programming mistakes, or legal mandates. Logs may be leaked or seized, and their contents used against either party or even unrelated parties. Telephones of all types expose specific identifiers, or selectors [PB14], whether telephone numbers, location data, or email addresses, leaving records that may be valuable [JAS13] to an adversary. Wiretapping [TM67] and pen registers [PJ78], lawful [Cou16] and otherwise [SC13], have long been used to collect metadata and content entirely apart from the *semantic* nature of the content. A variety of state actors systematically collect and store data shared over telephone or Internet channels, notable examples being XKeyscore [AGG+14a], operated by the United States National Security Agency (NSA); CALEA [10394] and DCSNet [Sin07], operated by the United States Federal Bureau of Investigation (FBI); and the series of Russian systems known generally as SORM [Wik17b] [Kri18] that was mentioned in Section 4.

In this thesis chapter we consider the problem of how to establish secure networked communication as a follow-up to a physical, offline meeting without computers, that is, a rendezvous. Informal offline protocols [Bla06] for exchanging contact information take many forms. People commonly exchange business cards with telephone numbers, email addresses, and even PGP fingerprints, as well as writing down a specific time and place to meet. These informal protocols may leave behind so-called *pocket litter* [Wik21n] [Nak08b] [RP14, Identity intelligence: Image Is Everything] that can be used as evidence in a variety of circumstances, legal or otherwise. It is also well known that the mere exchange of PGP fingerprints does not ensure that users will correctly verify and use them.

We propose REUNION, a network protocol designed to ensure that data used in a digital rendezvous, such as a shared passphrase, becomes effectively worthless before it can be used in unintended ways. We introduce the term *cryptographic rendezvous protocol*. Such a protocol allows two or more parties to selectively and securely discover desired users who share the same passphrase, and then to exchange contact information unobserved. We believe REUNION is the first transitionally post-quantum [SWZ16b] secure cryptographic rendezvous protocol.

REUNION allows users to safely use a low-entropy shared passphrase to exchange a single secure message in each direction. For example, participants could safely transmit difficult-to-remember contact information such as phone numbers, Ricochet identifiers [Joh14], email addresses [Res08], XMPP addresses [SA15b] and OTR fingerprints [BGB04], IP addresses and port numbers, VCARDs [Per11], or other short messages. RE-

---

[*] This work was previously unpublished. It is joint work with Johan Kjær, David Robinson, Leif Ryge, Kit Smeets, and David Stainton.

UNION could be used as an add-buddy wizard in a variety of applications, and is particularly well suited to secure messaging systems which eschew human-readable global identifiers.

8.2 — Background and related work

Password Authenticated Key Exchange (PAKE) protocols generally use a shared secret passphrase or password, or a machine-generated value, to perform and authenticate a key exchange. This chapter builds upon the use of a specific PAKE: Encrypted Key Exchange 2 (EKE2 [BPR00]) as implemented in the Phrase Automated Nym Discovery Authentication (PANDA) protocol [Lan12a]. PANDA is a cryptographic arbitrary passphrase or shared-secret rendezvous protocol. It is a two-round protocol as implemented by Langley in his Pond [Lan12b] messaging system. Historical details about PANDA deployment are available in subsection 8.2.1. The PANDA protocol exchange is shown in Figure 8.1.

The PANDA construction uses a shared passphrase to derive two URLs, each called a mailbox ID, and to derive a symmetric key used to encrypt a Curve25519 public key. Users find their respective peers' public key at the first URL, and the payload at the second URL. A record of these URLs is maintained by the server. The payload is encrypted with the resulting shared Diffie-Hellman key derived from their respective keypairs. The protocol provides a straightforward way to send a forward-secret payload such that later guessing of the password does not reveal the payload.

Later work by Warner called Magic Wormhole [War15a], built on SPAKE2 [AP05] with a generalized technique for sending a file in a single direction using a computer-generated, short, and easy-to-remember phrase. Magic Wormhole does not attempt to hide the metadata link between two parties.

PANDA and Magic Wormhole are both more than simple PAKE protocols. Rather than performing a key exchange over a pre-existing channel, they send a message to one or two parties without requiring a preexisting, bidirectional communications channel. The shared phrase not only authenticates a public key pair, it also creates a bidirectional channel and exchanges a message or file. We consider our REUNION protocol as a follow-up to the use case of PANDA and orthogonal to the use-case of Magic Wormhole as Magic Wormhole is not used bidirectionally for rendezvous. We distinguish cryptographic rendezvous protocols such as PANDA and REUNION as belonging to the general family of rendezvous protocols. There are several rendezvous protocols including PANDA and Magic Wormhole. In the world of IP based protocols, decentralized peer discovery often includes solving the problem of connecting n indirectly related parties to create n different communications channels. This problem appears in peer to peer centric protocols like Bittorent [Coh01] which require NAT-punching techniques such as Interactive Connectivity Establishment (ICE) [Ros10] that are used for creating end-to-end communication channels on the Internet when users are not otherwise directly reachable. However, we note that protocols for rendezvous are usually not trying to achieve any of the advanced security properties such as confidentiality *of the rendezvous*, post-quantum content protections, or any notion of unlinkability. In the following sections, we analyze PANDA and identify weaknesses.

8.2.1 – PANDA deployment history. The original PANDA protocol [Lan12a] was implemented as a Go program running on the Google App Engine service. It is reachable as a web service over HTTPS at an easy-to-remember URL. This default URL is embedded in the Pond client software [Lan12b]. To utilize the PANDA protocol in Pond, two users must agree on a common string to be treated as a shared secret; the base URL of the meeting location is embedded in the Pond client and is not user selectable. The shared secret is either a phrase chosen by the users arbitrarily or a value randomly generated by the Pond software which includes an embedded checksum to catch typing mistakes. The full meeting-place URL for any pair of peers is constructed from a URL and then a string that starts with a literal "/" followed by a hex-encoded meeting ID tag value that is derived from the shared secret. This serves as a location for Laura and Glenn to use HTTP POST and HTTP GET to run the three-round PANDA protocol 8.1. Each meeting place is valid for any two users to use once. After the protocol has been executed, the server blocks reuse of that URL again for approximately two weeks.

8.2.2 – PANDA and PANDA′. We have modeled PANDA with the symbolic formal proving system Verifpal [KNT20]. The PANDA protocol as described in Figure 8.1 matches the model in Listing 8.1.

8.2.3 – PANDA message flow. An informal message flow between Laura and Glenn is presented in Figure 8.1 for PANDA. They need only share a passphrase and to use a common PANDA server. The PANDA protocol's algorithm is explained in Algorithm 8.1. We present a model of PANDA in Listing 8.1 for use with Verifpal and we further discuss it in subsection 8.2.4.

Phrase Automated Nym Discovery Authentication (PANDA)

Laura	**Server**	**Glenn**
generates $\mathtt{msg}_a$		generates $\mathtt{msg}_b$
$skA \in \mathbb{Z}$		$skB \in \mathbb{Z}$
$pkA \leftarrow skA \cdot P \in E(\mathbb{F}_p)$		$pkB \leftarrow skB \cdot P \in E(\mathbb{F}_p)$
$(k, MP1, MP2) \leftarrow \mathsf{KDF}(Q)$		$(k, MP1, MP2) \leftarrow \mathsf{KDF}(Q)$
$r1_a \leftarrow \text{rijndael-enc}(k, pk_A)$		$r1_B \leftarrow \text{rijndael-enc}(k, pk_B)$
$\xrightarrow[\text{to url MP1}]{\text{send r1\_a}}$	stored (r1_a,MP1)	
	stored (r1_b,MP1)	$\xleftarrow[\text{to url MP1}]{\text{send r1\_b}}$
$\xrightarrow[\text{from url MP1}]{\text{fetch r1\_b}}$		$\xleftarrow[\text{from url MP1}]{\text{fetch r1\_a}}$
$pk_B \leftarrow \text{rijndael-dec}(k, r1\_b)$		$pk_A \leftarrow \text{rijndael-dec}(k, r1\_a)$
$s \leftarrow \mathsf{DH}(esk_A, pk_B)$		$s \leftarrow \mathsf{DH}(esk_B, pk_A)$
$r2\_a \leftarrow \text{secretBox.Seal}(s, \mathtt{msg}_a)$		$r2\_b \leftarrow \text{secretBox.Seal}(s, \mathtt{msg}_b)$
$\xrightarrow[\text{to url MP2}]{\text{send r2\_a}}$	stored (r2_a,MP2)	
	stored (r2_b,MP2)	$\xleftarrow[\text{to url MP2}]{\text{send r2\_b}}$
$\xrightarrow[\text{from url MP2}]{\text{fetch r2\_b}}$		$\xleftarrow[\text{from url MP2}]{\text{fetch r2\_a}}$
$\mathtt{msg}_b \leftarrow \text{secretBox.Open}(r2\_b, s)$		$\mathtt{msg}_a \leftarrow \text{secretBox.Open}(r2\_a, s)$

Figure 8.1: PANDA full exchange

Algorithm 8.1 PANDA protocol

Public Input: Curve25519 $E/\mathbb{F}_p$, base point $P \in E(\mathbb{F}_p)$, secretBox.Open() function, secretBox.Seal() function, key derivation function KDF() utilizing scrypt to derive 3 values, padding function randpad(msg, n) which pads msg with random data up to n bytes, block cipher rijndael-enc/ rijndael-dec returning 32 byte blocks, `PadMsg1Len` set to 32768, `PadMsg2Len` set to 32768, with timer `t` defined for 15 seconds.

Secret Input (Laura): Symmetric shared value `Q`, PANDA rendezvous server `RS`, Laura's Message to Glenn $\texttt{ContactBlob}_a$.

Output: Glenn's Message to Laura: $\texttt{ContactBlob}_b$

```
Laura generates ephemeral secret key $esk_A \in \mathbb{Z}$, public key $epk_A = esk_A \cdot P \in E(\mathbb{F}_p)$.
Laura computes $(PhraseDerivedKey, \texttt{MeetingPoint1}, \texttt{MeetingPoint2}) \leftarrow \mathrm{KDF}(\texttt{Q})$.
Laura computes enc-pk$_A \leftarrow$ rijndael-enc$(epk_A, PhraseDerivedKey)$.
Laura constructs msg$(a, 1) \leftarrow$ randpad(enc-pk$_A$, PadMsg1Len).
Laura publishes msg$(a, 1)$ to RS at the MeetingPoint1 location.
while msg$(b, 1) = 0$ do
    Laura waits for t seconds.
    Laura polls MeetingPoint1 location for Glenn's msg$(b, 1)$.
    if msg$(b, 1) \neq 0$ then
        Laura downloads msg$(b, 1)$ from MeetingPoint1.
    end if
    Laura sets $\texttt{t} = \texttt{t} \cdot 2$.
    if $\texttt{t} >= 3600$ then $\texttt{t} = 3600$.
    end if
end while
The PANDA server RS refuses new publications to the MeetingPoint1 location.
Laura checks the length of msg$(b, 1)$.
Laura reads the first 32 bytes of msg$(b, 1)$: enc-pk$_B \leftarrow$ msg$(b, 1)$.
Laura computes $epk_B \leftarrow$ rijndael-dec(enc-pk$_B$, $PhraseDerivedKey$).
Laura computes $s = \mathrm{DH}(esk_A, epk_B)$.
Laura computes $\texttt{M} \leftarrow$ secretBox.Seal(randpad($\texttt{ContactBlob}_a$, PadMsg2Len), s).
Laura creates a uint32 length prefixed message with the original message appended as msg$(a, 2) \leftarrow$ uint32(length(M))||M.
Laura publishes msg$(a, 2)$ to RS at the MeetingPoint2 location.
while msg$(b, 2) = 0$ do
    Laura waits for t seconds.
    Laura polls MeetingPoint2 location for Glenn's msg$(b, 2)$.
    if msg$(b, 2) \neq 0$ then
        Laura downloads msg$(b, 2)$ from MeetingPoint2.
    end if
    Laura sets $\texttt{t} = \texttt{t} \cdot 2$.
    if $\texttt{t} >= 3600$ then $\texttt{t} = 3600$.
    end if
end while
The PANDA server RS refuses new publications to the MeetingPoint2 location.
Laura checks that the length of the nonce is not less than 24 bytes; failure aborts.
Laura checks that the first two bytes of s are not zero; failure aborts.
Laura decrypts $\texttt{ContactBlob}_b \leftarrow$ secretBox.Open(msg$(b, 2)$, s); failure aborts.
Laura checks that the length of $\texttt{ContactBlob}_b$ is not less than four bytes; failure aborts.
Laura sets the first four bytes of $\texttt{ContactBlob}_b$ to represent a uint32 called ContactBlobLength.
Laura checks that the length of ContactBlobLength is larger than the length of $\texttt{ContactBlob}_b$; failure aborts.
Laura strips ContactBlobLength bytes of random padding from $\texttt{ContactBlob}_b$.
return $ContactBlob_b$.
```

8.2.4 – PANDA Verifpal model. The protocol model for PANDA is presented is presented in Listing 8.1. The following model proves that the PANDA protocol is secure in the active attacker model providing that both parties have a shared secret in common and use the same PANDA server for their run of the protocol.

```
attacker[active]

principal Laura[
        knows public panda_server
        knows password shared_secret_q
        generates message_a_0
        generates e_a_sk
        e_a_pk = G^e_a_sk
        phrasederivedkey_a, meetingpoint1_a, meetingpoint2_a = HKDF(nil, PW_HASH(shared_secret_q), nil)
        round1_a = ENC(phrasederivedkey_a, e_a_pk)
]

principal Panda_server[
        knows public panda_server
]

principal Glenn[
        knows public panda_server
        knows password shared_secret_q
        generates message_b_0
        generates e_b_sk
        e_b_pk = G^e_b_sk
        phrasederivedkey_b, meetingpoint1_b, meetingpoint2_b = HKDF(nil, PW_HASH(shared_secret_q), nil)
        round1_b = ENC(phrasederivedkey_b, e_b_pk)
]

phase[1]

Laura -> Panda_server: meetingpoint1_a, round1_a

Glenn -> Panda_server: meetingpoint1_b, round1_b

Panda_server -> Laura: round1_b

Panda_server -> Glenn: round1_a

principal Laura[
        glenn_pk = DEC(phrasederivedkey_a, round1_b)
        ss_a = glenn_pk^e_a_sk
        round2_key_a_0 = HKDF(nil, PW_HASH(ss_a), nil)
        round2_a = AEAD_ENC(round2_key_a_0, message_a_0, panda_server)
]

principal Glenn[
        laura_pk = DEC(phrasederivedkey_b, round1_a)
        ss_b = laura_pk^e_b_sk
        round2_key_b_0 = HKDF(nil, PW_HASH(ss_b), nil)
        round2_b = AEAD_ENC(round2_key_b_0, message_b_0, panda_server)
]

phase[2]

Laura -> Panda_server: meetingpoint2_a, round2_a

Glenn -> Panda_server: meetingpoint2_b, round2_b

principal Panda_server[
        ____ = HASH(round2_a)
        _____ = HASH(round2_b)
        _ = HASH(meetingpoint1_a)
        __ = HASH(meetingpoint1_b)
]

Panda_server -> Laura: round2_b

Panda_server -> Glenn: round2_a

phase[3]

principal Laura[
        contact_blob_b = AEAD_DEC(round2_key_a_0, round2_b,
            panda_server)?
]

principal Glenn[
        contact_blob_a = AEAD_DEC(round2_key_b_0, round2_a,
            panda_server)?
]

phase[4]

principal Laura[
        leaks shared_secret_q
]

principal Glenn[
        leaks shared_secret_q
]

queries[
        freshness? e_a_pk
        freshness? e_b_pk
        equivalence? phrasederivedkey_a, phrasederivedkey_b
        equivalence? meetingpoint1_a, meetingpoint1_b
        equivalence? meetingpoint2_a, meetingpoint2_b
        confidentiality? message_a_0
        confidentiality? message_b_0
        freshness? meetingpoint1_a
        freshness? meetingpoint2_a
        freshness? meetingpoint1_b
        freshness? meetingpoint2_b
        authentication? Panda_server -> Laura: round2_b[
                precondition[Glenn -> Panda_server: round2_b]
        ]
        authentication? Panda_server -> Glenn: round2_a[
                precondition[Laura -> Panda_server: round2_a]
        ]
        confidentiality? meetingpoint1_a
        confidentiality? meetingpoint2_a
        confidentiality? meetingpoint1_b
        confidentiality? meetingpoint2_b
        unlinkability? meetingpoint1_a, meetingpoint2_a
        unlinkability? meetingpoint1_b, meetingpoint2_b
        unlinkability? meetingpoint1_a, meetingpoint1_b
        unlinkability? meetingpoint1_a, meetingpoint2_b
        unlinkability? meetingpoint1_b, meetingpoint1_a
        unlinkability? meetingpoint2_b, meetingpoint2_a
]
```

Listing 8.1: "PANDA protocol Verifpal model"

The Verifpal model queries in Listing 8.1 are written with the expectation that some of the queries will pass and some will fail. The first seven queries pass and confirm that the expressed properties hold, including message confidentiality. The rest of the queries do not hold and expose weaknesses in the protocol. The equivalence of meetingpoint1_a and meetingpoint1_b, freshness of meetingpoint1_a and meetingpoint1_b, authentication of messages relayed by the Panda server between Laura and Glenn, confidentiality of meetingpoint1_a and meetingpoint2_a as well as meetingpoint1_b and meetingpoint2_b, do not hold. Furthermore, all of the subsequent unlinkability queries fail due to the failure to ensure freshness for those same values. These findings reveal that the server has the ability to link a pair of users, as well as to modify messages. Most critically, we learn that the outputs for a given secret are identical unfresh outputs that leak information about the secret. This latter issue allows for two critical vulnerabilities: the one-time possibility of precomputing a dictionary of words or phrases into meetingpoint candidates, and ability of a dishonest server to search this dictionary for any meetingpoint values used by clients. The search is performed offline by trying each item in the dictionary using either the meetingpoint1_a or meetingpoint1_b as an oracle for confirmation. We have implemented this attack in Section 8.2.6. PANDA′ primarily changes the pre-computation ability of an adversary from building a dictionary once, anytime, to starting the process at the time a shared random value (SRV) [SJK+17] [Pro17] is released and valid.

We additionally define a simple derivative protocol, *PANDA′* introduced in preproduction software [Sta18], by replacing scrypt [Per12] with argon2id [BDK16] and by using an SRV as the salt to the password hashing process. This change is applied to the KDF shown at Step 2 in Algorithm 8.1. PANDA′ performs user rendezvous in a chat program known as Katzen [Cat19], which is built on the Katzenpost [Kat] mixnet. We consider PANDA′ a marginal improvement over the original PANDA protocol.

8.2.5 – Exploiting PANDA. We present an attack breaking both PANDA and PANDA′. Beyond this attack, we also comment on the security consequences of how PANDA and PANDA′ are deployed. First-generation PANDA was implemented on top of Tor and Tor onion services. Second-generation PANDA′ was implemented for the Katzenpost mixnet [Sta18]. In these deployments, the anonymity protections are almost entirely external, being provided by the Tor or Katzenpost transport protocols. The original PANDA protocol was first implemented and deployed on Google App Engine, a provider widely known to have been targeted and compromised [Nak13b] [BA13] by state sponsored, and other well-funded adversaries. The protocol itself ensures a one-to-one correspondence for each round of messages for parties who correctly rendezvous, while attempting to make every pair of rounds unlinkable to any other pair of round messages. While exclusively used by Pond historically and only through the Tor network [DMS04], IP address information is available to any server unless care is taken in the client to establish an anonymous communication channel. Tor circuit reuse may make it easier for an adversary to link rounds of the protocol to specific client requests.

Because PANDA's mailbox IDs are derived solely from the passphrase and stored persistently on the server, a dictionary of password guesses can be precomputed before users even begin to use the protocol. If a malicious server is able to guess the password before users complete their run of the protocol, it will be able to read and modify the messages exchanged. When PANDA is used to bootstrap a secure communication channel in applications such as Pond, this allows a malicious server which has guessed the password before the protocol is completed to become a persistent man-in-the-middle. We present Python code in subsection 8.2.6 to implement an offline brute-force attack against the mailbox ID.

We analyzed the protocol manually and modeled it later in hopes of automating the discovery of similar results. Consider our three party (Server, Laura, Glenn) PANDA Verifpal model in Listing 8.1. When processed by Verifpal, a number of issues are automatically found in the protocol. The protocol verification execution on a four core i7-8705G CPU takes roughly 7 minutes and 12.284 seconds. The proof and results are available in subsection 8.2.4.

8.2.6 – PANDA offline brute-force. In listing 8.2 we have developed example code in Python to calculate the mailbox ID used by PANDA to establish contact between the participants. This ID is derived directly from the password and static values, implying that the server can use this ID to guess the password by performing brute-force calculations, rendering the PAKE part of the protocol ineffective at protecting the passphrase from offline attacks.

```
#!/usr/bin/env python2
import hashlib
import hmac
import scrypt

secret = 'foo'

def deriveKey(key, context):
  return hmac.new(key, context + key, hashlib.sha256).digest()

keySlice = scrypt.hash(secret , salt=b'', N = 1<<16, r=16, p=4, buflen=32)

a_tag_1 = deriveKey(keySlice , 'round one tag' )

print 'sc: ', keySlice.encode('hex')
print a_tag_1.encode('hex')
```

Listing 8.2: "Deriving PANDA mailbox ID from passphrase"

8.3 — Introducing REUNION

In this section we outline the REUNION protocol, explain the specific terms for discussing the protocol, and select appropriate cryptographic primitives. Building on PANDA, REUNION advances the state of secure rendezvous protocols in a number of areas.

REUNION is a three-round protocol that is run pairwise by each participant n times per epoch where n is the number of participants. These n users participate in the protocol run over a shared broadcast medium or via a centralized server. Each epoch lasts until the server increments the current EpochID as explained in subsection 8.3.1 or until the shared random value (SRV) used by end users has changed, the former is visible to all users of a server while the latter is only visible to client software who do not reveal which SRV they have chosen to the server. Without any public key infrastructure (PKI), exchanging contact information is extremely error prone. Unlike PANDA, REUNION ensures interactivity among *all* participating users of the protocol, increases receiver anonymity and sender deniability, removes assumptions about peers having a direct channel of communication, and gives both parties plausible deniability about a given phrase being associated with their transactions. Strictly speaking, REUNION is an inefficient messaging protocol: A participant commits to reveal a single plaintext message to other participants who share a common secret passphrase. Unlike PANDA, REUNION prevents intermediaries from discerning if a successful rendezvous has occurred, because all users communicate with each other, though only the subsets that share a passphrase can decrypt each other's messages. We call this property *rendezvous unobservability*.

The client software may choose to participate in further epochs to ensure an observer cannot link participation in the protocol and thus rendezvous success or failure to any given round of the protocol. We recommend that participation in any given round should be a stochastic process. After they have successfully performed a rendezvous with another party or n parties, the user should continue to use the protocol but with uniformly random passphrase shared with no other party in the next round, rather than a valid pass phrase.

The use of a random key ensures that the client software behaves in a random manner, and the coming and going of clients does not communicate information about success or failure but rather it is the result of random chance. Care must be taken of course to hide the use of REUNION itself. While a centralized REUNION server cannot tell who has performed a rendezvous, it can tell things like an end user IP address, and it can attempt to link users to REUNION's use - and look for intersections, and other trivial deanonymization attacks. Hiding a *user's use of REUNION* probably requires the use of an anonymity network as an intermediary, and we consider this fact out of scope for the notion of rendezvous unobservability. However, we do not think practically it is completely safe to ignore and any practical deployment must be analyzed with regard to the ability of systems such as XKeyscore to link users together in a variety of ways. As we see in MixMinion XKeyscore rules in Listing 4.1, as we see with the Tor network related XKeyscore rules, merely being linked to the use of an IP address associated with a service or network is enough to be selected for further surveillance. Considered from this perspective, if only two people in an entire city were to directly connect to a REUNION service and use it for one epoch, we could not known whether they had succeeded or not. However it is reasonable to infer that they did if their other communications channels are used to directly communicate. Thus care must be taken to consider the limits of the rendezvous unobservability in the cryptographic sense, and users must consider what they exchange and how that data is used.

8.3.1 – Building blocks. Before running the protocol, users agree out-of-band on an arbitrary passphrase. This string is a secret value used to mutually identify and authenticate two or more participants.

We define *REUNION-SERVICE* as a string for an agreed upon rendezvous service. An example might be the URL for a Tor onion service, a standard HTTPS URL, or even a service on the Katzenpost mixnet. It must have a normalizable way to be referenced as a string or another service-specific, globally unique identifier.

To bind every run of the protocol to a given point in time and to a location, we use a shared random value (SRV) as first introduced in Section 8.2.2, which is a commonly agreed upon byte-string synchronized from an out-of-band source. It must be impossible to predict it ahead of time. Both Katzenpost [ACD^{+}21] and Tor [Pro17] produce a regular SRV that is suitable for use with REUNION.

The server-issued EpochID value is at least 256 bits in length and is monotonically increased in each new epoch. It may also be randomly generated but care should be taken to never repeat an EpochID unless all SRV values in use have rotated.

Each protocol epoch uses a salt which is constructed by concatenation of the agreed upon SRV, the current EpochID value, and the normalized server URL. The salt is mixed into the password hashing function. Its purpose is to bind messages to a given channel at a given time, such that challenges and their responses cannot be cross-posted to several servers or a broadcast channel either by benign users or by attackers. This allows a user to limit interactive authentication attempts to users of a certain REUNION server's database, or to limit the number of attempts by different amounts for each REUNION server's database of round messages. An example could be that a protocol round message could only be accepted over a payment system's communication channel or through an authenticated channel.

We additionally require a number of standard cryptographic primitives such as the password hash function Argon2id [BDK16], the hash function BLAKE2 [ANWW13], and a standard HKDF [KE10a] construction to produce all of the round keys derived from the shared passphrase.

For message authentication, we have selected the Encrypt-then-MAC (EtM) ChaCha20 + Poly1305 construction as defined in RFC 7905 [LCM^{+}16]. Nonce compression here is trivial and overhead can be reduced to 0 bytes as each message uses each key only once.

For a secure pseudo-random permutation (PRP), we select Rijndael [DR02a] for a single block [1] Unlike AES [DBN+01], Rijndael may be used with a 256 bit block size and with 256 bit keys. This conveniently matches the size of a Curve25519 public key and has the desired block cipher security level.

To avoid providing an oracle with which offline attackers can guess passphrases, we encode our Curve25519 public key with Elligator 2 prior to encrypting it. Elligator 2 is a map from elliptic curve points to values that are indistinguishable from uniformly random strings, as well as a reverse map from random strings to curve points.

A Curve25519 public key is a curve point that is usually represented as a 256-bit wide string [2] which elligator maps to a 254-bit wide string. Two bits of random padding are added to produce a 256-bit value suitable for encrypting with our PRP.

Generating Curve25519 key pairs for use with Elligator 2 is more nuanced than it might initially seem [Vai20a, Vai20b]. First, only approximately half of all curve points can be encoded, so some keys must be rejected and a new key generated. Secondly, for any secret key, there are actually *eight* possible public keys. If the same choice of these eight were always used, as normal Curve25519 keypair generation does, then the public keys would be distinguishable from random curve points. In order for invalid passphrase guesses to be indistinguishable from valid guesses, it is critical that the public key we encode with elligator is indistinguishable from the curve points obtained by unelligatoring the random strings that decryption using incorrect guesses will produce. Our REUNION implementation uses the Monocypher [Vai19] library which provides a `crypto_hidden_key_pair` function for performing Curve25519 key generation in a way that is suitable for use with Elligator 2. This function takes a random 256-bit seed value and returns a 256-bit Curve25519 secret key and a 256-bit Elligator 2 encoding of a public key for it. The `unelligator()` function used in Algorithm 8.2 corresponds to Monocypher's `crypto_hidden_to_curve` function, which takes any 256-bit string and returns a point on the curve.

The use of Elligator 2 encoding ensures that a password-guessing adversary does not find a distinguisher as *every guess* leads to a valid Curve25519 public key. The lack of a distinguisher for a valid password guess forces participants to commit to a specific decryption of a public key with a single password as described in subsection 8.5.2. The PRP used to encrypt an Elligator 2 encoded Curve25519 public key, which is thus represented as a unformly random bit string, using a symmetric key.

In addition to Curve25519 and X25519 for computing shared values between two parties, we require a post-quantum, non-interactive key exchange (NIKE) primitive. This area of cryptography has one reasonable contender: CSIDH [CLM+18a]. We use the name CSIDH to refer to the function defined in [CLM+18a] that outputs a CSIDH shared key given a CSIDH secret key, and a CSIDH public key. CSIDH is considered an experimental primitive and may yet be broken by the wider cryptographic community. We think using an experimental primitive encourages development of attacks on CSIDH and it may also provide the claimed post-quantum protections. With the use of both X25519 and CSIDH, as well as with Elligator 2 and the use of a PRP with a passphrase, we emphasize that this is a hybrid protection scheme that should be as strong as the strongest of the set of cryptographic primitives.

---

[1] Only a single 256-bit block is encrypted with Rijndael, which thus acts as a permutation.

[2] Technically the top bit is always 0, so it's really a 255-bit value.

Algorithm 8.2 REUNION protocol

Public Input (Laura and Glenn): REUNION rendezvous server RS, Curve25519 $E/\mathbb{F}_p$ with Diffie-Hellman function $DH(private, public)$, base point $P \in E(\mathbb{F}_p)$, CSIDH $E'/\mathbb{F}'_p$ with Diffie-Hellman function $DH'(private, public)$, base point $P' \in E'(\mathbb{F}'_p)$, 32-byte EpochID, 32-byte SharedRandom, aead-enc($key, plaintext, ad$) function, aead-dec($key, ciphertext, ad$) function, hashing function $H()$ using BLAKE2, password hashing function argon2id( ), key derivation function HKDF( ), 32-byte block cipher rijndael-enc($key, plaintext$)/rijndael-dec($key, ciphertext$), random number generator function $RNG()$, Elligator 2 decode function unelligator($value$).

Secret Input (Laura): Shared secret `Q`, Laura's message to Glenn $\texttt{msg}_a$.

Output: Glenn's Message to Laura: $\texttt{msg}_b$ ▷ **Phase 0: Setup**

```
Laura enters shared secret $Q$ and $\texttt{msg}_a$ into her REUNION software.
Laura generates ephemeral Curve25519 and CSIDH key pairs:
        $esk_{A\alpha} \in \mathbb{Z}$, public key $epk_{A\alpha} = esk_{A\alpha} \cdot P \in E(\mathbb{F}_p)$.    ▷ Curve25519 (Elligator 2 encoded [a])
        $esk_{A\beta} \in \mathbb{Z}$, public key $epk_{A\beta} = esk_{A\beta} \cdot P' \in E'(\mathbb{F}_p{}')$.    ▷ CSIDH
Laura constructs salt $\leftarrow$ SharedRandom||EpochID.
Laura computes the password-derived key $pdk \leftarrow HKDF(salt, argon2id(salt, Q))$.
Laura generates two ephemeral secret symmetric keys, including entropy from the $msg_A$, the passphrase
  $Q$, and her random number generator:
        $sk_{A\gamma} \leftarrow H(pdk, RNG(32), msg_A)$,    $sk_{A\delta} \leftarrow H(pdk, RNG(32), msg_A)$.
Laura computes $T1_{A\gamma} \leftarrow$ aead-enc($sk_{A\gamma}$, "", RS).
Laura computes $T1_{A\delta} \leftarrow$ aead-enc($sk_{A\delta}, msg_a$, RS).
Laura computes $pdk_A \leftarrow H(pdk, epk_{A\beta}, T1_{A\gamma}, T1_{B\delta})$.
Laura computes $T1_{A\alpha} \leftarrow$ rijndael-enc($pdk_A, epk_{A\alpha}$.).
Laura constructs $T1_A \leftarrow T1_{A\alpha} \| epk_{A\beta} \| T1_{A\gamma} \| T1_{A\delta}$.
Laura transmits her $T1_A$ message to the RS.    ▷ Phase 1: Transmit $T1_A$
while $NewEpoch \neq EpochID$ do    ▷ Laura polls RS for replies to her $T1_A$.
    Laura asks the RS for all T1, T2, and T3 messages from EpochID.
    for each new $T1_{B_i}$ do    ▷ Phase 2: Process T1; transmit T2
        $pdk_{B_i} \leftarrow H(pdk, T1_{B_i\beta}, T1_{B_i\gamma}, T1_{B_i\delta})$.
        $epk_{B_i\alpha} \leftarrow$ unelligator(rijndael-dec($pdk_{B_i}, T1_{B_i\alpha}$)).
        $epk_{B_i\beta} \leftarrow T1_{B_i\beta}$.
        $dh1ss_i \leftarrow H(DH(esk_{A\alpha}, epk_{B_i\alpha}))$.    ▷ X25519
        $dh2ss_i \leftarrow H(DH'(esk_{A\beta}, epk_{B_i\beta}))$.    ▷ CSIDH
        $T2k_itx \leftarrow H(pdk_A, pdk_{B_i}, dh1ss_i, dh2ss_i)$.
        $T2k_irx \leftarrow H(pdk_{B_i}, pdk_A, dh1ss_i, dh2ss_i)$.
        $T2_{A_i} \leftarrow$ rijndael-enc($T2k_itx, sk_{A\gamma}$).
        Laura transmits her $T2_{A_i}$ message to the RS.
    end for
    for each new $T2_{B_i}$ do    ▷ Phase 3: Process T2, transmit T3
        $sk_{B_i\gamma} \leftarrow$ rijndael-dec($T2k_irx, T2_{B_i}$).
        if “” = aead-dec($sk_{B_i\gamma}, T1_{B_i\gamma}$, RS) then
            $T3k_itx \leftarrow H(T2k_itx, T2_{A_i}, T2_{B_i})$.
            $T3k_irx \leftarrow H(T2k_irx, T2_{B_i}, T2_{A_i})$.
            $T3_{A_i} \leftarrow$ rijndael-enc($T3k_itx, sk_{A\delta}$).    ▷ Encrypt $sk_{A\delta}$
        else
            $T3_{A_i} \leftarrow H(RNG(32))$.    ▷ Laura constructs a dummy T3 message
        end if
        Laura transmits her $T3_{A_i}$ message to the RS.
    end for
    for each new $T3_{B_i}$ do    ▷ Phase 4: Process T3; decrypt $\delta$
        $sk_{B_i\delta} \leftarrow$ rijndael-dec($T3k_irx, T3_{B_i}$).
        if $\texttt{msg}_{B_i} \leftarrow$ aead-dec($sk_{B_i\delta}, T1_{B_i\delta}$, RS) then
            Laura adds $\texttt{msg}_{B_i}$ to the list of results.
        end if
    end for
end while
```

---

[a] Curve25519 key generation here is limited to the space of elligatorable keys, which is approximately half of all keys. $epk_{A\alpha}$ refers to a 256-bit elligator encoding of the public key, including two bits of random padding. See Section 8.3.1 for details.

8.3.2 – Example dataflow for two users. In Figure 8.2 we show an overview of REUNION with $n = 2$ participants. In this example, they are exchanging messages using Ethernet as a broadcast medium as implemented in subsection 8.6.2. REUNION is a three-round protocol. It uses a shared broadcast medium or server, i.e. no password-specific addresses. Assume that there are n participants in a certain epoch, containing several pairs or cliques sharing the same password. Unlike PANDA, REUNION ensures interactivity among all participating users. The structure and meaning of the three types of messages is explained in subsection 8.3.3.

Message flow of REUNION on a local-area network, as described in subsection 8.6.2.

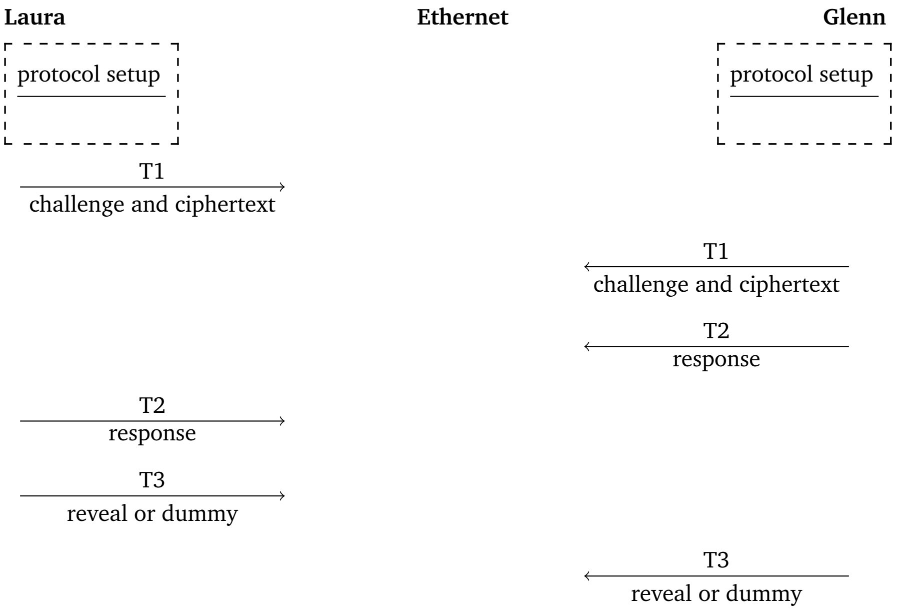


Figure 8.2: REUNION for $n = 2$; see Algorithm 8.2 for step-by-step details

8.3.3 – Protocol internals. The α part of the T1 message as seen in Step 9 in Algorithm 8.2 can be thought of as a challenge, and T1 as a whole may be considered a kind of commitment. The T1 message contains two public keys: one Curve25519 public key (α) and one unencrypted CSIDH public key (β). The final payload (δ) is decryptable by other participants who correctly respond to the challenge within a given protocol run.

The T1 message consists of four sections; the α, β, γ, and δ:

- α is an Elligator-encoded [BHKL13] Curve25519 [Ber06] public key. The Elligator-encoded public key is encrypted by a PRP using a symmetric key derived as described in Step 8. The encryption and decryption of α is unauthenticated, i.e. does not provide a validity check. The Elligator encoding ensures that plaintext is indistinguishable from random bytes and every sequence of random bytes maps to a valid public key.

- β is a CSIDH public key
- γ is an AEAD encryption of an empty string using a random key which is revealed by the T2.
- δ is an AEAD ciphertext containing the message payload which is only decryptable by a valid T3.

The T2 message consists of a single PRP ciphertext and is a response to the T1 challenge. The key is computed by using the shared passphrase `Q` and the DH key with the other party's T1 message.

A user recognizes T2 messages sent in reply to their T1 message and links it to the other party's T1 message by trial decrypting with the DH key. A valid response contains a symmetric key for the other party's MAC in the γ part of their T1.

If the decryption of the T2 is able to successfully verify the MAC of the γ, then a peer will be able to confirm that it is a valid response from a participant using the same passphrase.

The T3 message consists of a single PRP ciphertext and is a reveal for the previous challenge based on the T2 response. If the T1 γ ciphertext was decrypted successfully, then T3 message consists of a symmetric key for our T1 δ ciphertext. If the T1 γ was not successfully decrypted, the T3 is a uniformly random string with no relationship to any valid key.

It should be carefully noted that the encryption of the T2 and T3 messages does not need to be authenticated, because successful decryption can be confirmed by using their plaintexts to decrypt the γ and δ parts of a T1 message.

The protocol run lasts for some arbitrary but agreed upon period of time that it is specific to the client implementation. This we call ϵ, an epoch. A server may mediate all communication, or a broadcast channel may be used. For each T1 message sent by each participant, a T2 response and a T3 reveal must be sent by each of the other honest participants.

For each epoch, a server needs to maintain many append-only lists of messages. For each of the n T1 messages, there is a corresponding set of n T2 messages and a corresponding set of n T3 messages. Alternatively, a broadcast channel may be used, and clients perform basic bookkeeping tasks to track all messages. The messages may be exchanged over a myriad of protocols ranging from HTTP to custom binary protocols.

A collection of epochs may be thought of as a collection of append-only lists, and all participants are permitted to append to, or retrieve, the lists during the epoch. Each epoch must have a unique method for retrieving messages for the current epoch. Efficient downloading of the messages in a given epoch needs to be considered carefully by an implementation for each transport protocol.

8.3.4 – REUNION formal verification. We have modeled REUNION in Verifpal [KNT20], a symbolic formal verification system, and the model is presented in Listing 8.3. The REUNION protocol as shown in Figure 8.2 and in expanded form in Algorithm 8.2 corresponds to the model in Listing 8.3. Consider our two party REUNION model in Listing 8.3. When processed by Verifpal 0.25.0, no issues are found in the protocol; all queries pass.

REUNION Verifpal proof. Our Verifpal model of the REUNION protocol proves that the queried properties of the protocol hold in the active attacker model providing that both parties have a shared secret in common, that they keep the shared secret confidential until the protocol completes, and that they use the same REUNION server for their run of the protocol. Most importantly, we see that messages remain confidential, even after leaking the shared pass phrase, once the protocol has completed.

```
attacker[active]

principal Laura[
        knows private shared_secret_q_plus_srv
        pdk_a = HKDF(nil, shared_secret_q_plus_srv, nil)
        generates a_dh1_sk
        a_dh1_pk = G^a_dh1_sk
        generates a_dh2_sk
        a_dh2_pk = G^a_dh2_sk
        generates a_msg
        generates a_t1_gamma_seed
        generates a_t1_delta_seed
        a_t1_gamma_key = HASH(pdk_a, a_t1_gamma_seed, a_msg)
        a_t1_delta_key = HASH(pdk_a, a_t1_delta_seed, a_msg)
        a_t1_gamma = AEAD_ENC(a_t1_gamma_key, nil, nil)
        a_t1_delta = AEAD_ENC(a_t1_delta_key, a_msg, nil)
        a_t1_alpha_key = HASH(pdk_a, a_t1_gamma, a_t1_delta, a_dh2_pk)
        a_t1_alpha = ENC(a_t1_alpha_key, a_dh1_pk)
]

principal Glenn[
        knows private shared_secret_q_plus_srv
        pdk_b = HKDF(nil, shared_secret_q_plus_srv, nil)
        generates b_dh1_sk
        b_dh1_pk = G^b_dh1_sk
        generates b_dh2_sk
        b_dh2_pk = G^b_dh2_sk
        generates b_msg
        generates b_t1_gamma_seed
        generates b_t1_delta_seed
        b_t1_gamma_key = HASH(pdk_b, b_t1_gamma_seed, b_msg)
        b_t1_delta_key = HASH(pdk_b, b_t1_delta_seed, b_msg)
        b_t1_gamma = AEAD_ENC(b_t1_gamma_key, nil, nil)
        b_t1_delta = AEAD_ENC(b_t1_delta_key, b_msg, nil)
        b_t1_alpha_key = HASH(pdk_b, b_t1_gamma, b_t1_delta, b_dh2_pk)
        b_t1_alpha = ENC(b_t1_alpha_key, b_dh1_pk)
]

phase[1]

Laura -> Glenn: a_t1_alpha, a_dh2_pk, a_t1_gamma, a_t1_delta

Glenn -> Laura: b_t1_alpha, b_dh2_pk, b_t1_gamma, b_t1_delta

```

```
principal Laura[
        b_t1_alpha_key_a = HASH(pdk_a, b_t1_gamma, b_t1_delta, b_dh2_pk)
        b_dh1_pk_a = DEC(b_t1_alpha_key_a, b_t1_alpha)
        dh1_ss_a = b_dh1_pk_a^a_dh1_sk
        dh2_ss_a = b_dh2_pk^a_dh2_sk
        a_t2_key = HASH(a_t1_alpha_key, b_t1_alpha_key_a, dh1_ss_a, dh2_ss_a)
        b_t2_key_a = HASH(b_t1_alpha_key_a, a_t1_alpha_key, dh1_ss_a, dh2_ss_a)
        a_t2 = ENC(a_t2_key, a_t1_gamma_key)
]

principal Glenn[
        a_t1_alpha_key_b = HASH(pdk_b, a_t1_gamma, a_t1_delta, a_dh2_pk)
        a_dh1_pk_b = DEC(a_t1_alpha_key_b, a_t1_alpha)
        dh1_ss_b = a_dh1_pk_b^b_dh1_sk
        dh2_ss_b = a_dh2_pk^b_dh2_sk
        b_t2_key = HASH(b_t1_alpha_key, a_t1_alpha_key_b, dh1_ss_b, dh2_ss_b)
        a_t2_key_b = HASH(a_t1_alpha_key_b, b_t1_alpha_key, dh1_ss_b, dh2_ss_b)
        b_t2 = ENC(b_t2_key, b_t1_gamma_key)
]

phase[2]

Laura -> Glenn: a_t2

Glenn -> Laura: b_t2

principal Laura[
        b_t1_gamma_key_a = DEC(b_t2_key_a, b_t2)
        _ = AEAD_DEC(b_t1_gamma_key_a, b_t1_gamma, nil)?
        a_t3_key = HASH(a_t2_key, a_t2, b_t2)
        b_t3_key_a = HASH(b_t2_key_a, b_t2, a_t2)
        a_t3 = ENC(a_t3_key, a_t1_delta_key)
]

principal Glenn[
        a_t1_gamma_key_b = DEC(a_t2_key_b, a_t2)
        _ = AEAD_DEC(a_t1_gamma_key_b, a_t1_gamma, nil)?
        b_t3_key = HASH(b_t2_key, b_t2, a_t2)
        a_t3_key_b = HASH(a_t2_key_b, a_t2, b_t2)
        b_t3 = ENC(b_t3_key, b_t1_delta_key)
]

phase[3]

Laura -> Glenn: a_t3
```

```

Glenn -> Laura: b_t3

principal Laura[
        b_t1_delta_key_a = DEC(b_t3_key_a, b_t3)
        msg_b_a = AEAD_DEC(b_t1_delta_key_a, b_t1_delta, nil)?
]

principal Glenn[
        a_t1_delta_key_b = DEC(a_t3_key_b, a_t3)
        msg_a_b = AEAD_DEC(a_t1_delta_key_b, a_t1_delta, nil)?
]

phase[4]

principal Laura[
        leaks shared_secret_q_plus_srv
]

principal Glenn[
        leaks shared_secret_q_plus_srv
]

queries[
        freshness? a_dh1_pk
        freshness? b_dh1_pk
        unlinkability? a_dh1_pk, b_dh1_pk
        freshness? a_dh2_pk
        freshness? b_dh2_pk
        freshness? a_t1_gamma_key
        freshness? a_t1_delta_key
        freshness? a_t2_key
        freshness? a_t3_key
        freshness? b_t1_gamma_key
        freshness? b_t1_delta_key
        freshness? b_t2_key
        freshness? b_t3_key
        unlinkability? a_t1_gamma_key, b_t1_gamma_key
        unlinkability? a_t1_delta_key, b_t1_delta_key
        unlinkability? a_t2_key, b_t2_key
        unlinkability? a_t3_key, b_t3_key
        unlinkability? a_t1_alpha, b_t1_alpha
        unlinkability? a_t1_gamma, b_t1_gamma
        unlinkability? a_t2, b_t2
        equivalence? pdk_a, pdk_b
        equivalence? a_t1_gamma_key, a_t1_gamma_key_b
        equivalence? b_t1_gamma_key, b_t1_gamma_key_a
        equivalence? a_t3_key, a_t3_key_b
        equivalence? b_t3_key, b_t3_key_a
        authentication? Laura -> Glenn: a_t1_alpha
        authentication? Laura -> Glenn: a_t1_gamma
        authentication? Laura -> Glenn: a_t1_delta
        authentication? Glenn -> Laura: b_t1_alpha
        authentication? Glenn -> Laura: b_t1_gamma
        authentication? Glenn -> Laura: b_t1_delta
        confidentiality? a_dh1_sk
        confidentiality? b_dh1_sk
        confidentiality? a_dh2_sk
        confidentiality? b_dh2_sk
        confidentiality? a_t1_gamma_key
        confidentiality? b_t1_gamma_key
        confidentiality? a_t1_delta_key
        confidentiality? b_t1_delta_key
        confidentiality? a_msg
        confidentiality? b_msg
        confidentiality? msg_a_b
        confidentiality? msg_b_a
]
```

Listing 8.3: "Verifpal REUNION model protocol"

While the verification for the model in Listing 8.3 completes in approximately 0.5 core hours on a modern Intel CPU (Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz), the minimal single property model in Listing 8.4 took roughly 158,469 core hours (dual-socket AMD EPYC 7451 24-Core Processor), with verifpal-v0.23.0 24,632 core hours (dual-socket AMD EPYC 7742), and verifpal-v0.26.0 136,881 core hours (dual-socket AMD EPYC 7742).

```
attacker[active]

principal Laura[
        knows private shared_secret_q_plus_srv
        pdk_a = HKDF(nil, shared_secret_q_plus_srv, nil)
        generates a_dh1_sk
        a_dh1_pk = G^a_dh1_sk
        generates a_dh2_sk
        a_dh2_pk = G^a_dh2_sk
        generates a_msg
        generates a_t1_gamma_seed
        generates a_t1_delta_seed
        a_t1_gamma_key = HASH(pdk_a, a_t1_gamma_seed, a_msg)
        a_t1_delta_key = HASH(pdk_a, a_t1_delta_seed, a_msg)
        a_t1_gamma = AEAD_ENC(a_t1_gamma_key, nil, nil)
        a_t1_delta = AEAD_ENC(a_t1_delta_key, a_msg, nil)
        a_t1_alpha_key = HASH(pdk_a, a_t1_gamma, a_t1_delta, a_dh2_pk)
        a_t1_alpha = ENC(a_t1_alpha_key, a_dh1_pk)
]

principal Glenn[
        knows private shared_secret_q_plus_srv
        pdk_b = HKDF(nil, shared_secret_q_plus_srv, nil)
        generates b_dh1_sk
        b_dh1_pk = G^b_dh1_sk
        generates b_dh2_sk
        b_dh2_pk = G^b_dh2_sk
        generates b_msg
        generates b_t1_gamma_seed
        generates b_t1_delta_seed
        b_t1_gamma_key = HASH(pdk_b, b_t1_gamma_seed, b_msg)
        b_t1_delta_key = HASH(pdk_b, b_t1_delta_seed, b_msg)
        b_t1_gamma = AEAD_ENC(b_t1_gamma_key, nil, nil)
        b_t1_delta = AEAD_ENC(b_t1_delta_key, b_msg, nil)
        b_t1_alpha_key = HASH(pdk_b, b_t1_gamma, b_t1_delta, b_dh2_pk)
        b_t1_alpha = ENC(b_t1_alpha_key, b_dh1_pk)
]

phase[1]

Laura -> Glenn: a_t1_alpha, a_dh2_pk, a_t1_gamma, a_t1_delta

Glenn -> Laura: b_t1_alpha, b_dh2_pk, b_t1_gamma, b_t1_delta

principal Laura[
        b_t1_alpha_key_a = HASH(pdk_a, b_t1_gamma, b_t1_delta, b_dh2_pk)
        b_dh1_pk_a = DEC(b_t1_alpha_key_a, b_t1_alpha)
        dh1_ss_a = b_dh1_pk_a^a_dh1_sk
        dh2_ss_a = b_dh2_pk^a_dh2_sk
        a_t2_key = HASH(a_t1_alpha_key, b_t1_alpha_key_a, dh1_ss_a, dh2_ss_a)
        b_t2_key_a = HASH(b_t1_alpha_key_a, a_t1_alpha_key, dh1_ss_a, dh2_ss_a)
        a_t2 = ENC(a_t2_key, a_t1_gamma_key)
]

principal Glenn[
        a_t1_alpha_key_b = HASH(pdk_b, a_t1_gamma, a_t1_delta, a_dh2_pk)
        a_dh1_pk_b = DEC(a_t1_alpha_key_b, a_t1_alpha)
        dh1_ss_b = a_dh1_pk_b^b_dh1_sk
        dh2_ss_b = a_dh2_pk^b_dh2_sk
        b_t2_key = HASH(b_t1_alpha_key, a_t1_alpha_key_b, dh1_ss_b, dh2_ss_b)
        a_t2_key_b = HASH(a_t1_alpha_key_b, b_t1_alpha_key, dh1_ss_b, dh2_ss_b)
        b_t2 = ENC(b_t2_key, b_t1_gamma_key)
]

phase[2]

Laura -> Glenn: a_t2

Glenn -> Laura: b_t2
```

```

principal Laura[
        b_t1_gamma_key_a = DEC(b_t2_key_a, b_t2)
        _ = AEAD_DEC(b_t1_gamma_key_a, b_t1_gamma, nil)?
        a_t3_key = HASH(a_t2_key, a_t2, b_t2)
        b_t3_key_a = HASH(b_t2_key_a, b_t2, a_t2)
        a_t3 = ENC(a_t3_key, a_t1_delta_key)
]

principal Glenn[
        a_t1_gamma_key_b = DEC(a_t2_key_b, a_t2)
        _ = AEAD_DEC(a_t1_gamma_key_b, a_t1_gamma, nil)?
        b_t3_key = HASH(b_t2_key, b_t2, a_t2)
        a_t3_key_b = HASH(a_t2_key_b, a_t2, b_t2)
        b_t3 = ENC(b_t3_key, b_t1_delta_key)
]

phase[3]

Laura -> Glenn: a_t3

Glenn -> Laura: b_t3

principal Laura[
        b_t1_delta_key_a = DEC(b_t3_key_a, b_t3)
        msg_b_a = AEAD_DEC(b_t1_delta_key_a, b_t1_delta, nil)?
]

principal Glenn[
        a_t1_delta_key_b = DEC(a_t3_key_b, a_t3)
        msg_a_b = AEAD_DEC(a_t1_delta_key_b, a_t1_delta, nil)?
]

phase[4]

principal Laura[
        leaks shared_secret_q_plus_srv
]

principal Glenn[
        leaks shared_secret_q_plus_srv
]

queries[
  confidentiality? a_msg
]
```

Listing 8.4: "Verifpal REUNION model protocol; confidentiality only"

8.4 — Threat Model

The primary adversaries that we consider are Eve, the passive adversary; Mallory 1, the active guessing adversary; and Mallory2, the active fully participating confirming adversary.

Eve, the passive eavesdropper, should only be able to tell how many sessions per epoch there are. They should not be able to discern successful sessions from unsuccessful ones, and of course should not be able to tell who has successfully performed one or more rendezvous. Entirely offline attacks on the passphrase should not be possible. A non-participant like Eve is a passive confirming adversary when they try to confirm the use of the service after the fact from pocket litter or other information using a transcript of the protocol. This adversary is completely defeated by REUNION.

Mallory1, an active attacker who sends only T1 messages, can make online guesses about passphrases that might currently be in use. Her guess will be confirmed when the honest users using the passphrase send their T2 messages and allow her to decrypt the β portion of their T1 messages. Mallory1 is thus able to learn how many users are currently using this passphrase. Mallory1 is not detectable. Mallory1 does not learn the content of messages. If Mallory1 keeps the secret keys for her T1 message, creates a protocol transcript and later acquires a quantum computer, Mallory1 may be able to break the metadata but not the message confidentiality for correctly guessed passwords, after the protocol Epoch ends. If the secret is high-entropy, even a quantum adversary will not be able to break the metadata or content protections of REUNION.

Mallory2, an active attacker who completes the protocol, makes online guesses like Mallory1, but also sends corresponding T2 messages as an honest user would. If Mallory2 learns or guesses the password and successfully rendezvous with Laura before Glenn does in the same epoch, nothing can stop Mallory2, because the password is the **only** authenticator. However, if Glenn subsequently participates and there is either an honest server or the messages are exchanged over a censorship-resistant broadcast channel, Mallory2 is detectable by Laura and Glenn if they continue to run the protocol for the same or the next epoch.

A variation of the use case of the protocol would allow for n-way rendezvous. Instead of considering Mallory2 an attacker, the protocol as described above allows for n-way rendezvous. If n is greater than the number of people Laura and Glenn expected to know the passphrase, then they could realize that the passphrase has been compromised. Their client software could ask them to enter the expected number of peers in the rendezvous as n, and if the number of disclosures is larger than n, the client software could automatically be alerted to this fact.

8.4.1 – Safety against passphrase misclosure. Users write things down, and this is true for passphrases as well as other meaningful pieces of information. All of these things are subject to misclosure [Cai09]. In the PANDA protocol design, users could post messages to a server through use of a passphrase. However, PANDA leaked information about the shared secret phrase between two peers in a way that was checkable by a third party, potentially to an adversary's advantage. REUNION solves this problem. A passphrase must be kept secret until directly after the epoch in which it is successfully used. After a successful run of the REUNION protocol, a passive adversary cannot confirm the use of any secret phrase unless both parties suffered a failure of their random number generator. An active adversary with a quantum computer is able to confirm phrases in certain circumstances as we enumerate in Table 8.1.

8.4.2 – Pre-computation protection with an shared random value (SRV). REUNION also improves on existing designs by removing any pre-computation advantage through the use of dictionaries or the creation of rainbow tables [Hel80]. This is achieved through the use of a public SRV as part of the key derivation process. Key derivation is also improved by using a password hashing algorithm such as Argon2id that is tunable to the limits of a user's tolerance for perceivable computation. REUNION ensures that passive attackers are not able to guess at a passphrase offline; they must commit to a guess and participate in the protocol while it is being run by all other parties.

8.4.3 – The principle of least authority. REUNION follows the principle of least authority [Mil06] by removing centralized server responsibilities and redistributing that responsibility to clients. Servers are often a single point of failure [Wol04] (SPOF) in a variety of ways. Our design effectively limits the responsibility of the REUNION server to securely storing key-value data. The protocol makes a novel contribution to rendezvous protocols by abandoning the assumption of an anonymous communication channel, and instead creating receiver anonymity through interactivity among all peers.

8.4.4 – n-way rendezvous. All participating users send encrypted messages to a broadcast medium or server data store and download messages sent by all other users – from this process, all peers who share a single secret phrase will be able to decrypt each other's respective messages.

8.4.5 – Forward secrecy and deniability. Unlike PANDA, the shared secret phrase is not passively checkable, it is forward-secret, and it is deniable after the protocol is completed, when the secret may be revealed without users losing any of the security properties of the protocol. The one exception is noted in Table 8.1. This exception applies only for an adversary who participated in the protocol during the protocol run, records all messages sent in the epoch, and later uses a quantum computer in an attempt to confirm which passphrases were used.

8.5 — Security Evaluation

We consider a range of adversary strategies in Table 8.1 and Table 8.2.

The security of the REUNION protocol, as shown in Algorithm 8.2, is based on a combination of different fundamental hardness assumptions and/or Adversary capabilities: The entropy for a given shared passphrase, the entropy of randomly generated cryptographic key material, the security of the Curve25519 Elliptic Curve Diffie-Hellman Key Agreement, the security of the CSIDH Diffie-Hellman Key Agreement, the security of ChaCha20Poly1305 in the AEAD construction for protecting the payload message, and interactivity or participation in a given protocol run. In Table 8.2 and in Table 8.1 we consider how each of the protocol properties above contribute to defeating would-be adversaries. Some of these things are not like the others – interactivity for example makes it easier to sort between passive and active adversaries. Passive adversaries are nearly entirely defeated unless there are numerous cryptographic failures. Active adversaries remain powerful, but we have greatly reduced their capabilities in contrast to PANDA. If an active adversary acquires a quantum computer later, they will have special advantages over a passive adversary who later gains access to a quantum computer.

8.5.1 – REUNION Security Goals. The REUNION protocol is indirect, forward-secret, deniable, mutually authenticated, unlinkable, resistant to passive or offline brute-force guessing of the shared secret, resistant to precomputing passphrases, time limited, and interactive:

- **Weak Perfect Forward Secrecy** (wPFS [Kra05]): Our messages cannot be recovered retroactively by a non-interactive attacker.
- **Plausible Deniability**: No adversary can confirm that a certain passphrase was used for a given set of messages after a successful protocol run.
- **Indirect**: Data is transmitted and received from an otherwise untrusted facilitator.
- **Disclosure or misclosure protection**: Selectors [PB14] over insecure or compromised [Cai09] channels are not useful after use.
- **Brute force resistance**: Adversaries are forced to participate in the protocol, and cannot perform offline pre-computation.
- **Message confidentiality**: Unless a non-quantum adversary guesses a password interactively, they can never violate the confidentiality of the payload.
- **Metadata confidentiality**: Unless a non-quantum adversary guesses the password interactively, or participates *and* later obtains a quantum computer, while sending at least one T1 and receiving at least one T2 response, they cannot violate the confidentiality of the message metadata.
- **Post-quantum message confidentiality**: Unless an adversary guesses a password

	Attack description
A1	Interactive quantum adversary; sends a T1 with a CSIDH public key, runs trial-decryption of the X25519 public key with the candidate's password guesses. For each guess they compute the X25519 private key for their T1 commitment and compute the X25519+CSIDH for each T1 (eventually encompassing the target's T1), and are thus able to complete the protocol run.
A2	Non-interactive quantum adversary; utilizes a hypothetical vulnerability in CSIDH and the quantum ability to break X25519.
A3	Active adversary that knows the shared passphrase, having overheard the secret being shared between the two parties.
A4	Online brute force (limited by network speed and storage capacity of the infrastructure).
A5	A non-interactive attacker; knowing the shared secret gains no abilities. They can decrypt the T1 X25519 public keys, but cannot compute CSIDH and cannot derive the keys necessary to decrypt target T2 messages.
A6	An interactive quantum adversary; sends a T1 with a CSIDH key, runs trial-decryption of the X25519 public key from each T1, is able to learn the shared secret. Different from A1 because they do not learn the payload, and the participants don't get confirmation that the passphrase/metadata was compromised.
A7	Executes the protocol like the only remote party by censoring the legitimate T1. An attacker who knows or correctly interactively guesses the password *and* is in a position to censor messages between honest participants can perform a full message replacement attack.

Table 8.1: Adversary attack strategies

Table 8.2: Evaluation of attacks described in Table 8.1

Name:	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**
Requirements							
Send a T1's	yes		yes	yes		yes	yes
Send b T2's	yes		yes	yes			yes
MITM (censor T1's)							yes
Knows shared secret			yes		yes		yes
low-entropy (bruteforceable) secret	yes	yes				yes	
Can break X25519	yes	yes				yes	
Can break CSIDH		yes					
Attacker Ability							
Learns/confirms low-entropy shared secret	yes iff $a >= 1$	yes	yes	probability $a/keyspace$		yes	yes
Learn payload message	yes iff $b >= 1$	yes	yes	probability $b/keyspace$			yes
Prevent payload transfer							yes
Participant ability							
Can detect the attack			yes	yes			

interactively, they can never violate the confidentiality of the payload.

- **Rendezvous unobservability**: No third party can distinguish between participants who have successfully performed a rendezvous and those who have not. This notion is related to sender and receiver anonymity.

8.5.2 – Attack Analysis. We consider the T1 message which includes four sections: α, β, γ, and δ. We further consider the T2 and T3 messages which are used to unlock the γ AEAD ciphertext, and the δ ciphertext respectively.

The T1 α is an Elligator 2 encoding of a Curve25519 public key. This encoding ensures that there is no possible offline, non-interactive confirmation of a guess of a given shared phrase. We further increase the difficulty of correctly guessing a shared phrase with the construction of a key derivation function integrating HKDF [KE10a], the password hash function Argon2id, hash function BLAKE2, and by limiting possible pre-computation with the use of a high-entropy shared random value that is released at fixed time intervals. The α is valid for the Epoch ϵ, and encrypted under the key valid for the current Epoch. A password-guessing attacker with a quantum computer needs to decrypt and then solve the elliptic curve discrete logarithm problem (ECDLP) for each password guess using Shor [Sho99], and would need to do so interactively to perform a man in the middle, or after the fact, to attempt to decrypt. While we consider Shor sufficient to eliminate security from Curve25519 in theory, we remark that it may take years [ES21] or decades to exhaustively search, and solve ECDLP for all possible password guesses. This makes the protocol quantum-annoying [ES21]. The T1 β is an unencrypted CSIDH public key whose NIKE output is used in constructing the key used to encrypt the T2: The α and β are used to derive two shared secrets which are combined together through hashing. This can be thought of as combining the derived values from X25519 and CSIDH to create a transitionally secure Diffie-Hellman function that is at least as strong as X25519, and ideally as strong as CSIDH and X25519.

The resulting shared secret derived from a peer's α in the T1 message is resistant to pre-computation as explained in Section 8.4.2. The peer's β is claimed by the CSIDH authors to be post-quantum non-interactive key exchange (NIKE). It is not encrypted and so it does not present an opportunity for using the underlying CSIDH structure as a password guessing oracle.

The T2 response message is a thirty-two byte ciphertext encrypted using the selected PRP. When correctly decrypted, it contains a key for the peer's respective γ AEAD ciphertext. The generation of the T2 key used encrypt the T2 payload is shown in Steps 20, 21, and 22 in Algorithm 8.2. The T2 is a mix of previous protocol message hashes, shared secret values, and the passphrase. With the agreement of a high entropy shared secret, we observe that an attacker will not find a key pair with probability greater than 1 in 2^{256} guesses with random keys. A non-quantum adversary cannot use the challenge messages as a possible confirmation oracle. A quantum adversary will not be stopped by the hardness of ECDLP, though it is quantum-annoying. The entropy in the AEAD and the PRP keys is a barrier even for the quantum adversary. A non-quantum attacker should be stopped by either or both. A non-quantum attacker may not meaningfully interfere unless they are able to properly decrypt a peer's representative public key during the protocol run. They are unable to confirm their guess except by fully following the protocol to a trial decryption step and must flood the server with messages. An adversary who compromises a server is able to censor messages and rounds of the protocol, they may also add messages; in both cases absent knowledge of the passphrase, the server will only be able to perform various denial of service attacks.

After receiving a valid T2, the T1 γ may be unlocked. The T1 γ is a ciphertext which is used to signal that a participant's passphrase interactively matched the expected values. This ciphertext has 2^{256} possible keys and the ciphertext is designed to encrypt an empty string. The payload key is specially constructed to extract entropy from several sources even when the random number generator is potentially sabotaged. An attacker has a negligible chance of guessing this key as shown in Step 5. We postulate that with Grover [Gro96] this payload has 128 bits of post-quantum security. The key for the γ is resistant to pre-computation as it is not only directly derived from the shared passphrase but also the SRV and the participant's random number generator. The key generation process is designed to remain unpredictable even when the random number generator fails in certain conditions such as when an adversary strategically sabotages random number generator standards and implementations.

The T3 reveal message is a 32-byte ciphertext encrypted using the selected PRP. When correctly decrypted, it contains a key for the peer's respective δ AEAD ciphertext. The generation of the T3 key used to encrypt the T3 payload is shown in Steps 28, 29, and 30 in Algorithm 8.2.

After receiving a T3, the T1 δ may be unlocked. The T1 δ is a ciphertext which protects the participant's message to peers in the protocol. This ciphertext has 2^{256} possible keys and the ciphertext has an application-defined length. The payload is padded up to some application-specific size, and then encrypted with our selected AEAD. The payload key is specially constructed to extract entropy from several sources even when the random number generator is potentially sabotaged. An attacker has a negligible chance of guessing this key as shown in Step 5. We postulate that with Grover [Gro96] this payload has 128 bits of post-quantum security. The key for the δ is resistant to pre-computation as it is not only directly derived from the shared secret but also the SRV and the participants random number generator. The key generation process is designed to remain unpredictable even when the random number generator fails in certain conditions.

8.5.3 – Space complexity. We consider the space to store messages in a REUNION deployment using a server as the communication medium, as well as the per-user bandwidth to be transmitted and received. In an epoch with n participants, each user will send one T1 and will receive n T1 messages, and will send and receive n T2 messages and n T3 messages. Therefore, the bandwidth space complexity from a user perspective is $\mathcal{O}(n)$, and on the server is $\mathcal{O}(n^2)$. The storage space complexity on the server is also $\mathcal{O}(n^2)$ with regards to the smaller T2 and T3 messages, while it is $\mathcal{O}(n)$ with regard to the larger T1 messages.

In Table 8.3 we show lower bounds on the per-epoch data volume of messages at various numbers of users with T1 messages defined as 2048 bytes, while T2 and T3 messages are 32 bytes each. The first column shows the amount of data each user needs to download from the server, while the second column shows the amount of data that the server needs to store.

Users	Per-user Bandwidth (n)	Server Storage (n^2)
10	21.1kB	26.9kB
100	211.2kB	844.8kB
1k	2.1MB	66.0MB
10k	21.1MB	6.4GB
100k	211.2MB	640.2GB
1M	2.1GB	64.0TB
10M	21.1GB	6.4PB

Table 8.3: Per-user and total network bandwidth cost estimates per epoch

An application's choice of T1 payload size has a large influence on the per-user bandwidth costs, but despite its larger size has little effect on the storage costs as it is dwarfed by the $\mathcal{O}(n^2)$ cost of the T2 and T3 messages.

8.6 — Implementations

We have implemented and released [REU] the REUNION protocol in Python as Free Libre Open Source Software (FLOSS).

The protocol description in this section details the core REUNION protocol messages and is sufficient for describing the protocol for two or more participants. In order for the REUNION protocol to work on a network mediated by a server or with a broadcast channel, each protocol message must be encapsulated in an outer framing protocol that explicitly links messages as responses to specific messages in rounds of the protocol. REUNION-on-a-LAN uses IP source and destination addresses of the local multicast peers as the outer framing protocol.

8.6.1 – REUNION on Single Point of Failure. An early Golang prototype of REUNION protocol was integrated as part of peer rendezvous in the Katzenpost Mix network [Sta19b] for use with the Catshadow [Sta19a] mix network messaging system. The Katzenpost REUNION server functions as a plugin [AKS17] to the Katzenpost mix server. Clients send Sphinx [ADD+17] packets across the mix network to the REUNION server. The REUNION server anonymously replies to clients by means of a Sphinx Single Use Reply Block (SURB) [DG09] [ADD+17]; that is, clients bundle a SURB in the payload of their Sphinx packets sent to the REUNION server to allow anonymous replies from the server.

8.6.2 – REUNION on a local-area network. The Python implementation [REU] is a variant of the REUNION protocol that we call REUNION-on-a-LAN. It is a serverless design where each client communicates the T1, T2, and T3 protocol messages directly to all of the other participants using UDP packets. The protocol uses multicast IP packets to discover other REUNION peers on the same Ethernet segment. The protocol is run in a pairwise fashion for each discovered peer. Scaling the basic protocol to work over the Internet is largely limited by peer discovery methods, as well as the deployment of network gateways such as Network Address Translation (NAT) devices that break end-to-end discoverability on the Internet.

8.7 — Future Work

Future research remains to replace Curve25519 with a fully post-quantum Diffie-Hellman function that has a defined Elligator map or an equivalent uniformly random representation. Similarly, censorship-resistant publication systems such as DSpool [RS] are promising as replacements to simple servers.

8.7.1 – Post-Quantum considerations. The current design of CSIDH [CLM+18a] lacks an Elligator [BHKL13] construction. If it were possible to map any random string to a valid public key, as is the case with Curve25519 and Elligator, it would be trivial to encrypt both public keys using the shared phrase. The creation of an Elligator style map for CSIDH would change REUNION from a transitionally post-quantum system, one which is certainly quantum-annoying [ES21], to a post-quantum protocol. This is considered to be a difficult problem worth solving [AJK+20].

8.7.2 – Censorship resistance. In the current construction, a server or broadcast medium may behave incorrectly in an undetectable manner that is impossible to distinguish from attempts to rendezvous with participants that use a different passphrase. A single accidental bitflip in a T1 message is sufficient to mount a denial of service attack. If the REUNION implementation's underlying transport is provided by a mixnet such as Katzenpost, an adversary would need to be slightly more careful about their dishonesty. A smart user may query for their own messages to ensure the server is representing their participation in the protocol properly. Ideally, REUNION should be combined with a censorship-resistant system such as DSpool [RS] or some kind of cryptographically secure append-only data structure. In addition to censorship resistance, any improvements to the protocol should be balanced with the needs of participants who wish to use REUNION in an offline setting such as an airgapped local-area network.

8.8 — Conclusions

In this chapter of the thesis we present a refresher about PANDA, introduce a tweaked protocol that we call PANDA′, and introduce a third protocol, REUNION. REUNION as introduced in Section 8.3 improves on PANDA as described in subsection 8.2.1, subsection 8.2.2, and in subection 8.2.4 in a number of ways: REUNION forces interactivity; it uses argon2id rather than scrypt; it mitigates the possibility of precomputation; it removes the need to trust the server for anything other than availability and general censorship; it eliminates passive adversaries' ability to discern who has rendezvoused with whom, or if any rendezvous was successful at all; it allows n-way rendezvous; it requires n-way communication without revealing to any third party observer which pairwise or n-way communication was meaningful; and it does not implicitly link users in a pairwise manner visible to the server or a curious, correctly guessing, actively probing attacker.

Bibliography

[10394] 103rd Congress of the United States. CALEA: The Communications Assistance for Law Enforcement Act, 1994. https://en.wikisource.org/wiki/Communications_Assistance_for_Law_Enforcement_Act_of_1994.

[AAL$^+$05] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS security introduction and requirements. *IETF RFC 4033*, March 2005.

[AAMMZ12] Julian Assange, Jacob Appelbaum, Andy Müller-Maguhn, and Jérémie Zimmermann. Cypherpunks: freedom and the future of the Internet, 2012. https://www.orbooks.com/catalog/cypherpunks/ and https://archive.org/details/pdfy-ekVVZgGOThtG6fXb and magnet:?xt=urn:btih:07FDDDCEF69027BE5F53D3ECCC07CB7A4E662C66.

[AAMS01] Z. Albanna, K. Almeroth, D. Meyer, and M. Schipper. IANA Guidelines for IPv4 Multicast Address Assignments. RFC 3171 (Best Current Practice), August 2001. Obsoleted by RFC 5771 and http://www.ietf.org/rfc/rfc3171.txt.

[AB12] Jean-Philippe Aumasson and Daniel J. Bernstein. Siphash: A fast short-input PRF. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, volume 7668 of *Lecture Notes in Computer Science*, pages 489–508. Springer, 2012. https://eprint.iacr.org/2012/351.

[ABD$^+$15] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In *22nd ACM Conference on Computer and Communications Security*, October 2015. https://weakdh.org/.

[ABG$^+$13] Jacob Appelbaum, Nikolaus Blome, Hubert Gude, Ralf Neukirch, René Pfister, Laura Poitras, Marcel Rosenbach, Jörg Schindler, Gregor Peter

Schmitz, and Holger Stark. Der unheimliche Freund. *Der Spiegel*, 28:20–26, 2013.

[Acc17] Accelerated Development Team. UNITEDRAKE Manual, 2017. https://assets.documentcloud.org/documents/3987443/The-Shaow-Brokers-UNITEDRAKE-Manual.pdf.

[ACD+21] Yawning Angel, Sofía Celi, Claudia Diaz, Ania Piotrowska, David Stainton, and Masala. Katzenpost mix network public key infrastructure specification, 2021. https://github.com/katzenpost/docs/blob/master/specs/pki.rst.

[ACDR20] Gora Adj, Jesús-Javier Chi-Domínguez, and Francisco Rodríguez-Henríquez. On new Vélu's formulae and their applications to CSIDH and B-SIDH constant-time implementations. IACR Cryptol. ePrint Arch., vol. 2020/1109, 2020. https://eprint.iacr.org/2020/1109.

[ACDR21] Gora Adj, Jesús-Javier Chi-Domínguez, and Francisco Rodríguez-Henríquez. SIBC Python library. https://github.com/JJChiDguez/sibc/, 2021.

[Ada14] Andrew A. Adams. Report of a debate on Snowden's actions by ACM members. *SIGCAS Computers and Society*, 44(3):5–7, 2014.

[ADD+17] Yawning Angel, George Danezis, Claudia Diaz, Ania Piotrowska, and David Stainton. Sphinx mix network cryptographic packet format specification. https://github.com/katzenpost/docs/blob/master/specs/sphinx.rst, 2017.

[Age02] Agence France-Presse. Belgium: Apology For Lumumba Killing, 2 2002. https://www.nytimes.com/2002/02/06/world/world-briefing-europe-belgium-apology-for-lumumba-killing.html.

[AGG+14a] Jacob Appelbaum, Aaron Gibson, John Goetz, Volker Kabisch, Lena Kampf, and Leif Ryge. NSA targets the privacy-conscious, 07 2014. https://daserste.ndr.de/panorama/aktuell/nsa230_page-1.html.

[AGG+14b] Jacob Appelbaum, Aaron Gibson, John Goetz, Volker Kabisch, Lena Kampf, and Leif Ryge. NSA XKeyscore source code, 07 2014. https://daserste.ndr.de/panorama/xkeyscorerules100.txt.

[AGG+14c] Jacob Appelbaum, Aaron Gibson, Christian Grothoff, Andy Müller-Maguhn, Laura Poitras, Michael Sontheimer, and Christian Stöcker. FISA intercept of OTR chat, 12 2014. https://www.spiegel.de/media/af108995-0001-0014-0000-000000035552/media-35552.pdf.

[AGG+14d] Jacob Appelbaum, Aaron Gibson, Christian Grothoff, Andy Müller-Maguhn, Laura Poitras, Michael Sontheimer, and Christian Stöcker. Inside the NSA's War on Internet Security, 12 2014.

https://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html.

[AGG+14e] Jacob Appelbaum, Aaron Gibson, Christian Grothoff, Andy Müller-Maguhn, Laura Poitras, Michael Sontheimer, and Christian Stöcker. Inside the NSA's War on Internet Security. https://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html, 12 2014.

[AGG+15a] Jacob Appelbaum, Aaron Gibson, Claudio Guarnieri, Andy Müller-Maguhn, Laura Poitras, Marcel Rosenbach, Leif Ryge, Hilmar Schmundt, and Michael Sontheimer. NSA Preps America for Future Battle. *Der Spiegel*, 1 2015. https://www.spiegel.de/international/world/new-snowden-docs-indicate-scope-of-nsa-preparations-for-cyber-battle-a-1013409.html.

[AGG+15b] Jacob Appelbaum, Aaron Gibson, Claudio Guarnieri, Andy Müller-Maguhn, Laura Poitras, Marcel Rosenbach, Leif Ryge, Hilmar Schmundt, and Michael Sontheimer. NSA QWERTY Malware sample. *Der Spiegel*, 2015. https://www.spiegel.de/media/66ae8d11-0001-0014-0000-000000035668/media-35668.pdf.

[AGK+14] Jacob Appelbaum, Matthias Gebauer, Susanne Koelbl, Laura Poitras, Gordon Repinski, Marcel Rosenbach, and Holger Stark. Obama's Lists: A Dubious History of Targeted Killing in Afghanistan. *Spiegel Online*, 2014. https://www.spiegel.de/international/world/secret-docs-reveal-dubious-details-of-targeted-killings-in-afghanistan-a-1010358.html.

[AHS13] Jacob Appelbaum, Judith Horchert, and Christian Stöcker. Shopping for Spy Gear: Catalog Advertises NSA Toolbox, 12 2013. https://www.spiegel.de/international/world/catalog-reveals-nsa-has-back-doors-for-numerous-devices-a-940994.html.

[AJK+20] Reza Azarderakhsh, David Jao, Brian Koziel, Jason T. LeGrow, Vladimir Soukharev, and Oleg Taraskin. How not to create an isogeny-based pake. Cryptology ePrint Archive, Report 2020/361, 2020. https://eprint.iacr.org/2020/361.

[AKS17] Yawning Angel, Kali Kaneko, and David Stainton. Katzenpost provider-side autoresponder extension. https://github.com/katzenpost/docs/blob/master/specs/kaetzchen.rst, 2017.

[AKSE18] A Abdelrahman, H Khaled, Eman Shaaban, and Wail S Elkilani. WPA-WPA2 PSK cracking implementation on parallel platforms. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pages 448–453. IEEE, 2018.

[AL21] Tomer Ashur and Atul Luykx. *An Account of the ISO/IEC Standardization of the Simon and Speck Block Cipher Families*, pages 63–78. Springer International Publishing, Cham, 2021. https://link.springer.com/chapter/10.1007/978-3-030-10591-4_4.

[Alb16] Jan Philipp Albrecht. How the GDPR will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016. https://edpl.lexxion.eu/data/article/10073/pdf/edpl_2016_03-005.pdf.

[Amb13] Marc Ambinder. An Educated Guess About How the NSA Is Structured, 8 2013. https://www.theatlantic.com/technology/archive/2013/08/an-educated-guess-about-how-the-nsa-is-structured/278697/.

[AMC$^+$14] Bernard D. Aboba, Jouni Malinen, Paul Congdon, Joseph A. Salowey, and Mark Jones. RADIUS Attributes for IEEE 802 Networks. RFC 7268, July 2014. https://rfc-editor.org/rfc/rfc7268.txt.

[AMW19] Jacob Appelbaum, Chloe Martindale, and Peter Wu. Tiny WireGuard Tweak. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje-eddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2019 - 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9-11, 2019, Proceedings*, volume 11627 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2019. https://doi.org/10.1007/978-3-030-23696-0_1.

[And20] Ross Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2020. https://www.cl.cam.ac.uk/~rja14/book.html.

[And21a] Patrick D. Anderson. Mediastan: A Wikileaks Road Movie, 2013, 2021. https://www.tandfonline.com/doi/pdf/10.1080/08821127.2021.1976029.

[And21b] Jensine Andresen. Two Elephants in the Room of Astrobiology. *Astrobiology: Science, Ethics, and Public Policy*, pages 193–231, 2021. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119711186.ch10.

[ANWW13] Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein. BLAKE2: simpler, smaller, fast as MD5. *IACR Cryptology ePrint Archive*, 2013:322, 2013. http://eprint.iacr.org/2013/322.

[AP05] Michel Abdalla and David Pointcheval. Simple password-based encrypted key exchange protocols. In *Topics in cryptology–CT-RSA 2005*, pages 191–208. Springer, 2005.

[APR$^+$13a] Jacob Appelbaum, Laura Poitras, Marcel Rosenbach, Christian Stöcker, Jörg Schindler, and Holger Stark. Documents reveal top NSA hacking unit, December 2013. https://www.spiegel.de/international/world/the-nsa-uses-powerful-toolbox-in-effort-to-spy-on-global-networks-a-940969.html.

[APR$^+$13b] Jacob Appelbaum, Laura Poitras, Marcel Rosenbach, Christian Stöcker, Jörg Schindler, and Holger Stark. Inside TAO: Documents Reveal Top NSA Hacking Unit, 12 2013. https://www.spiegel.de/international/world/the-nsa-uses-powerful-toolbox-in-effort-to-spy-on-global-networks-a-940969.html.

[AR21] Matthieu Aikins and Najim Rahim. Afghan Family Says Errant U.S. Missile Killed 10, Including 7 Children, 8 2021. https://www.nytimes.com/2021/09/10/world/asia/us-air-strike-drone-kabul-afghanistan-isis.html.

[ARAHS19] Mansoor Ahmed-Rengers, Ross Anderson, Darija Halatova, and Ilia Shumailov. Snitches Get Stitches: On The Difficulty of Whistleblowing. In *Cambridge International Workshop on Security Protocols*, pages 289–303. Springer, 2019.

[ARBO04] Götz Aly, Karl Heinz Roth, Edwin Black, and Assenka Oksiloff. *The Nazi census: Identification and control in the Third Reich*, volume 61. Temple University Press, 2004.

[Ass06] Julian Assange. State and Terrorist Conspiracies, 11 2006. https://cryptome.org/0002/ja-conspiracies.pdf.

[Ass07] Julian Assange. On the take and loving it, Academic recipients of the US. intelligence budget, 10 2007. https://wikileaks.org/wiki/On_the_take_and_loving_it.

[Ass11] Julian Assange, 2011. *"The goal is to use Afghanistan to wash money out of the tax bases of the US and Europe through Afghanistan and back into the hands of a transnational security elite. The goal is an endless war, not a successful war*", https://twitter.com/wikileaks/status/1427929346262642688.

[ATE07] B. Aboba, D. Thaler, and L. Esibov. Link-local Multicast Name Resolution (LLMNR). RFC 4795 (Informational), January 2007.

[ATL06] Hedayat Alghassi, Shahram Tafazoli, and Peter Lawrence. The Audio Surveillance Eye. In *2006 IEEE International Conference on Video and Signal Based Surveillance*, pages 106–106, 2006.

[Aus09] Stefan Aust. *Baader-Meinhof: The inside story of the RAF*. Oxford University Press, USA, 2009.

[BÏ7] Jill Bähring. Individual Redress and State Responsibility. The law and politics of compensation payments and their application to drone strike victims, 2017. Vrije Universiteit Amsterdam; Masters thesis.

[BA13] Barton Gellman and Ashkan Soltani. NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say. https://www.washingtonpost.com/world/national-security/2013/10/30/e51d661e-4166-11e3-8b74-d89d714ca4dd_story.html, 10 2013.

[BAG12] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):1–22, 2012.

[Bam82] James Bamford. *The Puzzle Palace: a report on NSA, America's most secret agency*. Houghton Mifflin Harcourt, 09 1982.

[Bam02] James Bamford. *Body of Secrets: Anatomy of the Ultra-secret National Security Agency*. Anchor Books, 2002.

[Bam05] James Bamford. *A Pretext for War: 9/11, Iraq, and the Abuse of America's Intelligence Agencies*. Knopf Doubleday Publishing Group, 2005.

[Bam09] James Bamford. *The Shadow Factory: The Ultra-secret NSA from 9/11 to the Eavesdropping on America*. Anchor Books, 2009.

[Bam12] James Bamford. The NSA is building the country's biggest spy center (watch what you say). *Wired Magazine*, 15, 2012.

[Bam16] James Bamford. A Death in Athens: The Inherent Vulnerability of 'Lawful Intercept', 2016. http://www.sicherheitsforschung-magdeburg.de/uploads/journal/MJS_048_Bamford_DeathinAthens.pdf.

[Bar11] R. Barnes. Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE). RFC 6394 (Informational), October 2011.

[Bar21] Adam Barnes. In bizarre scheme, controversial whistleblower Chelsea Manning invited to Canada just to be thrown out, 10 2021. https://thehill.com/changing-america/enrichment/arts-culture/575927-in-bizarre-scheme-controversial-whistleblower.

[Bau21] Chris Baumohl. Piercing the Veil: Reconciling FISA and the State Secrets Privilege in the Schrems II Era. *Comment, Piercing the Veil: Reconciling FISA and the State Secrets Privilege in the Schrems II Era*, 71, 2021.

[Baz12] Hatem Bazian. Muslims–Enemies of the State: The New Counter-Intelligence Program (COINTELPRO). *Islamophobia Studies Journal*, 1(1):165–206, 2012.

[BBG13] James Ball, Julian Borger, and Glenn Greenwald. Revealed: how US and UK spy agencies defeat Internet privacy and security, 9 2013. https://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security.

[BCRT15] Desmond Ball, Duncan Campbell, Bill Robinson, and Richard Tanter. Expanded communications satellite surveillance and intelligence activities utilising multi-beam antenna systems. *Special Reports, Nautilus Institute*, 2015. `https://core.ac.uk/reader/51344110`.

[BDK16] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 292–302. IEEE, 2016.

[BDL+11] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-Speed High-Security Signatures. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 124–142. Springer, 2011. `https://doi.org/10.1007/978-3-642-23951-9_9`.

[BDLP+15] K. Bhargavan, A. Delignat-Lavaud, A. Pironti, A. Langley, and M. Ray. Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension. RFC 7627 (Proposed Standard), September 2015.

[Ber97] Daniel J. Bernstein. SYN cookies. `http://cr.yp.to/syncookies.html`, 1997.

[Ber02] Daniel J. Bernstein. Dishonest behavior by government lawyers, 2002. `https://cr.yp.to/export/dishonesty.html`.

[Ber03] Daniel J. Bernstein. Bernstein v. United States, 1992–2003. `http://export.cr.yp.to/`.

[Ber05] Daniel J. Bernstein. The Poly1305-AES message-authentication code. In *International workshop on fast software encryption*, pages 32–49. Springer, 2005.

[Ber06] Daniel J. Bernstein. Curve25519: New Diffie-Hellman Speed Records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, April 24-26, 2006, Proceedings*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228. Springer, 2006.

[Ber08a] Daniel J. Bernstein. ChaCha, a variant of Salsa20. In *Workshop record of SASC*, volume 8, pages 3–5, 2008.

[Ber08b] Daniel J. Bernstein. DNSCurve: Usable security for DNS, 08 2008. `http://dnscurve.org/`.

[Ber08c] Daniel J Bernstein. The Salsa20 family of stream ciphers. In *New stream cipher designs*, pages 84–97. Springer, 2008.

[Ber10] Daniel J. Bernstein. High-speed high-security cryptography: encrypting and authenticating the whole Internet. `https://cr.yp.to/talks/2010.12.28/slides.pdf`, 12 2010.

[Ber20] Daniel J. Bernstein. Cryptographic competitions, 12 2020. `https://cr.yp.to/papers.html#competitions`.

[Beu22] Ward Beullens. Breaking Rainbow Takes a Weekend on a Laptop. Cryptology ePrint Archive, Report 2022/214, 2022. `https://ia.cr/2022/214`.

[BG07] U. Blumenthal and P. Goel. Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS). RFC 4785 (Proposed Standard), January 2007.

[BGB04] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use PGP. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 77–84, 2004. `https://otr.cypherpunks.ca/otr-wpes.pdf`.

[BGH$^+$19] Andrea Bittau, Daniel B. Giffin, Mark J. Handley, David Mazieres, Quinn Slack, and Eric W. Smith. Cryptographic Protection of TCP Streams (tcpcrypt). RFC 8548, May 2019. `https://rfc-editor.org/rfc/rfc8548.txt`.

[BHG13] James Ball, Luke Harding, and Juliette Garside. BT and Vodafone among telecoms companies passing details to GCHQ. *The Guardian*, 2, 2013. `https://www.theguardian.com/business/2013/aug/02/telecoms-bt-vodafone-cables-gchq`.

[BHH$^+$15] Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 368–397. Springer, 2015. `https://sphincs.cr.yp.to/`.

[BHK$^+$19] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The SPHINCS+ Signature Framework. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2129–2146, New York, NY, USA, 2019. Association for Computing Machinery. `https://doi.org/10.1145/3319535.3363229`.

[BHKL13] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 967–980. ACM, 2013.

[Bie15] Felix Bieker. Can courts provide effective remedies against violations of fundamental rights by mass surveillance? The case of the United Kingdom. In David Aspinall, Jan Camenisch, Marit Hansen, Simone Fischer-Hübner, and Charles D. Raab, editors, *Privacy and Identity Management. Time for a Revolution? - 10th IFIP WG 9.2, 9.5, 9.6/11.7, 11.4, 11.6/SIG 9.2.2 International Summer School, Edinburgh, UK, August 16-21, 2015, Revised Selected Papers*, volume 476 of *IFIP Advances in Information and Communication Technology*, pages 296–311. Springer, 2015.

[Bio10] Phillipe Biondi. Scapy. http://www.secdev.org/projects/scapy/, 2010.

[BL17] Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017.

[Bla04] Edwin Black. The Nazi Party: IBM & 'Death's Calculator,'. *Jewish Virtual Library*, 2004.

[Bla06] Matt Blaze. Toward a broader view of security protocols. In Bruce Christianson, Bruno Crispo, James A. Malcolm, and Michael Roe, editors, *Security Protocols*, pages 106–120, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Bla12] Edwin Black. *IBM and the Holocaust: The Strategic Alliance Between Nazi Germany and America's Most Powerful Corporation-Expanded Edition*. Dialog press, 2012.

[BLFF96] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945 (Informational), May 1996.

[BLN16] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. Dual EC: A standardized back door. In Peter Y. A. Ryan, David Naccache, and Jean-Jacques Quisquater, editors, *The New Codebreakers - Essays Dedicated to David Kahn on the Occasion of His 85th Birthday*, volume 9100 of *Lecture Notes in Computer Science*, pages 256–281. Springer, 2016. http://projectbullrun.org/dual-ec/index.html.

[BM10] Jason Bau and John Mitchell. A Security Evaluation of DNSSEC with NSEC3. In *Proceedings of the Network and Distributed System Security Symposium, NDSS*, 2010.

[Boa14] Internet Architecture Board. IAB statement on Internet confidentiality, 11 2014. https://mailarchive.ietf.org/arch/msg/ietf-announce/0bCNmWcsFPNTIdMX5fmbuJoKFR8.

[Bon13] Trudy Bond. Dirty wars: Peace psychologists and the need to confront reality: A review of Dirty wars: The world is a battlefield., 2013. https://ethicalpsychology.org/materials/Dirty-Wars.pdf.

[Boo15] Werner Boote. Alles unter Kontrolle, 2015. https://www.imdb.com/title/tt5284200/.

[Bor13a] Julian Borger. NSA files: why the Guardian in London destroyed hard drives of leaked files, 8 2013. https://www.theguardian.com/world/2013/aug/20/nsa-snowden-files-drives-destroyed-london.

[Bor13b] S. Bortzmeyer. Possible solutions to DNS privacy issues. http://tools.ietf.org/html/draft-bortzmeyer-dnsop-privacy-sol-00, December 2013. Accessed: 2017-10-30.

[Bor15] S. Bortzmeyer. DNS Privacy Considerations. RFC 7626 (Informational), August 2015.

[Bor16a] S. Bortzmeyer. DNS Query Name Minimisation to Improve Privacy. RFC 7816 (Experimental), March 2016.

[Bor16b] S. Bortzmeyer. Next step for DPRIVE: resolver-to-auth link. https://tools.ietf.org/html/draft-bortzmeyer-dprive-step-2-01, July 2016. Accessed: 2017-10-30.

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In *International conference on the theory and applications of cryptographic techniques*, pages 139–155. Springer, 2000.

[BPR14] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2014.

[BPT15] M. Belshe, R. Peon, and M. Thomson. Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540 (Proposed Standard), May 2015.

[Bra92] Conny Braam. *Operatie Vula: Zuidafrikanen en Nederlanders in de strijd tegen apartheid*, volume 1284. Meulenhoff, 1992.

[Bra04] Conny Braam. *Operation Vula*. Jacana Media, 2004.

[Bro04] Robert Brown. Who Will Watch the Watchmen? A Response to the Patriot Act. *Reflections*, 4(1), 2004.

[Bro17] Jordan Brown. Stare Into the Lights My Pretties, 2017. https://www.imdb.com/title/tt7762882/.

[Bru11] Finn Brunton. WikiLeaks and the Assange papers. *Radical Philosophy*, 166:8–20, 2011. http://finnb.net/writing/keyspace.pdf.

[BSI09] BSI Group. *Information technology: automatic identification and data capture techniques, QR code 2005 bar code symbology specification*. BSI Group, London, 2009.

[BSJ+15a] R. Barnes, B. Schneier, C. Jennings, T. Hardie, B. Trammell, C. Huitema, and D. Borkmann. Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement. RFC 7624 (Informational), August 2015.

[BSJ+15b] Richard L. Barnes, Bruce Schneier, Cullen Jennings, Ted Hardie, Brian Trammell, Christian Huitema, and Daniel Borkmann. Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement. *RFC*, 7624:1–24, 2015.

[BSSC14] Peter Bergen, David Sterman, Emily Schneider, and Bailey Cahall. *Do NSA's Bulk Surveillance Programs Stop Terrorists?* JSTOR, 2014.

[CAG05] Stuart Cheshire, Bernard Aboba, and Erik Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, May 2005. https://rfc-editor.org/rfc/rfc3927.txt.

[Cai09] Kelly E. Caine. Supporting privacy by preventing misclosure. In Dan R. Olsen Jr., Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Extended Abstracts Volume, Boston, MA, USA, April 4-9, 2009*, pages 3145–3148. ACM, 2009.

[Cam76] Duncan Campbell. The Eavesdroppers. *Time Out*, 1976. https://www.duncancampbell.org/PDF/1976-may-time-out-the-eavesdroppers.pdf.

[Cam79] Duncan Campbell. Official Secrecy and British Libertarianism. *Socialist Register*, 16, 1979. https://socialistregister.com/index.php/srv/article/view/5434.

[Cam82] Duncan Campbell. GCHQ's lost secrets. *New Statesman*, 5, 1982.

[Cam98] Duncan Campbell. Listening in silence. *Index on Censorship*, 27(5):46–53, 1998. https://journals.sagepub.com/doi/pdf/10.1080/03064229808536417.

[Cam99] Duncan Campbell. Interception Capabilities 2000, 4 1999. https://www.duncancampbell.org/menu/surveillance/echelon/IC2000_Report%20.pdf.

[Cam00] Duncan Campbell. Inside Echelon. *Telepolis*, 2000. http://www.heise.de/tp/r4/artikel/6/6929/1.html.

[Cam17] Duncan Campbell. London Internet Exchange members vote no to constitution tweak; Peering peers reject proposed rules on keeping quiet about secret govt gagging orders, 2017. https://www.theregister.com/2017/02/22/linx_members_vote_to_block/.

[Can13] Samuel C Cannon. Terrorizing Wikileaks: Why the embargo against Wikileaks will fail. *J. on Telecomm. & High Tech. L.*, 11:305, 2013.

[Cat19] Catchat team. Catchat. https://github.com/katzenpost/catchat, 2019.

[Cav09] Ann Cavoukian. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada*, 5:12, 2009. https://www.ipc.on.ca/wp-content/uploads/resources/7foundationalprinciples.pdf.

[CBN98] Noam Chomsky, David Barsamian, and Arthur Naiman. *The Common Good*. Real story series. Odonian Press, 1998.

[CCC$^+$19] Daniel Cervantes-Vázquez, Mathilde Chenu, Jesús-Javier Chi-Domínguez, Luca De Feo, Francisco Rodríguez-Henríquez, and Benjamin Smith. Stronger and faster side-channel protections for CSIDH. In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, volume 11774 of *Lecture Notes in Computer Science*, pages 173–193. Springer, 2019. https://doi.org/10.1007/978-3-030-30530-7_9.

[CCTM16] Marco Crocco, Marco Cristani, Andrea Trucco, and Vittorio Murino. Audio Surveillance: A Systematic Review. *ACM Comput. Surv.*, 48(4), feb 2016.

[CGSN20] Jean-Claude Caraco, Rémi Géraud-Stewart, and David Naccache. Kerckhoffs' legacy. Cryptology ePrint Archive, Report 2020/556, 2020. https://ia.cr/2020/556.

[Cim21] Catalin Cimpanu. FBI document shows what data can be obtained from encrypted messaging apps, 11 2021. https://therecord.media/fbi-document-shows-what-data-can-be-obtained-from-encrypted-messaging-apps/.

[CK13a] Stuart Cheshire and Marc Krochmal. DNS-Based Service Discovery. RFC 6763, February 2013. https://rfc-editor.org/rfc/rfc6763.txt.

[CK13b] Stuart Cheshire and Marc Krochmal. Multicast DNS. RFC 6762, February 2013. https://rfc-editor.org/rfc/rfc6762.txt.

[CK13c] Stuart Cheshire and Marc Krochmal. Special-Use Domain Names. RFC 6761 (Proposed Standard), February 2013.

[CLM$^+$18a] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.

[CLM+18b] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH reference implementation source code, 2018. `https://yx7.cc/code/csidh/csidh-latest.tar.xz`.

[CMG+16] Stephen Checkoway, Jacob Maskiewicz, Christina Garman, Joshua Fried, Shaanan Cohney, Matthew Green, Nadia Heninger, Ralf-Philipp Weinmann, Eric Rescorla, and Hovav Shacham. A Systematic Analysis of the Juniper Dual EC Incident. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 468–479, New York, NY, USA, 2016. Association for Computing Machinery.

[CMM02] Russ Cox, Athicha Muthitacharoen, and Robert Morris. Serving dns using a peer-to-peer lookup service. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 155–165, London, UK, UK, 2002. Springer-Verlag.

[CNE+14] Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. On the practical exploitability of Dual EC in TLS implementations. In *23rd USENIX Security Symposium (USENIX) Security 14)*, pages 319–335, 2014.

[Coh01] Bram Cohen. BitTorrent, 2001. `http://www.bittorrent.com/`.

[Coh10] Elliot D Cohen. The Foreign Intelligence Surveillance Court of Review: Purveyor of 'DoubleThink'. In *Mass Surveillance and State Control*, pages 39–45. Springer, 2010.

[Col14] David Cole. Michael Hayden: "we kill people based on metadata". David Cole quoting former director of the CIA Michael Hayden (2014) `https://www.justsecurity.org/10311/michael-hayden-kill-people-based-metadata/`, 5 2014.

[Com73] Committee for Action/Research on the Intelligence Community (Washington, D.C.) and Organizing Committee for a Fifth Estate (Washington, D.C.). *CounterSpy Magazine*. Number Bd. 1 in 1. The Committee, 1973.

[Com75] Church Committee. The FBI, COINTELPRO and Martin Luther King, Jr. *Final Report of the Select Committee to Study Governmental Operations With Respect to Intelligence Activities. Washington, DC: US Congressional Report*, 1975.

[Cou16] U.S. Courts. Wiretap Report 2016. `http://www.uscourts.gov/statistics-reports/wiretap-report-2016`, retrieved on 2017-06-28, 2016.

[CPP10] Andrew Clement, Nancy Paterson, and David J Phillips. IXmaps: Interactively mapping NSA surveillance points in the Internet 'cloud'. *A Global Surveillance Society*, 2010.

[Cra96] Matt Crawford. A Method for the Transmission of IPv6 Packets over Ethernet Networks. RFC 1972 (Proposed Standard), August 1996. Obsoleted by RFC 2464.

[Cra15] Jeremy W Crampton. Collect it all: National security, big data and governance. *GeoJournal*, 80(4):519–531, 2015.

[Cro19] Dave Crocker. DNS Attrleaf Changes: Fixing Specifications That Use Underscored Node Names. RFC 8553, March 2019. https://rfc-editor.org/rfc/rfc8553.txt.

[Cry22] Crypto Mueseum. Breaking the PX-1000Cr, 2 2022. https://www.cryptomuseum.com/crypto/philips/px1000/stef.htm.

[CS20] Joo Yeon Cho and Andrew Sergeev. Post-Quantum MACsec Key Agreement for Ethernet Networks. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, New York, NY, USA, 2020. Association for Computing Machinery. https://dl.acm.org/doi/10.1145/3407023.3409220.

[Cun15] Patrick Cuninghame. Mapping the terrain of struggle: autonomous movements in 1970s Italy. *Viewpoint Magazine*, 1, 2015.

[Cus21] Tim Cushing. Fifth Circuit Says Man Can't Sue Federal Agencies For Allegedly Targeting Him After He Refused To Be An FBI Informant, 12 2021. https://www.techdirt.com/articles/20211031/15594847855/fifth-circuit-says-man-cant-sue-federal-agencies-allegedly-targeting-him-after-he-refused-to-be-fbi-informant.shtml.

[CvdGLK16] C. Contavalli, W. van der Gaast, D. Lawrence, and W. Kumari. Client Subnet in DNS Queries. RFC 7871 (Informational), May 2016.

[CVW02] Ward Churchill and Jim Vander Wall. *Agents of repression: The FBI's secret wars against the Black Panther Party and the American Indian Movement*, volume 7. South End Press, 2002.

[CYp14] Cryptome, John Young, and @paulmd199. Toward the identity of Country X" in MYSTIC, 2014. https://cryptome.org/2014/05/nsa-mystic-identity.pdf.

[DA99] T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), January 1999. Obsoleted by RFC 4346, updated by RFCs 3546, 5746, 6176, 7465, 7507.

[D'A07] Hugh D'Andrade. Qwest CEO: NSA Punished Qwest for Refusing to Participate in Illegal Surveillance–Pre-9/11!, 10 2007. https://www.eff.org/de/deeplinks/2007/10/qwest-ceo-nsa-punished-qwest-refusing-participate-illegal-surveillance-pre-9-11.

[DA12] Suelette Dreyfus and Julian Assange. *Underground: tales of hacking, madness and obsession on the electronic frontier*. Canongate Books, 2012.

[Dam03] Lutz Dammbeck. Das netz: The unabomber. *LSD, and the Internet*, 2003. https://www.imdb.com/title/tt0434231/.

[Dav92] James Kirkpatrick Davis. *Spying on America: The FBI's domestic counterintelligence program*. Greenwood Publishing Group, 1992.

[DBN$^+$01] Morris Dworkin, Elaine Barker, James Nechvatal, James Foti, Lawrence Bassham, E. Roback, and James Dray. Advanced encryption standard (aes), 2001-11-26 2001.

[DC85] S.E. Deering and D.R. Cheriton. Host groups: A multicast extension to the Internet Protocol. RFC 966, December 1985. Obsoleted by RFC 988.

[DC05] George Danezis and Jolyon Clulow. Compulsion resistant anonymous communications. In *International Workshop on Information Hiding*, pages 11–25. Springer, 2005.

[DEH35] DEHOMAG. DEHOMAG poster, 1935. https://www.historyofinformation.com/detail.php?id=617.

[Dev21] Ryan Devereaux. Daniel Hale receives international whistleblower award for drone document leak, 12 2021. https://theintercept.com/2021/12/08/daniel-hale-whistleblower-drone-leak-award/.

[DFM01] Roger Dingledine, Michael J Freedman, and David Molnar. The Free Haven Project: Distributed Anonymous Storage Service. In *Designing Privacy Enhancing Technologies*, pages 67–95. Springer, 2001. https://www.freehaven.net/doc/berk/freehaven-berk.ps.

[DG09] George Danezis and Ian Goldberg. Sphinx: A compact and provably secure mix format. https://cypherpunks.ca/~iang/pubs/Sphinx_Oakland09.pdf, 2009.

[DGP14] Ryan Devereaux, Glenn Greenwald, and Laura Poitras. Data pirates of the Caribbean: The NSA is recording every cell phone call in the Bahamas. *The Intercept*, 19, 2014. https://theintercept.com/2014/05/19/data-pirates-caribbean-nsa-recording-every-cell-phone-call-bahamas/.

[DGR16] S. Dickinson, D. Gillmor, and T. Reddy. Authentication and (d)tls profile for dns-over-tls and dns-over-dtls. http://tools.ietf.org/html/draft-wijngaards-dnsop-confidentialdns-02, 03 2016.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976. https://ee.stanford.edu/~hellman/publications/24.pdf.

[DH98] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), December 1998. Updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112.

[Dic77] Philip K. Dick. *A Scanner Darkly*. Doubleday, 1977.

[Dif83] Whitfield Diffie. Securing Networks: End-to-End Encryption vs. Link Encryption and Trusted Systems. In *Proceedings of the 1983 IEEE Symposium on Security and Privacy, Oakland, California, USA, April 25-27, 1983*, pages 136–138. IEEE Computer Society, 1983. https://doi.org/10.1109/SP.1983.10021.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[DNI21] Zach Dorfman, Sean D. Naylor, and Michael Isikoff. Kidnapping, assassination and a London shoot-out: Inside the CIA's secret war plans against WikiLeaks, 9 2021. https://news.yahoo.com/kidnapping-assassination-and-a-london-shoot-out-inside-the-ci-as-secret-war-plans-against-wiki-leaks-090057786.html.

[Don17a] Jason A. Donenfeld. Wireguard: Next generation kernel network tunnel. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017*. The Internet Society, 2017. https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/wireguard-next-generation-kernel-network-tunnel/.

[Don17b] Jason A. Donenfeld. WireGuard: Next generation kernel network tunnel. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017*. The Internet Society, 2017.

[Don18a] Jason A. Donenfeld. Wireguard Black Hat 2018 talk slides, 2018. see slide 41.

[Don18b] Jason A. Donenfeld. WireGuard: Next generation kernel network tunnel, 2018. version 416d63b 2018-06-30.

[Don19a] Jason A. Donenfeld. Source code for the Go implementation of WireGuard, 2019. commit c2a2b8d739cb.

[Don19b] Jason A. Donenfeld. Source code for the Rust implementation of WireGuard, 2019. commit a7a2e5231571.

[Don19c] Jason A. Donenfeld. WireGuard Android application source, 2019.

[Don19d] Jason A. Donenfeld. WireGuard Linux kernel source, 2019. tag 0.0.20190227, commit ab146d92c353.

[Don19e] Jason A. Donenfeld. WireGuard MacOS and iOS application source, 2019.

[Don19f] Jason A. Donenfeld. WireGuard Windows application source, 2019.

[DP03] James DiEugenio and Lisa Pease. *The assassinations: Probe magazine on JFK, MLK, RFK and Malcolm X*. Feral House, 2003.

[DR02a] Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer-Verlag, Berlin, Heidelberg, 2002.

[DR02b] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer, 2002.

[DR06] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346 (Proposed Standard), April 2006. Obsoleted by RFC 5246, updated by RFCs 4366, 4680, 4681, 5746, 6176, 7465, 7507.

[DR08] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905.

[Dro97] Ralph Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997. `https://rfc-editor.org/rfc/rfc2131.txt`.

[DSL10] Nick Davies, Jonathan Steele, and David Leigh. Iraq war logs: secret files show how us ignored torture. *The Guardian*, 22, 2010.

[Duk14] V. Dukhovni. Opportunistic Security: Some Protection Most of the Time. RFC 7435 (Informational), December 2014.

[Dum14] Dumazet, E. Linux kernel patch: ipv6: Limit mtu to 65575 bytes, 2014.

[Dun01] Neil Dunbar. IPsec networking standards – an overview. *Inf. Sec. Techn. Report*, 6(1):35–48, 2001.

[DVGD96] C. Davis, P. Vixie, T. Goodwin, and I. Dickinson. A Means for Expressing Location Information in the Domain Name System. RFC 1876 (Experimental), January 1996.

[DW01] Susie Day and Laura Whitehorn. Human rights in the United States: The unfinished story of political prisoners and COINTELPRO. *New Political Science*, 23(2):285–297, 2001.

[DW02] Ludo De Witte. *The assassination of Lumumba*. Verso, 2002.

[DY83] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Trans. Information Theory*, 29(2):198–208, 1983.

[DZM05] Dino A Dai Zovi and Shane A Macaulay. Attacking automatic wireless network selection. In *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, pages 365–372. IEEE, 2005. `https://ieeexplore.ieee.org/abstract/document/1495975/`.

[Ear21] Ears and Eyes. List of found surveillance devices, 8 2021. https://earsandeyes.noblogs.org/files/2021/08/list-pictures-en-2021-08.pdf.

[EGA+15] Yves Eudes, Christian Grothoff, Jacob Appelbaum, Monika Ermert, Laura Poitras, and Matthias Wachs. MoreCowBell - Nouvelles révélations sur les pratiques de la NSA, 01 2015. http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html.

[EH95] Dr. Taher Elgamal and Kipp E.B. Hickman. The SSL Protocol. Internet-Draft draft-hickman-netscape-ssl-00, Internet Engineering Task Force, April 1995. Work in Progress.

[EKGBSBA16] Asma El Kissi Ghalleb, Riwa Ben Slamia, and Najoua Essoukri Ben Amara. Contribution to the fusion of soft facial and body biometrics for remote people identification. In *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 252–257, 2016.

[Ele21] Electrospaces. SIGINT Activity Designators (SIGADs), 11 2021. https://www.electrospaces.net/p/sigint.html.

[Ell61] Daniel Ellsberg. Risk, ambiguity, and the Savage axioms. *The quarterly journal of economics*, pages 643–669, 1961.

[Ell72] Daniel Ellsberg. *Papers on the War*. Simon and Schuster, 1972.

[Ell03] Daniel Ellsberg. *Secrets: A memoir of Vietnam and the Pentagon Papers*. Penguin, 2003.

[Ell10] Daniel Ellsberg. Secrecy and national security whistleblowing. *Social Research: An International Quarterly*, 77(3):773–804, 2010.

[Ell14] Daniel Ellsberg. Surveillance, Secrecy, and Democracy. *Juniata Voices*, 14:192, 2014.

[Ell17] Daniel Ellsberg. *The doomsday machine: Confessions of a nuclear war planner*. Bloomsbury Publishing USA, 2017.

[EM13] Justin Elliott and Theodoric Meyer. Claim on 'Attacks Thwarted' by NSA Spreads Despite Lack of Evidence. *ProPublica*, 23, 10 2013.

[EMUM90] C.F. Everhart, L.A. Mamakos, R. Ullmann, and P.V. Mockapetris. New DNS RR Definitions. RFC 1183 (Experimental), October 1990. Updated by RFCs 5395, 5864, 6195, 6895.

[EO06] Per-Erik Eriksson and Björn Odenhammar. VDSL2: Next important broadband technology. *Ericsson Review*, 1:36–47, 2006.

[Erm16] Monika Ermert. Analyse: USA gibt ihre Wächterrolle im DNS ab. https://www.heise.de/newsticker/meldung/Analyse-USA-gibt-ihre-Waechterrolle-im-DNS-ab-3339640.html, October 2016.

[Erw15] Marshall Erwin. The Latest Rules on How Long NSA Can Keep Americans' Encrypted Data Look Too Familiar, 2015. https://www.justsecurity.org/19308/congress-latest-rules-long-spies-hold-encrypted-data-familiar/.

[ES21] Edward Eaton and Douglas Stebila. The "quantum annoying" property of password-authenticated key exchange protocols. Cryptology ePrint Archive, Report 2021/696, 2021. https://eprint.iacr.org/2021/696.

[Eur78] European Court of Human Rights (ECHR). Klass and Others v. Germany, 1978.

[Eur84] European Court of Human Rights (ECHR). Malone v. United Kingdom, 1984.

[Eur87] European Court of Human Rights (ECHR). Leander v. Sweden, 1987.

[Eur06] European Court of Human Rights (ECHR). Weber and Saravia v. Germany, 2006.

[Eur10] European Court of Human Rights (ECHR). Kennedy v. United Kingdom, 2010.

[Eur15] European Court of Human Rights (ECHR). Roman Zakharov v. Russia, 2015.

[Eur16] European Court of Human Rights (ECHR). Szabó and Vissy v. Hungary, 2016.

[Eur18] European Court of Human Rights (ECHR). Big Brother Watch and others v. the United Kingdom, Sep 2018. Applications nos. 58170/13, 62322/14, 24960/15. http://hudoc.echr.coe.int/eng?i=001-186048.

[Fan64] Frantz Fanon. *The wretched of the earth: The handbook for the black revolution that is changing the shape of the world*. Grove press, 1964.

[fANN15] Internet Corporation for Assigned Names and Numbers. Why Top Level Domains Should Not Use Wildcard Resource Records. https://www.icann.org/groups/ssac/documents/sac-015-en, 2015. Accessed: 2017-10-30.

[FC14] Joe FitzPatrick and Miles Crabill. Stupid PCIe Tricks, 2014. https://milescrabill.com/files/playset-pcie.pdf.

[FGM$^+$97] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068 (Proposed Standard), January 1997. Obsoleted by RFC 2616.

[FH05] P. Ford-Hutchinson. Securing FTP with TLS. RFC 4217 (Proposed Standard), October 2005.

[Fit16] Joe FitzPatrick. The Tao of hardware, the Te of implants. *Black Hat, USA*, 2016. https://www.blackhat.com/docs/us-16/materials/us-16-FitzPatrick-The-Tao-Of-Hardware-The-Te-Of-Implants-wp.pdf.

[Fit20] Joseph Fitsanakis. *Redesigning Wiretapping: The Digitization of Communications Interception*. Springer, 2020.

[FK11] Sheila Frankel and Suresh Krishnan. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. RFC 6071, February 2011. https://rfc-editor.org/rfc/rfc6071.txt.

[FKMS20] Scott Fluhrer, Panos Kampanakis, David McGrew, and Valery Smyslov. Mixing Preshared Keys in IKEv2 for Post-quantum Security. draft-ietf-ipsecme-qr-ikev2-11, 2020. https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-qr-ikev2.

[Fle17] Flent Authors. Flent source code and web page. https://flent.org/, 2017.

[Fle19] Alaina Fleming. The Infringement of Rightful Justice, 2019. https://digitalcommons.butler.edu/fys_ww_f2019/2/.

[Fon20] Juan Font. An open source, self-hosted implementation of the Tailscale control server. https://github.com/juanfont/headscale, 2020.

[Fou98] Electronic Frontier Foundation. Cracking DES: Secrets of encryption research, wiretap politics and chip design, 1998.

[Fox15] Fox-IT. Deep dive into QUANTUM INSERT, 4 2015. https://blog.fox-it.com/2015/04/20/deep-dive-into-quantum-insert/.

[Fra17] Nina Franz. Targeted killing and pattern-of-life analysis: weaponised media. *Media, Culture & Society*, 39(1):111–121, 2017.

[Fre] FreeBSD Project. Chapter 8. IPv6 Internals - Jumbo Payload. https://www.freebsd.org/doc/en/books/developers-handbook/ipv6.html#ipv6-jumbo.

[Fri14] Conor Friedersdorf. Edward Snowden's Other Motive for Leaking, 5 2014. https://www.theatlantic.com/politics/archive/2014/05/edward-snowdens-other-motive-for-leaking/370068/.

[FSF90] FSF. *What is Free Software?* Free Software Foundation, 1990. https://www.gnu.org/philosophy/free-sw.html.

[FT14] S. Farrell and H. Tschofenig. Pervasive Monitoring Is an Attack. RFC 7258 (Best Current Practice), May 2014.

[Gal14a] Ryan Gallagher. The inside story of how British spies hacked Belgium's largest telco, 12 2014. https://theintercept.com/2014/12/13/belgacom-hack-gchq-inside-story/.

[Gal14b] Sean Gallagher. NSA's automated hacking engine offers hands-free pwning of the world, 03 2014. http://arstechnica.com/information-technology/2014/03/nsas-automated-hacking-engine-offers-hands-free-pwning-of-the-world/.

[Gal16] Ryan Gallagher. Inside Menwith Hill: The NSA's British Base at the Heart of U.S. Targeted Killing, 2016. https://theintercept.com/2016/09/06/nsa-menwith-hill-targeted-killing-surveillance/.

[Gal18] Ryan Gallagher. How U.K. spies hacked a European ally and got away with it, 2 2018. https://theintercept.com/2018/02/17/gchq-belgacom-investigation-europe-hack/.

[GBC11] Antonio Gramsci, J.A. Buttigieg, and A. Callari. *Prison Notebooks*. Number v. 1 in European perspectives : a series in social thought and cultural criticism. Columbia University Press, 2011.

[GBM+11] Travis Goodspeed, Sergey Bratus, Ricky Melgares, Rebecca Shapiro, and Ryan Speers. Packets in Packets: Orson Welles' In-Band Signaling Attacks for Modern Radios. In David Brumley and Michal Zalewski, editors, *5th USENIX Workshop on Offensive Technologies, WOOT'11, August 8, 2011, San Francisco, CA, USA, Proceedings*, pages 54–61. USENIX Association, 2011. http://static.usenix.org/event/woot11/tech/final_files/Goodspeed.pdf.

[GC16] Jennifer Stisa Granick and Matthew J Craig. Brief of amici curiae computer scientists and technologists in support of plaintiff-appellant Wikimedia and reversal, 02 2016. http://cyberlaw.stanford.edu/files/publication/files/2016.02.24%20Wikimedia%20v.%20NSA%20amicus%20brief%20final%20(1).pdf.

[GE07] R Kelly Garrett and Paul N Edwards. Revolutionary secrets: Technology's role in the South African anti-apartheid movement. *Social Science Computer Review*, 25(1):13–26, 2007.

[Gei16] Peter Geissle. upc keys, 2016. https://haxx.in/upc_keys.c.

[Gel13] Barton Gellman. NSA broke privacy rules thousands of times per year, audit finds. *Washington Post*, 15(08), 2013.

[Gel21] Barton Gellman. *Dark mirror: Edward Snowden and the American surveillance state*. Penguin, 2021.

[Ger19] Mike German. *Disrupt, Discredit, and Divide: How the New FBI Damages Democracy*. The New Press, 2019.

[GG14] Ryan Gallagher and Glenn Greenwald. How the NSA plans to infect 'millions' of computers with malware, 3 2014. `https://theintercept.com/2014/03/12/nsa-plans-infect-millions-computers-malware/`.

[GGG13] Glenn Greenwald, Ryan Gallagher, and Ryan Grim. Top-Secret Document Reveals NSA Spied On Porn Habits As Part Of Plan To Discredit 'Radicalizers', 11 2013. `https://web.archive.org/web/20131127144711/https://www.huffingtonpost.com/2013/11/26/nsa-porn-muslims_n_4346128.html`.

[GH15] John Goetz and Poul-Erik Heilbuth. Terminal F/Chasing Edward Snowden, 2015. `https://www.imdb.com/title/tt4477936/`.

[Gil13] John Gilmore. Re: [Cryptography] Opening Discussion: Speculation on "BULLRUN". `https://www.mail-archive.com/cryptography@metzdowd.com/msg12325.html`, 2013.

[GM13a] Barton Gellman and Greg Miller. 'Black budget' revealed: Top-secret summary details U.S spy network's successes, failures and objectives. `http://www.washingtonpost.com/wp-srv/special/national/black-budget/` and `https://s3.amazonaws.com/s3.documentcloud.org/documents/781537/cbjb-fy13-v1-extract.pdf`, 8 2013.

[GM13b] Glenn Greenwald and Ewen MacAskill. NSA Prism program taps in to user data of Apple, Google and others, 6 2013. `https://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data`.

[GM18] Ryan Gallagher and Henrik Moltke. The Wiretap Rooms. *The Intercept*, 2018. `https://theintercept.com/2018/06/25/att-internet-nsa-spy-hubs/`.

[GME16] Mordechai Guri, Matan Monitz, and Yuval Elovici. USBee: Air-gap covert-channel via electromagnetic emission from USB. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 264–268. IEEE, 2016.

[GMP13] Glenn Greenwald, Ewen MacAskill, and Laura Poitras. Edward Snowden: the whistleblower behind the NSA surveillance revelations, 6 2013. `https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance`.

[GNP$^+$14] Sharon Goldberg, Moni Naor, Dimitrios Papadopoulos, Leonid Reyzin, Sachin Vasant, and Asaf Ziv. NSEC5: Provably Preventing DNSSEC Zone Enumeration. *IACR Cryptology ePrint Archive*, 2014:582, 2014.

[GNU20] GNU Project. glibc: System Databases and Name Service Switch, 1992-2020. https://www.gnu.org/software/libc/manual/html_node/Name-Service-Switch.html.

[Gol18] Jack Goldsmith. The Failure of Internet Freedom, 6 2018. https://knightcolumbia.org/content/failure-internet-freedom.

[Goo21] Google. Supplemental Information on Geofence Warrants in the United States, 2021. https://services.google.com/fh/files/misc/supplemental_information_geofence_warrants_united_states.pdf.

[Gor13] Siobhan Gorman. NSA Officers Spy on Love Interests, 8 2013. http://blogs.wsj.com/washwire/2013/08/23/nsa-officers-sometimes-spy-on-love-interests/.

[Gra13] Jennifer Granick. NSA SEXINT IS THE ABUSE YOU'VE ALL BEEN WAITING FOR, 11 2013. http://cyberlaw.stanford.edu/blog/2013/11/nsa-sexint-abuse-you%E2%80%99ve-all-been-waiting.

[Gra16] Thomas M. Grace. *Kent State: Death and Dissent in the Long Sixties*. Culture and Politics in the Co. University of Massachusetts Press, 2016. https://scholarworks.umass.edu/umpress_kentstate/1/.

[Gre12] Andy Greenberg. *This Machine Kills Secrets: How WikiLeakers, Hacktivists, and Cypherpunks Are Freeing the World's Information*. Random House, 2012.

[Gre13a] Glenn Greenwald. FISA , 6 2013. https://s3.documentcloud.org/documents/709012/verizon.pdf.

[Gre13b] Glenn Greenwald. NSA collecting phone records of millions of Verizon customers daily, 6 2013. https://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order.

[Gre13c] Glenn Greenwald. The crux of the NSA story in one phrase: 'collect it all'. https://www.theguardian.com/commentisfree/2013/jul/15/crux-nsa-collect-it-all, 2013.

[Gre13d] Glenn Greenwald. XKeyscore: NSA tool collects 'nearly everything a user does on the internet', 2013. https://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data.

[Gre14a] Glenn Greenwald. How covert agents infiltrate the Internet to manipulate, deceive, and destroy reputations. *The Intercept*, 2 2014. https://theintercept.com/2014/02/24/jtrig-manipulation/.

[Gre14b] Glenn Greenwald. *No place to hide: Edward Snowden, the NSA, and the US surveillance state*. Macmillan, 2014.

[Gre17] Glenn Greenwald. Obama Killed a 16-Year-Old American in Yemen. Trump Just Killed His 8-Year-Old Sister., 1 2017. https://theintercept.com/2017/01/30/obama-killed-a-16-year-old-american-in-yemen-trump-just-killed-his-8-year-old-sister/.

[Gre20] Friedhelm Greis. Ein Assange-Vertrauter im Visier der CIA, 12 2020. Hardware backdoor found in modified Cryptophone: https://www.golem.de/news/andy-mueller-maguhn-ein-assange-vertrauter-im-visier-der-cia-2012-153049.html.

[Gro96] Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.

[Gro18] Christian Grothoff. The Internet: We deserve a GNU one!, 2018. https://grothoff.org/christian/bern2018.pdf.

[GRS14] James Glanz, Sebastian Rotella, and David E. Sanger. In 2008 Mumbai attacks, piles of spy data, but an uncompleted puzzle. *New York Times*, 21, 2014. https://www.propublica.org/article/mumbai-attack-data-an-uncompleted-puzzle.

[GS13a] Barton Gellman and Ashkan Soltani. NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say, 10 2013. https://www.washingtonpost.com/world/national-security/nsa-infiltrates-links-to-yahoo-google-data-centers-worldwide-snowden-documents-say/2013/10/30/e51d661e-4166-11e3-8b74-d89d714ca4dd_story.html.

[GS13b] Barton Gellman and Ashkan Soltani. NSA tracking cellphone locations worldwide, Snowden documents show. *The Washington Post*, 4:2013, 2013. https://www.washingtonpost.com/world/national-security/nsa-tracking-cellphone-locations-worldwide-snowden-documents-show/2013/12/04/5492873a-5cf2-11e3-bc56-c6ca94801fac_story.html.

[GS14] Barton Gellman and Ashkan Soltani. NSA surveillance program reaches "into the past" to retrieve, replay phone calls. *The Washington Post*, 18, 2014. https://www.washingtonpost.com/world/national-security/nsa-surveillance-program-reaches-into-the-past-to-retrieve-replay-phone-calls/2014/03/18/226d2646-ade9-11e3-a49e-76adc9210f19_story.html.

[GTD15] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering privacy by design reloaded. In *Amsterdam Privacy Conference*, volume 21,

2015. https://software.imdea.org/~carmela.troncoso/papers/Gurses-APC15.pdf.

[Gue09] Guernica magazine. Wikileaks: Murder in Nairobi. Two Wikileaks-related senior human rights activists have been assassinated., 3 2009. https://www.guernicamag.com/wikileaks_murder_in_nairobi/.

[Gut94] Israel Gutman. *Resistance: The Warsaw Ghetto Uprising*. Houghton Mifflin Harcourt, 1994.

[GWE+15] Christian Grothoff, Matthias Wachs, Monika Ermert, Jacob Appelbaum, David Larousserie, Yves Eudes, and Laura Poitras. MoreCowBells: Nouvelles révélations sur les pratiques de la NSA, January 2015.

[GWEA18a] Christian Grothoff, Matthias Wachs, Monika Ermert, and Jacob Appelbaum. Toward secure name resolution on the Internet. *Computers & Security*, 77:694–708, 2018.

[GWEA18b] Christian Grothoff, Matthias Wachs, Monika Ermert, and Jacob Appelbaum. Towards secure name resolution on the Internet. In *NDSS 2017 DNS Privacy Workshop DPRIV17 '17*, pages 1–20, 2018. Version 1.1; Network and Distributed System Security Symposium (NDSS), February 26, 2017, San Diego, CA., USA, NDSS DNS Privacy Workshop ; Conference date: 26-02-2017 Through 03-03-2017.

[Haa11] Jeffrey Haas. *The assassination of Fred Hampton: how the FBI and the Chicago police murdered a Black Panther*. Chicago Review Press, 2011.

[Hel79] Martin E Hellman. "DES will be totally insecure within ten years". *IEEE spectrum*, 16(7):32–40, 1979.

[Hel80] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Information Theory*, 26(4):401–406, 1980.

[Her00] Jeffrey Herbst. Economic incentives, natural resources and conflict in Africa. *Journal of African Economies*, 9(3):270–294, 2000.

[HH15] J. Alex Halderman and Nadia Heninger. Logjam: Diffie-Hellman, discrete logs, the NSA, and you, 2015. https://media.ccc.de/v/32c3-7288-logjam_diffie-hellman_discrete_logs_the_nsa_and_you.

[HJGHB17] Toke Høiland-Jørgensen, Carlo Augusto Grazia, Per Hurtig, and Anna Brunstrom. Flent: The flexible network tester. In *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*, pages 120–125, 2017.

[HL21] Ryan Hübert and Andrew T Little. Kompromat Can Align Incentives but Ruin Reputations. *American Journal of Political Science*, 2021. https://ryanhubert.com/files/hubert-little-kompromat.pdf.

[HNS^{+}20] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Philip R. Zimmermann. Post-quantum WireGuard source code, 2020. https://cryptojedi.org/crypto/data/pqwireguard-20200402.tar.bz2.

[HNS^{+}21] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Philip R. Zimmermann. Post-Quantum WireGuard. In *2021 IEEE Symposium on Security and Privacy (S&P)*, pages 511–528, Los Alamitos, CA, USA, May 2021. IEEE Computer Society. https://doi.ieeecomputersociety.org/10.1109/SP40001.2021.00030.

[Hof99] P. Hoffman. SMTP Service Extension for Secure SMTP over TLS. RFC 2487 (Proposed Standard), January 1999. Obsoleted by RFC 3207.

[Hog15] Mél Hogan. Data flows and water woes: The Utah data center. *Big Data & Society*, 2(2):2053951715592429, 2015.

[Hol13] Ralph Holz. *Empirical analysis of Public Key Infrastructures and investigation of improvements*. PhD thesis, TU Munich, submitted December 2013.

[Hor72] David Horowitz. U.S. Electronic Espionage: A Memoir, 8 1972. https://wikileaks.org/wiki/Perry_Fellwock#Ramparts:_U.S._Electronic_Espionage:_A_Memoir.

[Hor84] C. Hornig. A Standard for the Transmission of IP Datagrams over Ethernet Networks. RFC 894 (INTERNET STANDARD), April 1984.

[How09] John Howard. *Concentration camps on the home front*. University of Chicago Press, 2009.

[HPV^{+}99] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. Point-to-Point Tunneling Protocol (PPTP). RFC 2637 (Informational), July 1999.

[HS13] Amir Herzberg and Haya Shulman. Fragmentation considered poisonous: or one-domain-to-rule-them-all.org. In *IEEE Conference on Communications and Network Security (CNS)*, pages 224–232. IEEE, 2013.

[HS16] Michael Hart-Slattery. Paying the Piper: Monetary Incentives for Federal Whistleblowers. *Fed. Cir. BJ*, 26:447, 2016.

[HT02] B. Haberman and D. Thaler. Unicast-Prefix-based IPv6 Multicast Addresses. RFC 3306 (Proposed Standard), August 2002. Updated by RFCs 3956, 4489, 7371.

[Hu21] Margaret Hu. The Taliban reportedly have control of US biometric devices – a lesson in life-and-death consequences of data privacy, 2021. https://theconversation.com/the-taliban-reportedly-have-control-of-us-biometric-devices-a-lesson-in-life-and-death-consequences-of-data-privacy-166465.

[Hug93] Eric Hughes. A cypherpunk's manifesto. *Crypto anarchy, cyberstates, and pirate utopias*, pages 81–83, 1993.

[HW12] P. Hoffman and W.C.A. Wijngaards. Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC. RFC 6605 (Proposed Standard), April 2012.

[HWF09] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW '09)*, pages 31–42, New York, NY, USA, October 2009. ACM. http://epub.uni-regensburg.de/11919/1/authorsversion-ccsw09.pdf.

[HZH$^+$16] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. Specification for DNS over Transport Layer Security (TLS). RFC 7858 (Proposed Standard), May 2016.

[IEE06] IEEE. IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security. *IEEE Std 802.1AE-2006*, pages 1–150, 2006.

[IEE09] IEEE. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Protected Management Frames. *IEEE Std 802.11w-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, and IEEE Std 802.11y-2008)*, pages 1–111, 2009.

[Int14] Internet Systems Consortium, Inc. (ISC). Join The Global Passive DNS (pDNS) Network Today & Gain Effective Tools To Fight Against Cyber Crime, 11 2014. https://repository.root-me.org/R%C3%A9seau/EN%20-%20Join%20the%20global%20Passive%20DNS.pdf.

[Int16] Privacy International. The President's Men? Inside The Technical Research Department, The Secret Player In Egypt's Intelligence Infrastructure, 2016. https://privacyinternational.org/report/666/presidents-men-inside-technical-research-department-secret-player-egypts-intelligence.

[Int21a] Amnesty International. Forensic Methodology Report: How to Catch NSO Group's Pegasus, 7 2021. https://www.amnesty.org/en/documents/doc10/4487/2021/en/.

[Int21b] Amnesty International. Uncovering the Iceberg: The Digital Surveillance Crisis Wrought by States and the Private Sector, 7 2021. https://www.amnesty.org/en/documents/doc10/4491/2021/en/.

[Isi08] Michael Isikoff. The Fed who blew the whistle. *Newsweek*, 12 2008.

[Jac20] Bart Jacobs. Maximator: European signals intelligence cooperation, from a Dutch perspective. *Intelligence and National Security*, 35(5):659–668, 2020.

[JAKJ19] Amir Jalali, Reza Azarderakhsh, Mehran Mozaffari Kermani, and David Jao. Towards optimized and constant-time CSIDH on embedded devices. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 215–231. Springer, 2019.

[Jam09] Joy James. *6 Framing the Panther: Assata Shakur and Black Female Agency*, pages 138–160. New York University Press, 2009.

[JAS13] Marcel Rosenbach Jacob Appelbaum, Holger Stark and Jörg Schindler. Berlin Complains: Did US Tap Chancellor Merkel's Mobile Phone?, 2013. `http://www.spiegel.de/international/world/merkel-calls-obama-over-suspicions-us-tapped-her-mobile-phone-a-929642.html`.

[Jay21] Mark M Jaycox. No Oversight, No Limits, No Worries: A Primer on Presidential Spying and Executive Order 12,333. *Harvard National Security Journal*, 12:58, 2021.

[Jen87] Tim Jenkin. *Escape from Pretoria*. Kliptown Books London, 1987.

[Jen95] Tim Jenkin. Talking to Vula. The Story of the Secret Underground Communications Network of Operation Vula. In *Mayibuye: Journal of the African National Congress*, volume 6, 1995.

[Joh90] David Johnston. CIA Tie Reported in Mandela Arrest. *New York Times*, 10(06), 1990.

[Joh98] Thomas R. Johnson. *American cryptology during the cold war, 1945–1989, book III: retrenchment and reform, 1972–1980*. United States National Security Agency, 1998. `https://archive.org/details/cold_war_iii-nsa`.

[Joh14] John Brooks. Anonymous peer-to-peer instant messaging, 03 2014. `https://github.com/ricochet-im/ricochet` `https://www.ricochet.im/`.

[Jul06] Julian Assange. Conspiracy as governance, 12 2006. `https://cryptome.org/0002/ja-conspiracies.pdf`.

[Jul15] Julian Assange and Dan Beeton and Phyllis Bennis and Michael Busch and Peter Certo and Conn Hallinan and Sarah Harrison and Richard Heydarian and Dahr Jamail and Jake Johnston and Alexander Main and Robert Naiman and Francis Njubi Nesbitt and Linda Pearson and Gareth Porter and Tim Shorrock and Russ Welle and Stephen Zunes. *The WikiLeaks files: the world according to US empire*. Verso Books, 2015.

[Kac95] Theodore John Kaczynski. Industrial society and its future, 1995. `http://web.cecs.pdx.edu/~harry/ethics/Unabomber.pdf`.

[Kah96] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Simon and Schuster, 1996.

[Kah02] David Kahn. Remarks of David Kahn Commemorating the 50th Anniversary of the National Security Agency, 11 2002. `https://irp.fas.org/eprint/kahn.html`.

[Kam20] Panos Kampanakis. Configuring post-quantum MACsec in cisco switches, 2020. `https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/configuring-post-quantum-macsec-in-cisco-switches.pdf`.

[Kat] Katzenpost team. Katzenpost mix network. `https://github.com/katzenpost/`.

[Kat90] Ahmed Kathrada. New terrain of struggle. *Indicator South Africa*, 7(2):11–12, 1990.

[KB77] Leonard V Kaplan and Susan Bittker. The Intelligence Network: A Clear and Present Danger. *Human Rights*, pages 135–154, 1977.

[KB09] Mark Klein and James Bamford. *Wiring Up the Big Brother Machine–and Fighting it*. BookSurge, 2009.

[KCBSN21] Lauren Krenzel, Thea Chaloner, Bridget Bentz, and Molly Seavy-Nesper. Documentary Exposes How The FBI Tried To Destroy MLK With Wiretaps, Blackmail, 2021. `https://www.npr.org/2021/01/18/956741992/documentary-exposes-how-the-fbi-tried-to-destroy-mlk-with-wiretaps-blackmail`.

[KE10a] H. Krawczyk and P. Eronen. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). IETF internet draft – RFC5869, IBM Research, May 2010. `https://tools.ietf.org/html/rfc5869`.

[KE10b] Hugo Krawczyk and Pasi Eronen. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). *RFC*, 5869:1–14, 2010.

[Ken11] Emily C Kendall. Guy Fawkes's Dangerous Remedy: The Unconstitutionality of Government-Ordered Assassination against US Citizens and Its Implications for Due Process in America. *J. Marshall L. Rev.*, 45:1121, 2011.

[Ker83] Auguste Kerckhoffs. *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef*. Librairie militaire de L. Baudoin, 1883.

[KGE⁺14] Julian Kirsch, Christian Grothoff, Monika Ermert, Jacob Appelbaum, Laura Poitras, Henrik Moltke, and Andreas Enge. Le programme HACIENDA, August 2014. Version légèrement remaniée de la première publication à http://heise.de/-2293122.

[KHAS21] Azmat Khan, Lila Hassan, Sarah Almukhtar, and Rachel Shorey. The Civilian Casuality Files, 12 2021. `https://www.nytimes.com/interactive/2021/us/civilian-casualty-files.html`.

[Kic] Russ Kick. CounterSpy: 27 Issues of the Infamous Magazine That Exposed the CIA (& Others). `http://altgov2.org/counterspy/`.

[KM10] Srinivas Krishnan and Fabian Monrose. Dns prefetching and its privacy implications: When good things go bad. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET'10, Berkeley, CA, USA, 2010. USENIX Association.

[KM13] Neal Koblitz and Alfred Menezes. Another look at HMAC. *J. Math. Cryptol.*, 7(3):225–251, 2013.

[KM19] Neal Koblitz and Alfred Menezes. Critical perspectives on provable security: Fifteen years of "another look" papers. *Adv. Math. Commun.*, 13(4):517–558, 2019.

[KN11] Cecilia Kang and Ellen Nakashima. Google says hackers based in China accessed U.S. officials' Gmail accounts, 6 2011. `https://www.washingtonpost.com/business/technology/google-says-hackers-based-in-china-accessed-us-officials-gmail-accounts/2011/06/01/AGwgRmGH_story.html`.

[KNT20] Nadim Kobeissi, Georgio Nicolas, and Mukesh Tiwari. Verifpal: Cryptographic Protocol Analysis for the Real World. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 151–202. Springer, 2020. `https://doi.org/10.1007/978-3-030-65277-7_8`.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[Kob19] Nadim Kobeissi. Verifpal: Cryptographic Protocol Analysis for Students and Engineers. *IACR Cryptol. ePrint Arch.*, 2019:971, 2019. `https://eprint.iacr.org/2019/971`.

[Koh90] Harold Hongju Koh. *The national security constitution: Sharing power after the Iran-Contra Affair*. Yale University Press, 1990.

[Koh13] Harold Hongju Koh. Targeted killing could be the most legal way to conduct warfare, Jun 2013. `https://verfassungsblog.de/targeted-killing-could-be-the-most-legal-way-to-conduct-warfare/`.

[Kot10] Philip Kotler. The prosumer movement. In *Prosumer Revisited*, pages 51–60. Springer, 2010.

[Kra05] Hugo Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol. Cryptology ePrint Archive, Report 2005/176, 2005. `http://eprint.iacr.org/2005/176`.

[Kri18] Alexei Krivolap. The glass man identity created by normative virtuality. *WEASA*, 2018.

[Kri19] Sankaran Krishna. Manhunt Presidency: Obama, race, and the Third World. *Third World Quarterly*, 40(2):284–297, 2019.

[Lan09] Adam Langley. Opportunistic encryption everywhere. *In W2SP*, 2009. `https://www.ieee-security.org/TC/W2SP/2009/papers/s1p2.pdf`.

[Lan12a] Adam Langley. Panda, 2012. `https://www.github.com/agl/panda/`.

[Lan12b] Adam Langley. Pond, 2012. `https://www.github.com/agl/pond/`.

[Lan13] Susan Landau. Making sense from Snowden: What's significant in the NSA surveillance revelations. *IEEE Security & Privacy*, 11(4):54–63, 2013.

[Lan14] Susan Landau. Highlights from making sense of Snowden, part II: what's significant in the NSA revelations. *IEEE Security & Privacy*, 12(1):62–64, 2014.

[Lan18a] Susan Landau. The Second Crypto War—What's Different Now. In *27th USENIX Security Symposium (USENIX Security 18)*, Baltimore, MD, August 2018. USENIX Association.

[Lan18b] Klaus Landefeld. G10, BND-Gesetz und der effektive Schutz vor Grundrechten, 2018. `https://fahrplan.events.ccc.de/congress/2018/Fahrplan/events/10016.html`.

[Lar13] Jeff Larson. Revealed: The NSA's Secret Campaign to Crack, Undermine Internet Security, 9 2013. `https://www.propublica.org/article/the-nsas-secret-campaign-to-crack-undermine-internet-encryption`.

[LCBM15] K. Lynn, S. Cheshire, M. Blanchet, and D. Migault. Requirements for Scalable DNS-Based Service Discovery (DNS-SD) / Multicast DNS (mDNS) Extensions. RFC 7558 (Informational), July 2015.

[LCM^{+}16] A. Langley, W. Chang, N. Mavrogiannopoulos, J. Strombergson, and S. Josefsson. ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS). RFC 7905 (Proposed Standard), June 2016.

[Lev01] Steven Levy. *Crypto: How the code rebels beat the government–saving privacy in the digital age*. Penguin, 2001.

[Lin20] Linux man-pages project. *capabilities(7) - Linux manual page*, 7 2020.

[LM94] David Martin Luebke and Sybil Milton. Locating the victim: An overview of census-taking, tabulation technology, and persecution in Nazi Germany. *IEEE Annals of the History of Computing*, 16(3):25, 1994.

[LM13] Daniel Le Métayer. Privacy by Design: A Formal Framework for the Analysis of Architectural Choices. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*, CODASPY '13, pages 95–104, New York, NY, USA, 2013. ACM. http://doi.acm.org/10.1145/2435349.2435361.

[LMV15] Eduardo Novella Lorente, Carlo Meijer, and Roel Verdult. Scrutinizing WPA2 password generating algorithms in wireless routers. In *9th USENIX Workshop on Offensive Technologies (WOOT-15)*, 2015.

[Loe] Avi Loeb. The Galileo Project for the Systematic Scientific Search for Evidence of Extraterrestrial Technological Artifacts at Harvard University. https://projects.iq.harvard.edu/galileo.

[LS92] Martin A Lee and Bruce Shlain. *Acid dreams: The complete social history of LSD: The CIA, the sixties, and beyond*. Grove Press, 1992.

[Lya16] Sarah Lyall. Laura Poitras Prepares 'Astro Noise' for the Whitney Museum, 1 2016. https://www.nytimes.com/2016/01/31/arts/design/laura-poitras-prepares-astro-noise-for-the-whitney-museum.html.

[Lyo03] David Lyon. *Surveillance as Social Sorting: Privacy, Risk, and Digital Discrimination*. Sociology / information and communication studies. Routledge, 2003.

[Lyo07] David Lyon. *Surveillance Studies: An Overview*. Wiley, 2007.

[Lyo09] David Lyon. *Identifying citizens: ID cards as surveillance*. Polity, 2009.

[Lyo14] David Lyon. Surveillance, Snowden, and Big Data: Capacities, consequences, critique. *Big Data & Society*, 1(2):2053951714541861, 2014. https://doi.org/10.1177/2053951714541861.

[MA21] Sarah Mainwaring and Richard J Aldrich. The secret empire of signals intelligence: GCHQ and the persistence of the colonial presence. *The International History Review*, 43(1):54–71, 2021. https://www.tandfonline.com/doi/full/10.1080/07075332.2019.1675082.

[Mac21] Alan Macleod. OTF – The "Independent" Internet Freedom Organization That Makes All Your Favorite Privacy Apps – is Staffed Full of Spies, 12 2021. https://www.mintpressnews.com/the-open-technology-fund-makes-privacy-apps-staffed-spies/279147/.

[Mad98] Deborah L Madsen. *American exceptionalism*. Univ. Press of Mississippi, 1998.

[Mad13] Wayne Madsen. *National security agency surveillance: Reflections and revelations 2001-2013*. Lulu, 2013.

[Mag90] Keith Maguire. The Intelligence War in Northern Ireland. *International Journal of Intelligence and Counter Intelligence*, 4(2):145–165, 1990.

[Mai20] Sarah Mainwaring. Division D: Operation Rubicon and the CIA's secret SIGINT empire. *Intelligence and National Security*, 35(5):623–640, 2020. https://www.tandfonline.com/doi/abs/10.1080/02684527.2020.1774854.

[Man16] Chelsea Manning. Manning v. Clapper (1:16-cv-02307), 2016. https://www.courtlistener.com/docket/6086328/manning-v-clapper/#entry-1.

[Mar06] Christopher D Martin. Ernest Hemingway: a psychological autopsy of a suicide. *Psychiatry: Interpersonal and Biological Processes*, 69(4):351–361, 2006.

[Mar12] Moxie Marlinspike. Cloud crack, 2012. https://web.archive.org/web/20160319232206/https://www.cloudcracker.com/.

[Mar15] Gary T Marx. Surveillance studies. *International encyclopedia of the social & behavioral sciences*, 23(2):733–741, 2015. http://web.mit.edu/gtmarx/www/surv_studies.pdf.

[Mar18] Lerone Martin. Bureau Clergyman: How the FBI Colluded with an African American Televangelist to Destroy Dr. Martin Luther King, Jr. *Religion and American Culture*, 28(1):1–51, 2018.

[Mar22] Stefan Marsiske. pocorgtfo 21 12 apocrypha, 2 2022. https://www.ctrlc.hu/~stef/blog/posts/pocorgtfo_21_12_apocrypha.html and https://github.com/stef/px1000cr.

[Mas98] L. Masinter. Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0). RFC 2324 (Informational), April 1998. Updated by RFC 7168.

[Mas14] Mike Masnick. Parallel Construction Revealed: How The DEA Is Trained To Launder Classified Surveillance Info. https://www.techdirt.com/articles/20140203/11143926078/parallel-construction-revealed-how-dea-is-trained-to-launder-classified-surveillance-info.shtml and https://www.muckrock.com/foi/united-states-of-america-10/dea-policies-on-parallel-construction-6434/, 2 2014.

[Mau18] Maurice Chiodo and Piers Bursill-Hall. Four levels of ethical engagement. *Ethics in Mathematics Discussion Papers*, 1, 2018. https://ethics.maths.cam.ac.uk/assets/dp/18_1.pdf.

[MB76] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Commun. ACM*, 19(7):395–404, July 1976.

[McC13] John McClurg. Comment on Der Spiegel Article Regarding NSA TAO Organization, 12 2013. https://web.archive.org/web/20140102181314/http://en.community.dell.com/dell-blogs/direct2dell/b/direct2dell/archive/2013/12/30/comment-on-der-spiegel-article-regarding-nsa-tao-organization.aspx.

[McK19] Nick McKerrell. "Spycops" in Scotland: Matilda Gifford's Judicial Review – No Right to the Truth? *Edinburgh Law Review*, 23(2):253–258, 2019.

[McL70] Marshall McLuhan. *Culture Is Our Business*. McGraw-Hill, 1970.

[MD05] Steven J Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 183–195. IEEE, 2005.

[MDK14] Bodo Möller, Thai Duong, and Krzysztof Kotowicz. This POODLE bites: exploiting the SSL 3.0 fallback, 09 2014. https://www.openssl.org/~bodo/ssl-poodle.pdf.

[Med14] Betty Medsger. *The Burglary: The Discovery of J. Edgar Hoover's Secret FBI*. Vintage, 2014.

[Mei68] Ulrike Meinhof. Vom Protest zum Widerstand, 5 1968. https://ghdi.ghi-dc.org/sub_document.cfm?document_id=895.

[Mei15] Andre Meister. How the German Foreign Intelligence Agency BND tapped the Internet Exchange Point DE-CIX in Frankfurt, since 2009, 2015. https://netzpolitik.org/2015/how-the-german-foreign-intelligence-agency-bnd-tapped-the-internet-exchange-point-de-cix-in-frankfurt-since-2009/.

[Men13] Joseph Menn. Secret contract tied NSA and security industry pioneer. *Reuters, December*, 13, 2013. https://www.reuters.com/article/us-usa-security-rsa-idUSBRE9BJ1C220131220.

[Men14] Joseph Menn. Exclusive: Nsa infiltrated rsa security more deeply than thought—study. *Reuters, Mar*, 2014. https://www.reuters.com/article/us-usa-security-nsa-rsa-idUSBREA2U0TY20140331.

[Mer78] Ralph C Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978. https://www.merkle.com/1974/PuzzlesAsPublished.pdf.

[MH11] Alexander Meleagrou-Hitchens. *As American as apple pie: How Anwar al-Awlaki became the face of Western jihad*. International Centre for the Study of Radicalisation and Political Violence, 2011.

[Mia21] Malik Miah. United States: New revelations lead to calls for probe into Malcolm X assassination. *Green Left Weekly*, 1(1299):17, 2021.

[Mic14] James Mickens. This world of ours. *USENIX; login: logout*, pages 8–11, 2014. https://www.usenix.org/system/files/1401_08-12_mickens.pdf.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[Mil97] Sybil Milton. Registering civilians and aliens in the Second World War. *Jewish History*, pages 79–87, 1997.

[Mil06] Mark Samuel Miller. *Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control*. PhD thesis, Johns Hopkins University, Baltimore, Maryland, USA, May 2006.

[Min76] Ministeriums für Staatssicherheit der DDR. Die Anwendung von Maßnahmen der Zersetzung, 1976. https://www.stasi-unterlagen-archiv.de/assets/bstu/content_migration/DE/Wissen/MfS-Dokumente/Downloads/Grundsatzdokumente/richtlinie-1-76_ov.pdf.

[MKR$^+$96] Robert Moskowitz, Daniel Karrenberg, Yakov Rekhter, Eliot Lear, and Geert Jan de Groot. Address Allocation for Private Internets. RFC 1918, February 1996. https://rfc-editor.org/rfc/rfc1918.txt.

[MM18] Andy Müller-Maguhn. Cryptophone IP19: Special Edition courtesy of CIA/NSA delivery and implant service, 2018. https://buggedplanet.info/lost+found/20180323.

[Moc87] Paul Mockapetris. Domain names - implementation and specification. RFC 1035, November 1987. https://rfc-editor.org/rfc/rfc1035.txt.

[Moc89] P.V. Mockapetris. DNS encoding of network names and other types. RFC 1101, April 1989.

[Mod08] Debra A Moddelmog. Telling Stories from Hemingway's FBI File: Conspiracy, Paranoia, and Masculinity. In *Modernism on File*, pages 53–72. Springer, 2008.

[MRG$^+$19] A. Theodore Markettos, Colin Rothwell, Brett F. Gutstein, Allison Pearce, Peter G. Neumann, Simon W. Moore, and Robert N. M. Watson. Thunderclap: Exploring Vulnerabilities in Operating System IOMMU Protection via DMA from Untrustworthy Peripherals. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, 2 2019.

[Mul] Mullvad. mullvad-wg-establish-psk. source code post `https://github.com/mullvad/oqs-rs/tree/master/mullvad-wg-establish-psk`.

[Mul17] Mullvad. Introducing a post-quantum VPN, Mullvad's strategy for a future problem. `https://mullvad.net/en/blog/2017/12/8/introducing-post-quantum-vpn-mullvads-strategy-future-problem/`, 12 2017.

[Mun20] Luke Munn. Machine Readable Race: Constructing Racial Information in the Third Reich. *Open Information Science*, 4(1):143–155, 2020.

[MvOV96] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. `http://cacr.uwaterloo.ca/hac/`.

[MWH$^+$14] Allison Mankin, Duane Wessels, John Heidemann, Liang Zhu, and Zi Hu. t-DNS: DNS over TCP and TLS, 12 2014. `https://ant.isi.edu/tdns/`.

[Nak01] Toshiyuki Nakagaki. Smart behavior of true slime mold in a labyrinth. *Research in Microbiology*, 152(9):767–770, 2001.

[Nak08a] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. `https://bitcoin.org/en/bitcoin-paper`.

[Nak08b] Ellen Nakashima. Travelers' Laptops May Be Detained At Border. *Washington Post*, 1, 2008. `https://www.washingtonpost.com/wp-srv/content/article/2008/08/01/laptops.html`.

[Nak13a] Ellen Nakashima. Chinese hackers who breached Google gained access to sensitive data, U.S. officials say, 5 2013. `https://www.washingtonpost.com/world/national-security/chinese-hackers-who-breached-google-gained-access-to-sensitive-data-us-officials-say/2013/05/20/51330428-be34-11e2-89c9-3be8095fe767_story.html`.

[Nak13b] Ellen Nakashima. Chinese hackers who breached google gained access to sensitive data, u.s. officials say. `https://www.washingtonpost.com/world/national-security/2013/05/20/51330428-be34-11e2-89c9-3be8095fe767_story.html`, 5 2013.

[ND16] George Nomikos and Xenofontas A. Dimitropoulos. traIXroute: Detecting IXPs in traceroute paths. In Thomas Karagiannis and Xenofontas A. Dimitropoulos, editors, *Passive and Active Measurement - 17th International Conference, PAM 2016, Heraklion, Greece, March 31 - April 1, 2016. Proceedings*, volume 9631 of *Lecture Notes in Computer Science*, pages 346–358. Springer, 2016. `https://doi.org/10.1007/978-3-319-30505-9_26`.

[New99] C. Newman. Using TLS with IMAP, POP3 and ACAP. RFC 2595 (Proposed Standard), June 1999. Updated by RFCs 4616, 7817.

[New21] New York Times Magazine. The Human Toll of America's Air Wars: Airstrikes allowed America to wage war with minimal risk to its troops., 12 2021. `https://www.nytimes.com/2021/12/19/magazine/victims-airstrikes-middle-east-civilians.html`.

[Nex16] Nex. Everything we know of NSA and Five Eyes malware, 5 2016. `https://tiny.one/nsa-five-eyes-malware`.

[NIS21] NIST. Post-Quantum Round 3 Submissions. `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions`, 2021.

[Nix13] Ron Nixon. U.S. Postal Service Logging All Mail for Law Enforcement, 7 2013. `https://www.nytimes.com/2013/07/04/us/monitoring-of-snail-mail.html`.

[NL15] Y. Nir and A. Langley. ChaCha20 and Poly1305 for IETF Protocols. RFC 7539 (Informational), May 2015.

[NL18] Yoav Nir and Adam Langley. ChaCha20 and Poly1305 for IETF Protocols. *RFC*, 8439:1–46, 2018.

[NSA05] NSA. (U//FOUO) Standardizing Our SIGAD/PDDG Naming Conventions, 3 2005. `https://cyberwar.nl/d/fromTheIntercept/sidtoday/documents/2005/2005-03-09_SIDToday_-_Standardizing_Our_SIGADPDDG_Naming_Conventions.pdf`.

[NYE17] Nir Nissim, Ran Yahalom, and Yuval Elovici. USB-based attacks. *Computers & Security*, 70:675–688, 2017.

[Oez09] Veysel Oezer. The evil karmetasploit upgrade. *Nullcon, Zuri, India*, 2009.

[oGOSoGIR81] United States. Congress. House. Committee on Government Operations. Subcommittee on Government Information and Individual Rights. *The Government's Classification of Private Ideas: Hearings Before a Subcommittee of the Committee on Government Operations, House of Representatives, Ninety-sixth Congress, Second Session, February 28, March 20, and August 21, 1980*. U.S. Government Printing Office, 1981.

[Ope13] Open Technology Fund. Internet Access and Openness: Myanmar 2012, 2 2013. https://web.archive.org/web/20151016011526/https://www.opentech.fund/files/reports/otf_myanmar_access_openness_public.pdf.

[Oss14] Michael Ossmann. The NSA Playset: RF Retroreflectors. *DEF CON*, 22(8), 2014.

[Owe12] Major Dave Owen. A review of intelligence oversight failure: NSA programs that affected Americans. *Military Intelligence*, pages 33–39, 2012. https://irp.fas.org/agency/army/mipb/2012_04-owen.pdf.

[PA21] Raluca Posteuca and Tomer Ashur. How to backdoor a cipher. Cryptology ePrint Archive, Report 2021/442, 2021. https://ia.cr/2021/442.

[Pag09] Trevor Paglen. *Blank spots on the map: the dark geography of the Pentagon's secret world*. Penguin, 2009.

[Pag10] Trevor Paglen. *I Could Tell You But Then You Would Have to be Destroyed by Me: Emblems from the Pentagon's Black World*. Melville House Pub, 2010.

[Pai17] Elisabeth Pain. A top mathematician joins the Macron revolution, 2017. https://www.science.org/doi/full/10.1126/science.356.6344.1223.

[Pap18] Vasileios Papageorgiou. The Greek wiretapping scandal and the false promise of intelligence cooperation in the information era, 2018. https://kedisa.gr/wp-content/uploads/2019/06/Research_no_30_Papageorgiou.pdf.

[Par15] Lisa Parks. Cover Your Webcam: Unencrypting Laura Poitras's Citizenfour. *Film Quarterly*, 68(3):11–16, 2015.

[Pau03] Jacques Pauwels. Profits über Alles!: American Corporations and Hitler. *Labour/Le Travailleur*, 51:223–250, 2003.

[PB14] Privacy and Civil Liberties Oversight Board. "Report on the Surveillance Program Operated Pursuant to Section 702 of the Foreign Intelligence Surveillance Act", 2014. https://www.pclob.gov/library/702-Report.pdf, retrieved on 2017-06-30.

[Pen16] Jonathon W. Penney. Chilling effects: Online surveillance and wikipedia use. *Berkeley Tech. L.J.. Berkeley Technology Law Journal*, 31(IR):117, 2016. http://lawcat.berkeley.edu/record/1127413.

[Per11] Simon Perreault. vCard Format Specification. RFC 6350, August 2011. https://rfc-editor.org/rfc/rfc6350.txt.

[Per12] Colin Percival. The scrypt password-based key derivation function. https://tools.ietf.org/html/draft-josefsson-scrypt-kdf-01, 2012.

[Per18] Trevor Perrin. The Noise Protocol Framework, 2018. http://www.noiseprotocol.org/noise.html.

[Pet13a] Andrea Peterson. A CEO who resisted NSA spying is out of prison. And he feels 'vindicated' by Snowden leaks, 9 2013. https://www.washingtonpost.com/news/the-switch/wp/2013/09/30/a-ceo-who-resisted-nsa-spying-is-out-of-prison-and-he-feels-vindicated-by-snowden-leaks/.

[Pet13b] Andrea Peterson. LOVEINT: When NSA officers use their spying power on love interests, 08 2013. https://www.washingtonpost.com/blogs/the-switch/wp/2013/08/24/loveint-when-nsa-officers-use-their-spying-power-on-love-interests/.

[Pet16] Andrea Peterson. Yahoo scanned all of its users' incoming emails on behalf of U.S. intelligence officials, 10 2016. https://www.washingtonpost.com/news/the-switch/wp/2016/10/04/yahoo-scanned-all-of-its-users-incoming-emails-on-behalf-of-u-s-intelligence-officials/.

[Pil11] Ed Pilkington. WikiLeaks: US opens grand jury hearing, 2011. https://www.theguardian.com/media/2011/may/11/us-opens-wikileaks-grand-jury-hearing.

[Pix09] Carrie Pixler. Setting the Boundaries of the Census Clause: Normative and Legal Concerns Regarding the American Community Survey. *Wm. & Mary Bill Rts. J.*, 18:1097, 2009.

[PJ78] Lewis F Powell Jr. Smith v. maryland, 1978. https://www.law.cornell.edu/supremecourt/text/442/735.

[PK17] Duong Hieu Phan and Neal Koblitz. Cryptography during the French and American Wars in Vietnam. *Cryptologia*, 41(6):491–511, 2017. https://doi.org/10.1080/01611194.2017.1292825.

[PL18] Pierre Pfister and Ted Lemon. Special-Use Domain 'home.arpa.'. RFC 8375, May 2018. https://rfc-editor.org/rfc/rfc8375.txt.

[PLS13] Nicole Perlroth, Jeff Larson, and Scott Shane. N.S.A. Able to Foil Basic Safeguards of Privacy on Web, 9 2013. https://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html and https://archive.nytimes.com/www.nytimes.com/interactive/2013/09/05/us/documents-reveal-nsa-campaign-against-encryption.html.

[Plu82] David C. Plummer. An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, November 1982. https://rfc-editor.org/rfc/rfc826.txt.

[PMTZ06] V. Pappas, D. Massey, A. Terzis, and L. Zhang. A comparative study of the dns design with dht-based alternatives. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–13, April 2006.

[Poi14] Laura Poitras. CITIZENFOUR, 2014. https://citizenfourfilm.com.

[Poi16] Laura Poitras. Asylum, 2016. Only released at Festival de Cannes 2016.

[Poi17] Laura Poitras. Risk, 2017. https://www.imdb.com/title/tt4964772/.

[Pon11] Jason Pontin. Secrets and Transparency: What is Wikileaks, and what is its future?, 1 2011. https://www.technologyreview.com/2011/01/26/197387/secrets-and-transparency/.

[Pos80a] J. Postel. Internet Protocol Handbook: Table of contents. RFC 774, October 1980.

[Pos80b] J. Postel. User Datagram Protocol. RFC 768 (INTERNET STANDARD), August 1980.

[Pos81a] J. Postel. Internet Control Message Protocol. RFC 777, April 1981. Obsoleted by RFC 792.

[Pos81b] J. Postel. Internet Control Message Protocol. RFC 792 (INTERNET STANDARD), September 1981. Updated by RFCs 950, 4884, 6633, 6918.

[Pos81c] J. Postel. Transmission Control Protocol. RFC 793 (INTERNET STANDARD), September 1981. Updated by RFCs 1122, 3168, 6093, 6528.

[Pos94] J. Postel. Domain Name System Structure and Delegation. RFC 1591 (Informational), March 1994.

[Pou10] Jerry Pournelle. The Iron Law of Bureaucracy, 9 2010. https://www.jerrypournelle.com/reports/jerryp/iron.html.

[Pre15] Bart Preneel. Post-Snowden Threat Models. In Edgar R. Weippl, Florian Kerschbaum, and Adam J. Lee, editors, *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, page 1. ACM, 2015.

[Pre16] Bart Preneel. The future of cryptography. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9665:XVI–XVII, 2016.

[Pre17] Bart Preneel. Post-Snowden Cryptography, 2017. https://handouts.secappdev.org/handouts/2017/Bart%20Preneel/preneel_snowden_secappdev_2017v1_print.pdf.

[Pri14] Privacy and Civil Liberties Oversight Board. Report on the Surveillance Program Operated Pursuant to Section 702 of the Foreign Intelligence Surveillance Act, 2014. July 2nd, 2014; see page 12.

[pro13] Dot-BIT project. The Dot-BIT project, A decentralized, open DNS system based on the bitcoin technology, April 2013. http://dot-bit.org/.

[Pro17] Tor Project. Tor Shared Random Subsystem Specification, 2017. https://gitweb.torproject.org/torspec.git/tree/srv-spec.txt.

[PRS13a] Laura Poitras, Marcel Rosenbach, and Holger Stark. How GCHQ Monitors Germany, Israel and the EU. *Der Spiegel*, 12 2013. https://www.spiegel.de/international/world/snowden-documents-show-gchq-targeted-european-and-german-politicians-a-940135.html.

[PRS13b] Laura Poitras, Marcel Rosenbach, and Holger Stark. How GCHQ Monitors Germany, Israel and the EU, 12 2013. https://www.spiegel.de/international/world/snowden-documents-show-gchq-targeted-european-and-german-politicians-a-940135.html.

[PS14] Stephanie K Pell and Christopher Soghoian. Your secret stingray's no secret anymore: The vanishing government monopoly over cell phone surveillance and its impact on national security and consumer privacy. *Harv. JL & Tech.*, 28:1, 2014. http://jolt.law.harvard.edu/articles/pdf/v28/28HarvJLTech1.pdf.

[Pub12] Public Intelligence. NSA Possible Domestic Interception/Collection Points Map, 4 2012. https://publicintelligence.net/nsa-domestic-collection-points/.

[QA01] Bob Quinn and Dr. Kevin C. Almeroth. IP Multicast Applications: Challenges and Solutions. RFC 3170, September 2001. https://www.rfc-editor.org/info/rfc3170.

[RA16] Tarquin Ramsay and Jörg Altekruse. Free Speech Fear Free, 2016. https://www.imdb.com/title/tt4462056/.

[RAKF12] Marsh Ray, Jacob Appelbaum, Karl Koscher, and Ian Finder. vpwns: Virtual Pwned Networks. In Roger Dingledine and Joss Wright, editors, *2nd USENIX Workshop on Free and Open Communications on the Internet, FOCI '12, Bellevue, WA, USA, August 6, 2012*. USENIX Association, 2012. https://www.usenix.org/conference/foci12/workshop-program/presentation/appelbaum.

[Ram11] Michael Ramsden. Targeted killings and international human rights law: the case of Anwar Al-Awlaki. *Journal of Conflict & Security Law*, 16(2):385–406, 2011.

[RE17] Michael Rogers and Grace Eden. The Snowden disclosures, technical standards and the making of surveillance infrastructures. In *International Journal of Communication*, volume 11, pages 802–823. International Journal of Communication, 2017.

[Rea81] Ronald Reagan. Executive Order 12333: United States Intelligence Activities. *US Federal Register*, 1981. https://www.archives.gov/federal-register/codification/executive-order/12333.html.

[Red14] Redacted (NSA, S32X). QUANTUMTHEORY, 03 2014. https://firstlook.org/theintercept/document/2014/03/12/nsa-gchqs-quantumtheory-hacking-tactics/.

[Ree12] Stuart Rees. Julian Assange and WikiLeaks: A Case Study in the Criminalisation of Dissent. *Argument and Critique*, pages 1–9, 2012.

[Rei21] Janet Reitman. 'I Helped Destroy People', 9 2021. https://www.nytimes.com/2021/09/01/magazine/fbi-terrorism-terry-albury.html and https://www.nytimes.com/2021/09/01/magazine/fbi-terrorism-terry-albury.html.

[Rep18] ETSI Technical Report. Quantum-safe virtual private networks. ETSI TR 103 617, 2018. https://www.etsi.org/deliver/etsi_tr/103600_103699/103617/01.01.01_60/tr_103617v010101p.pdf.

[Res00] E. Rescorla. HTTP Over TLS. RFC 2818 (Informational), May 2000. Updated by RFCs 5785, 7230.

[Res08] Pete Resnick. Internet Message Format. RFC 5322, October 2008. https://rfc-editor.org/rfc/rfc5322.txt.

[Res18] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018.

[REU] REUNION Authors. REUNION python3 implementation. https://codeberg.org/rendezvous/reunion/.

[Ric15] Daniel J Rice. DOCSIS 3.1® technology and hybrid fiber coax for multi-Gbps broadband. In *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–4. IEEE, 2015.

[Ric18] Norma M Riccucci. 2. privacy rights and us surveillance policy drifts. In *Policy Drift*, pages 21–82. New York University Press, 2018.

[RJ10] George Ritzer and Nathan Jurgenson. Production, consumption, prosumption: The nature of capitalism in the age of the digital 'prosumer'. *Journal of consumer culture*, 10(1):13–36, 2010.

[rK97] D. Eastlake 3rd and C. Kaufman. Domain Name System Security Extensions. RFC 2065 (Proposed Standard), 1 1997. Obsoleted by RFC 2535 http://www.ietf.org/rfc/rfc2065.txt.

[RLM⁺05] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. DNS Security Introduction and Requirements. RFC 4033, March 2005. https://rfc-editor.org/rfc/rfc4033.txt.

[RM12] E. Rescorla and N. Modadugu. Datagram Transport Layer Security Version 1.2. RFC 6347 (Proposed Standard), January 2012. Updated by RFCs 7507, 7905.

[RN05] Vivek Ramachandran and Sukumar Nandi. Detecting ARP spoofing: An active technique. In Sushil Jajodia and Chandan Mazumdar, editors, *Information Systems Security, First International Conference, ICISS 2005, Kolkata, India, December 19-21, 2005, Proceedings*, volume 3803 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2005. https://doi.org/10.1007/11593980_18.

[RNSL17] Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin E. Lauter. Quantum resource estimates for computing elliptic curve discrete logarithms. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 241–270. Springer, 2017.

[Rob14] John Robinson. The Snowden Disconnect: When the Ends Justify the Means, 2014. https://ssrn.com/abstract=2427412.

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 98–107, 2002.

[Rog15] Phillip Rogaway. The Moral Character of Cryptographic Work. Cryptology ePrint Archive, Report 2015/1162, 2015. http://eprint.iacr.org/2015/1162.

[Roo42] Franklin D Roosevelt. Executive Order 9066. *US National Archives & Records Administration*, 1942.

[Ros93] R. Rosenbaum. Using the Domain Name System To Store Arbitrary String Attributes. RFC 1464 (Experimental), May 1993.

[Ros10] J. Rosenberg. Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols. RFC 5245 (Proposed Standard), April 2010. Updated by RFC 6336.

[Rot15] Sebastian Rotella. The Hidden Intelligence Breakdowns Behind the Mumbai Attacks. *ProPublica*, 4 2015. https://www.propublica.org/article/the-hidden-intelligence-breakdowns-behind-the-mumbai-attacks.

[RP14] James Risen and Laura Poitras. N.S.A. Collecting Millions of Faces From Web Images, 5 2014. https://www.nytimes.com/2014/06/01/us/nsa-collecting-millions-of-faces-from-web-images.html and https://www.nytimes.com/interactive/2014/06/01/us/nsa-document.html.

[RS] Leif Ryge and David Stainton. DSpool: Durable ephemeral soft-queues without single points of failure. https://github.com/katzenpost/docs/blob/master/drafts/dspool.rst.

[RS04] Venugopalan Ramasubramanian and Emin Gün Sirer. The design and implementation of a next generation name service for the internet. *SIGCOMM Comput. Commun. Rev.*, 34(4):331–342, August 2004.

[RS14] Marcel Rosenback and Holger Stark. *Der NSA-Komplex: Edward Snowden und der Weg in die totale Überwachung*. DVA, 2014.

[RS19] B Indira Reddy and V Srikanth. Review on wireless security protocols (WEP, WPA, WPA2 & WPA3). *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2019.

[RSS15] Marcel Rosenbach, Hilmar Schmundt, and Christian Stöcker. Experts Unmask ’Regin’ Trojan as NSA Tool. *Der Spiegel*, 1 2015. https://www.spiegel.de/international/world/regin-malware-unmasked-as-nsa-tool-after-spiegel-publishes-source-code-a-1015255.html.

[Ruy20] Björn Ruytenberg. Breaking Thunderbolt Protocol Security: Vulnerability Report, 2020. https://thunderspy.io/assets/docs/breaking-thunderbolt-security-bjorn-ruytenberg-20200417.pdf.

[Ruy22] Björn Ruytenberg. When Lightning Strikes Thrice: Breaking Thunderbolt Security. Master’s thesis, Eindhoven University of Technology, February 2022. Master’s thesis to appear and https://bjornweb.nl/masters-thesis.

[Rya13] Kevin Ryan. How the NSA’s Spying on Americans Relates to 9/11. *Foreign Policy Journal*, pages 1–5, 2013. https://www.foreignpolicyjournal.com/wp-content/uploads/2013/10/131017-Ryan-NSA-911.pdf.

[Ryg16] Leif Ryge. "add wifi geolocation from space to the list of things that once sounded crazy but actually happens /ty", 2016. https://nitter.net/wiretapped/status/773136872323317760.

[S3210] S32X, unknown. 2010 SIGINT Development Conference: QUANTUMTHEORY, 2010. https://cryptome.org/2014/03/nsa-gchq-quantumtheory.pdf.

[SA15a] Markku-Juhani O. Saarinen and Jean-Philippe Aumasson. The BLAKE2 cryptographic hash and message authentication code (MAC). *RFC*, 7693:1–30, 2015.

[SA15b] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Address Format. RFC 7622, September 2015. https://rfc-editor.org/rfc/rfc7622.txt.

[SAH11] P. Saint-Andre and J. Hodges. Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS). RFC 6125 (Proposed Standard), March 2011.

[Sai02] Natsu Taylor Saito. Whose liberty – whose security – The USA PATRIOT Act in the context of COINTELPRO and the unlawful repression of political dissent. *Or. L. Rev.*, 81:1051, 2002.

[Sav20] Charlie Savage, 12 2020. https://www.nytimes.com/2020/12/03/us/politics/section-215-patriot-act.html.

[SB11] Scott Shane and John F Burns. US Subpoenas Twitter Over WikiLeaks Supporters. *The New York Times*, 2011.

[SB13] Andrei Soldatov and Irina Borogan. Russia's surveillance state. *World Policy Journal*, 30(3):23–30, 2013.

[SBF15] Jantine Schroeder, Radu Botez, and Marine Formentini. Radioactive Waste Management and Constructing Memory for Future Generations. Proceedings of the International Conference and Debate, 15-17 September 2014, Verdun, France. Technical report, Organisation for Economic Co-Operation and Development, 2015. https://www.oecd-nea.org/upload/docs/application/pdf/2020-12/7259-constructing-memory-2015.pdf.

[SC81] Diane St. Clair. Bibliography on Repression. *The Black Scholar*, 12(1):85–90, 1981.

[SC13] John Shiffman and Kristina Cooke. Exclusive: U.S. directs agents to cover up program used to investigate Americans. https://www.reuters.com/article/us-dea-sod-idUSBRE97409R20130805, 2013.

[Sca13a] Jeremy Scahill. *Dirty wars: The world is a battlefield*. Hachette UK, 2013.

[Sca13b] Jeremy Scahill. *Dirty wars: The world is a battlefield*. Hachette UK, 2013.

[Sch09] Michael P Scharf. International law and the torture memos. *Case W. Res. J. Int'l L.*, 42:321, 2009.

[Sco14] Katy Scoggin. Chokepoint, 2014. `https://www.imdb.com/title/tt4538388/`.

[Seg14] Kerry Segrave. *Wiretapping and Electronic Surveillance in America, 1862-1920*. McFarland, 2014.

[Sen96] United States Senate. CIA's use of journalists and clergy in intelligence operations, 1996. `https://www.intelligence.senate.gov/sites/default/files/hearings/ciasuseofjournal00unit.pdf`.

[Ser05] Victor Serge. *What Every Radical Should Know about State Repression: A Guide for Activists*. Ocean Press, 2005. First printing in 1926 as "Les coulisses d'une sûreté générale. Ce que tout révolutionnaire doit savoir sur la répression, Librairie du Travail, Paris".

[SG14] Jeremy Scahill and Glenn Greenwald. The NSA's secret role in the US assassination program. *The intercept*, 10:2014, 2014. `https://theintercept.com/2014/02/10/the-nsas-secret-role/`.

[SG21] Nathan Stephens Griffin. "Everyone was questioning everything": understanding the derailing impact of undercover policing on the lives of UK environmentalists. *Social Movement Studies*, 20(4):459–477, 2021.

[Sha49] Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.

[Sha10] Gene Sharp. From Dictatorship to Democracy, 2010.

[SHD91] C. Shue, W. Haggerty, and K. Dobbins. OSI connectionless transport services on top of UDP: Version 1. RFC 1240 (Historic), June 1991.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994.

[Sho99] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review*, 41(2):303–332, 1999.

[Sho13] Tim Shorrock. Obama's crackdown on whistleblowers. *The Nation*, 2013.

[Shu14] Haya Shulman. Pretty bad privacy: Pitfalls of DNS encryption. In *13th Workshop on Privacy in the Electronic Society*. ACM, 2014.

[Sil07] WMD Silences. Situated Ignorance and State Terrorism. *Violent Geographies: Fear, Terror, and Political Violence*, page 363, 2007.

[Sin07] Ryan Singel. Point, click ... eavesdrop: How the FBI wiretap net operates. `https://www.wired.com/2007/08/wiretap/`, 2007.

[SJK+17] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J Fischer, and Bryan Ford. Scalable bias-resistant distributed randomness. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 444–460. IEEE, 2017.

[Sla19] Slack Inc. Nebula. https://github.com/slackhq/nebula, 11 2019.

[SM13] Scott Shane and Colin Moynihan. Drug Agents Use Vast Phone Trove, Eclipsing N.S.A.'s, 9 2013. https://www.nytimes.com/2013/09/02/us/drug-agents-use-vast-phone-trove-eclipsing-nsas.html and https://archive.nytimes.com/www.nytimes.com/interactive/2013/09/02/us/hemisphere-project.html.

[Sny17] Timothy Snyder. *On tyranny: Twenty lessons from the twentieth century*. Random House, 2017.

[Söd11] Johan Söderberg. *Free software to open hardware: Critical theory on the frontiers of hacking*. University of Gothenburg, 02 2011. ISSN: 1650-4437 and ISBN: 978-91-975442-7-6.

[Spi86] Paul R. Spickard. Injustice Compounded: Amerasians and Non-Japanese Americans in World War II Concentration Camps. *Journal of American Ethnic History*, 5(2):5–22, 1986.

[Spi15] Spiegel Staff. An Attack on Press Freedom: SPIEGEL Targeted by US Intelligence, 7 2015. https://www.spiegel.de/international/germany/the-nsa-and-american-spies-targeted-spiegel-a-1042023.html.

[Spu00] Charles E Spurgeon. *Ethernet: the definitive guide*. O'Reilly Media, Inc., 2000.

[Sta87] William Stallings. *Handbook of computer-communications standards; Vol. 1: the open systems interconnection (OSI) model and OSI-related standards*. Macmillan Publishing Co., Inc., 1987.

[Sta02] Richard Stallman. *Free software, free society: Selected essays of Richard M. Stallman*. https://www.lulu.com, 2002. https://www.gnu.org/doc/fsfs3-hardcover.pdf.

[Sta18] David Stainton. PANDA protocol implementation for Katzenpost Mix Network, 2018. https://github.com/katzenpost/panda/.

[Sta19a] David Stainton. Katzenpost mix network encrypted messaging client library, 2019. https://github.com/katzenpost/reunion/.

[Sta19b] David Stainton. REUNION protocol implementation for Katzenpost Mix Network, 2019. https://github.com/katzenpost/reunion/.

[Ste94] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

[Sti05] Marc Stiegler. An introduction to petname systems. *Advances in Financial Cryptography*, 2005. https://www.financialcryptography.com/mt/archives/000499.html.

[Sti08] Sarah Lai Stirland. Cisco Leak: 'Great Firewall' of China Was a Chance to Sell More Routers, 5 2008. https://www.wired.com/images_blogs/threatlevel/files/cisco_presentation.pdf.

[Stö13] Christian Stöcker. GCHQ Surveillance: The Power of Britain's Data Vacuum, 7 2013. https://www.spiegel.de/international/world/snowden-reveals-how-gchq-in-britain-soaks-up-mass-internet-data-a-909852.html.

[Sto16] Oliver Stone. Snowden, 2016. https://www.imdb.com/title/tt3774114/.

[SW09] Frank Stajano and Paul Wilson. Understanding scam victims: seven principles for systems security. Technical Report UCAM-CL-TR-754, University of Cambridge, Computer Laboratory, August 2009.

[SW11] Frank Stajano and Paul Wilson. Understanding Scam Victims: Seven Principles for Systems Security. *Commun. ACM*, 54(3):70–75, mar 2011.

[Swa11] Aaron Swartz. Squaring the triangle: Secure, decentralized, human-readable names. http://www.aaronsw.com/weblog/squarezooko, 01 2011. Accessed: 2017-10-30.

[SWZ16a] John M Schanck, William Whyte, and Zhenfei Zhang. Circuit-extension handshakes for Tor achieving forward secrecy in a quantum world. *Proceedings on Privacy Enhancing Technologies*, 4:219–236, 2016.

[SWZ16b] John M. Schanck, William Whyte, and Zhenfei Zhang. Circuit-extension handshakes for tor achieving forward secrecy in a quantum world. *Proc. Priv. Enhancing Technol.*, 2016(4):219–236, 2016. https://doi.org/10.1515/popets-2016-0037.

[Sys02] Cisco Systems. Overview of the Public Security Sector, 2002. https://www.wired.com/images_blogs/threatlevel/files/cisco_presentation.pdf.

[Tai20] Tailscale. Tailscale. https://tailscale.com/, 2020.

[Tay88] Francis X Taylor. US Counterintelligence: From the Year of Intelligence to the Year of the Spy and Poised for the Future. Technical report, AIR WAR COLL MAXWELL AFB AL, 1988. https://apps.dtic.mil/sti/pdfs/ADA202072.pdf.

[The93] Athan Theoharis. The FBI, the Roosevelt Administration, and the 'Subversive' Press. *Journalism History*, 19(1):3–10, 1993.

[The17] The Intercept. The FBI's Secret Rules, 2017. `https://theintercept.com/series/the-fbis-secret-rules/`.

[The19] The Guardian. The Cambridge Analytica Files, 2019. `https://www.theguardian.com/news/series/cambridge-analytica-files`.

[Thi18] Johannes Thimm. *From exception to normalcy: the United States and the war on terrorism*, volume 7/2018 of *SWP Research Paper*. Stiftung Wissenschaft und Politik -SWP- Deutsches Institut für Internationale Politik und Sicherheit, Berlin, 2018. `https://www.swp-berlin.org/publications/products/research_papers/2018RP07_tmm.pdf`.

[Tic06] Russell D. Tice. Russell D. Tice's 2006 letters to the United States Senate and the United States House of Representatives, 2006. `https://irp.fas.org/news/2006/04/tice042506.pdf`.

[Tim02] Jacobo Timerman. *Prisoner without a Name, Cell without a Number*. Univ of Wisconsin Press, 2002.

[TM67] Athan G Theoharis and Elizabeth Meyer. The national security justification for electronic eavesdropping: An elusive exception. *Wayne L. Rev.*, 14:749, 1967.

[TNE08] Minh-Triet Tran, Thanh-Trung Nguyen, and Isao Echizen. Pool-Based APROB Channel to Provide Resistance against Global Active Adversary under Probabilistic Real-Time Condition. In *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, volume 2, pages 257–263, 2008.

[tOCS21] N. ten Oever, C. Cath, and S. Sahib. Establishing the Protocol Police. RFC 8962, 4 2021.

[Ton21] Tonari Inc. innernet. `https://github.com/tonarino/innernet`, 2021.

[TP11] S. Turner and T. Polk. Prohibiting Secure Sockets Layer (SSL) Version 2.0. RFC 6176 (Proposed Standard), March 2011.

[TT16] Craig Timberg and Hayley Tsukayama. Yahoo says 1 billion user accounts were hacked, 12 2016. `https://www.washingtonpost.com/business/economy/yahoo-says-1-billion-user-accounts-hacked/2016/12/14/a301a7d8-b986-4281-9b13-1561231417c0_story.html`.

[Tur14] Simone Turchetti. "In god we trust, all others we monitor": Seismology, surveillance, and the test ban negotiations. In *The Surveillance Imperative*, pages 85–102. Springer, 2014.

[TW17] Magdalena Taube and Krystian Woznicki, editors. *A Field Guide to the Snowden Files*. DIAMONDPAPER, 2017. https://diamondpaper.net/title_26.

[ua] Stasi unterlagen archive. Die Grundsatzdokumente des Ministeriums für Staatssicherheit. https://www.stasi-unterlagen-archiv.de/informationen-zur-stasi/quellensammlungen/die-grundsatzdokumente-des-ministeriums-fuer-staatssicherheit/.

[Uni13a] American Civil Liberties Union. Unleashed and unaccountable: The fbi's unchecked abuse of authority, 2013. https://www.aclu.org/sites/default/files/assets/unleashed-and-unaccountable-fbi-report.pdf.

[Uni13b] United States of America's National Security Agency. Nsa documents about "Big Access to Big Data" through TEMPORA dataset., 2013. https://www.spiegel.de/media/53dc023b-0001-0014-0000-000000034090/media-34090.pdf and https://www.spiegel.de/media/1ad582ff-0001-0014-0000-000000034103/media-34103.pdf.

[Unk13] Unknown. XKeyscore slide presentation, 2013. https://www.theguardian.com/world/interactive/2013/jul/31/nsa-xkeyscore-program-full-presentation.

[Unk14a] Unknown. NSA/CSS Threat Operations Center (NTOC): Bad guys are everywhere, good guys are somewhere! http://www.spiegel.de/media/media-34757.pdf, 09 2014.

[Unk14b] Unknown. There is More Than One Way to QUANTUM, 3 2014. https://theintercept.com/document/2014/03/12/one-way-quantum/.

[Unk14c] Unknown. There is more than one way to quantum, 03 2014. https://www.documentcloud.org/documents/1076891-there-is-more-than-one-way-to-quantum.html#document/p1.

[Unk15a] Unknown. IC OFF THE RECORD: XKeyscore document archive, 2013 – 2015. https://nsa.gov1.info/dni/xkeyscore.html.

[Unk15b] Unknown. Exploiting Foreign Lawful Intercept Roundtable, 9 2015. https://s3.documentcloud.org/documents/2434264/2012-lawful-intercept-redacted2.pdf.

[U.S22] U.S. Senator Ron Wyden and U.S. Sen. Martin Heinrich. Wyden and Heinrich: Newly Declassified Documents Reveal Previously Secret CIA Bulk Collection, Problems With CIA Handling of Americans' Information, 02 2022. https://www.wyden.senate.gov/news/press-releases/wyden-and-heinrich-newly-declassified-documents-reveal-previously-secret-cia-bulk-collection-problems-with-cia-handling-of-americans-information.

[Vai19] Loup Vaillant. Monocypher, 2019. https://monocypher.org/.

[Vai20a] Loup Vaillant. Cofactor Explained: Clearing Elliptic Curves' dirty little secret, 4 2020. https://loup-vaillant.fr/tutorials/cofactor.

[Vai20b] Loup Vaillant. Surrounded by Elligators: Implementing Crypto With Nothing to Compare to, 4 2020. https://loup-vaillant.fr/articles/implementing-elligator.

[VCB+04] John Vollbrecht, James D. Carlson, Larry Blunk, Dr. Bernard D. Aboba, and Henrik Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748, June 2004. https://rfc-editor.org/rfc/rfc3748.txt.

[VHM16] Stephanie Vogelgesang, Stefan Hessel, and Frederik Möllers. Hardware-Keylogger: Die Tastatur in der Hand des Feindes. *Datenschutz und Datensicherheit-DuD*, 40(11):729–734, 2016.

[Vie11] Stefan Viehböck. Brute forcing Wi-Fi Protected Setup, 2011. https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.

[VR20] Mathy Vanhoef and Eyal Ronen. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 517–533. IEEE, 2020.

[Vul21] Vula Authors. Vula: automatic local area network encryption. https://vula.link, 2021.

[vZCMS17] Khadija von Zinnenburg Carroll, Michał Murawski, and Jesse Weaver Shipley. The Art of Dissident Domesticity: Julian Assange, King Prempeh, and Ethnographic Conceptualism in the Prison House. *Social Text*, 35(4):113–152, 12 2017.

[Wah13] Johannes Wahlström. Mediastan: A WikiLeaks Road Movie, 10 2013. https://wikileaks.org/Watch-MEDIASTAN.html and https://www.youtube.com/watch?v=9n0Yu7bYF9E and https://www.imdb.com/title/tt3169780/.

[Wak19] Satohiro Wakabayashi. *Investigation of Radio Frequency Retroreflector Attacks*. PhD thesis, Waseda University, 2019.

[WAP08] Dan Wendlandt, David G. Andersen, and Adrian Perrig. *Perspectives:* improving ssh-style host authentication with multi-path probing. In Rebecca Isaacs and Yuanyuan Zhou, editors, *2008 USENIX Annual Technical Conference, Boston, MA, USA, June 22-27, 2008. Proceedings*, pages 321–334. USENIX Association, 2008. http://www.usenix.org/events/usenix08/tech/full_papers/wendlandt/wendlandt.pdf.

[War15a] Brian Warner. Magic wormhole, 2015. https://github.com/warner/magic-wormhole.

[War15b] Ian Warren. Surveillance, criminal law and sovereignty. *Surveillance & Society*, 13(2):300–305, 2015. https://ojs.library.queensu.ca/index.php/surveillance-and-society/article/download/law_sovereign/law_sov.

[Wea14] Nicholas Weaver. A close look at the NSA's most powerful Internet attack tool. *Wired*, 03 2014.

[Wea16] Russell L Weaver. Free Speech, Transparency, and Democratic Government: an American Perspective. *Revue Internationale des Gouvernements Ouverts*, 2:165–176, 2016. https://core.ac.uk/download/pdf/235040239.pdf.

[Wei14] Stephen Weis. Protecting data in-use from firmware and physical attacks. *Black Hat*, 2014.

[Whi19] Zack Whittaker. Documents reveal how Russia taps phone companies for surveillance, 9 2019. https://techcrunch.com/2019/09/18/russia-sorm-nokia-surveillance.

[Wie04] Michael J. Wiener. The full cost of cryptanalytic attacks. *J. Cryptology*, 17(2):105–124, 2004.

[Wij14] W. Wijngaards. Confidential DNS, 09 2014. http://tools.ietf.org/html/draft-wijngaards-dnsop-confidentialdns-02.

[Wik08] WikiLeaks. Kenya: The Cry of Blood - Report on Extra-Judicial Killings and Disappearances, 11 2008. https://wikileaks.org/wiki/Kenya:_The_Cry_of_Blood_-_Report_on_Extra-Judicial_Killings_and_Disappearances,_Sep_2008.

[Wik10a] WikiLeaks. Public Library of US Diplomacy, 1966 – 2010. https://wikileaks.org/plusd/.

[Wik10b] WikiLeaks. Afghan War Diary, 2004 – 2010, 07 2010. https://wardiary.wikileaks.org/.

[Wik10c] WikiLeaks. Cablegate: 250,000 US Embassy Diplomatic Cables, 11 2010. https://www.wikileaks.org/Cablegate-250-000-US-Embassy.html.

[Wik10d] WikiLeaks. Collateral murder, 4 2010. https://collateralmurder.wikileaks.org/.

[Wik10e] WikiLeaks. Kabul war diary, 7 2010. https://wikileaks.org/afg/.

[Wik11a] WikiLeaks. Spy files - release 1, 12 2011. https://wikileaks.org/spyfiles/.

[Wik11b] WikiLeaks. Spy files - release 2, 12 2011. https://wikileaks.org/spyfiles/.

[Wik13] WikiLeaks. Spy files - release 3, 9 2013. https://wikileaks.org/spyfiles/.

[Wik14] WikiLeaks. Spy files - release 4, 9 2014. https://wikileaks.org/spyfiles/.

[Wik17a] WikiLeaks. 2012 french presidential election hacking by the cia, 2 2017. https://wikileaks.org/cia-france-elections-2012/.

[Wik17b] WikiLeaks. Spy files: Russia. https://wikileaks.org/spyfiles/russia/#SORM, 09 2017.

[Wik17c] WikiLeaks. Vault 7: , 6 2017. https://wikileaks.org/vault7/#Pandemic.

[Wik17d] WikiLeaks. Vault 7: AfterMidnight, 5 2017. https://wikileaks.org/vault7/#AfterMidnight.

[Wik17e] WikiLeaks. Vault 7: Angelfire, 8 2017. https://wikileaks.org/vault7/#Angelfire.

[Wik17f] WikiLeaks. Vault 7: Archimedes, 5 2017. https://wikileaks.org/vault7/#Archimedes.

[Wik17g] WikiLeaks. Vault 7: Athena, 5 2017. https://wikileaks.org/vault7/#Athena.

[Wik17h] WikiLeaks. Vault 7: BothanSpy, 7 2017. https://wikileaks.org/vault7/#BothanSpy.

[Wik17i] WikiLeaks. Vault 7: Brutal Kangaroo, 6 2017. https://wikileaks.org/vault7/#Brutal%20Kangaroo.

[Wik17j] WikiLeaks. Vault 7: Cherry Blossom, 6 2017. https://wikileaks.org/vault7/#Cherry%20Blossom.

[Wik17k] WikiLeaks. Vault 7: CIA Hacking Tools Revealed, 3 2017. https://wikileaks.org/ciav7p1/.

[Wik17l] WikiLeaks. Vault 7: CouchPotato, 8 2017. https://wikileaks.org/vault7/#CouchPotato.

[Wik17m] WikiLeaks. Vault 7: Dark Matter, 3 2017. https://wikileaks.org/vault7/#Dark%20Matter.

[Wik17n] WikiLeaks. Vault 7: Dumbo, 8 2017. https://wikileaks.org/vault7/#Dumbo.

[Wik17o] WikiLeaks. Vault 7: Elsa, 6 2017. https://wikileaks.org/vault7/#Elsa.

[Wik17p] WikiLeaks. Vault 7: ExpressLane, 8 2017. https://wikileaks.org/vault7/#ExpressLane.

[Wik17q] WikiLeaks. Vault 7: Grasshopper, 4 2017. https://wikileaks.org/vault7/#Grasshopper.

[Wik17r] WikiLeaks. Vault 7: Highrise, 7 2017. https://wikileaks.org/vault7/#Highrise.

[Wik17s] WikiLeaks. Vault 7: Hive, 4 2017. https://wikileaks.org/vault7/#Hive.

[Wik17t] WikiLeaks. Vault 7: Imperial, 7 2017. https://wikileaks.org/vault7/#Imperial.

[Wik17u] WikiLeaks. Vault 7: Marble Framework, 3 2017. https://wikileaks.org/vault7/#Marble%20Framework.

[Wik17v] WikiLeaks. Vault 7: OutlawCountry, 6 2017. https://wikileaks.org/vault7/#OutlawCountry.

[Wik17w] WikiLeaks. Vault 7: Protego, 9 2017. https://wikileaks.org/vault7/#Protego.

[Wik17x] WikiLeaks. Vault 7: Scribbles, 4 2017. https://wikileaks.org/vault7/#Scribbles.

[Wik17y] WikiLeaks. Vault 7: UCL / Raytheon, 7 2017. https://wikileaks.org/vault7/#UCL%20/%20Raytheon.

[Wik17z] WikiLeaks. Vault 7: Weeping Angel, 4 2017. https://wikileaks.org/vault7/#Weeping%20Angel.

[Wik18] Wikipedia contributors. BATON — Wikipedia, The Free Encyclopedia, 2018. https://en.wikipedia.org/w/index.php?title=BATON&oldid=866536467 [Online; accessed 27-December-2021].

[Wik20a] Wikipedia. Onyx (Abhörsystem) — Wikipedia, die freie Enzyklopädie, 2020. [Online; Stand 16. Dezember 2021] https://de.wikipedia.org/w/index.php?title=Onyx_(Abh%C3%B6rsystem)&oldid=205909088.

[Wik20b] Wikipedia contributors. Computer network operations — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Computer_network_operations&oldid=964965630, 2020. [Online; accessed 31-December-2021].

[Wik20c] Wikipedia contributors. TRAFFICTHIEF — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=TRAFFICTHIEF&oldid=986162796, 2020. [Online; accessed 29-December-2021].

[Wik20d] Wikipedia contributors. TURBINE (US government project) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=TURBINE_(US_government_project)&oldid=950962842, 2020. [Online; accessed 29-December-2021].

[Wik21a] Wikipedia. Fichenskandal — Wikipedia, die freie Enzyklopädie, 2021. "[Online; Stand 16. Dezember 2021] https://de.wikipedia.org/w/index.php?title=Fichenskandal&oldid=217497919".

[Wik21b] Wikipedia contributors. Bernstein v. United States — Wikipedia, The Free Encyclopedia, 2021. [Online; accessed 31-August-2021].

[Wik21c] Wikipedia contributors. Black bag operation — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Black_bag_operation&oldid=1008683558, 2021. [Online; accessed 31-December-2021].

[Wik21d] Wikipedia contributors. Casualties of the Iraq War — Wikipedia, The Free Encyclopedia, 2021. [Online; accessed 17-September-2021].

[Wik21e] Wikipedia contributors. Civilian casualties in the war in Afghanistan (2001–2021) — Wikipedia, The Free Encyclopedia, 2021. [Online; accessed 17-September-2021].

[Wik21f] Wikipedia contributors. Dehomag — Wikipedia, The Free Encyclopedia, 2021. [Online; accessed 31-August-2021].

[Wik21g] Wikipedia contributors. Disposition Matrix — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Disposition_Matrix&oldid=1049582806, 2021. [Online; accessed 28-December-2021].

[Wik21h] Wikipedia contributors. Foreign Intelligence Surveillance Act — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Foreign_Intelligence_Surveillance_Act&oldid=1056215791, 2021. [Online; accessed 28-December-2021].

[Wik21i] Wikipedia contributors. Fusion center — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Fusion_center&oldid=1049173657, 2021. [Online; accessed 10-January-2022].

[Wik21j] Wikipedia contributors. Interdiction — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Interdiction&oldid=1004880009, 2021. [Online; accessed 31-December-2021].

[Wik21k] Wikipedia contributors. MAINWAY — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=MAINWAY&oldid=1062704867, 2021. [Online; accessed 31-December-2021].

[Wik21l] Wikipedia contributors. Nulla poena sine lege — Wikipedia, The Free Encyclopedia, 2021. https://en.wikipedia.org/w/index.php?title=Nulla_poena_sine_lege&oldid=1052378885 [Online; accessed 27-December-2021].

[Wik21m] Wikipedia contributors. Opabinia — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Opabinia&oldid=1060578005, 2021. [Online; accessed 28-December-2021].

[Wik21n] Wikipedia contributors. Pocket Litter — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Pocket_litter&oldid=984150384, 2021. [Online; accessed 28-December-2021].

[Wik21o] Wikipedia contributors. President's Surveillance Program — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=President%27s_Surveillance_Program&oldid=1050072262, 2021. [Online; accessed 28-December-2021].

[Wik21p] Wikipedia contributors. SIGINT Activity Designator — Wikipedia, The Free Encyclopedia, 2021. [Online; accessed 25-December-2021].

[Wik21q] Wikipedia contributors. Stellar Wind — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Stellar_Wind&oldid=1060521549, 2021. [Online; accessed 28-December-2021].

[Wik21r] Wikipedia contributors. The Shadow Brokers — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=The_Shadow_Brokers&oldid=1045163706, 2021. [Online; accessed 31-December-2021].

[Wik21s] Wikipedia contributors. ThinThread — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=ThinThread&oldid=1051207137, 2021. [Online; accessed 31-December-2021].

[Wik21t] Wikipedia contributors. Trailblazer Project — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Trailblazer_Project&oldid=1021403819, 2021. [Online; accessed 28-December-2021].

[Wik21u] Wikipedia contributors. Turbulence (NSA) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Turbulence_(NSA)&oldid=1026069496, 2021. [Online; accessed 29-December-2021].

[Wik21v] Wikipedia contributors. Zersetzung — Wikipedia, The Free Encyclopedia, 2021. https://en.wikipedia.org/w/index.php?title=Zersetzung&oldid=1056769799.

[Wik22a] Wikipedia contributors. Fourth Amendment to the United States Constitution — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Fourth_Amendment_to_the_United_States_Constitution&oldid=1071305598, 2022. [Online; accessed 12-February-2022].

[Wik22b] Wikipedia contributors. President's Surveillance Program — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=President%27s_Surveillance_Program&oldid=1070717732, 2022. [Online; accessed 12-February-2022].

[Win21] Jana Winter. Operation Whistle Pig: Inside the secret CBP unit with no rules that investigates Americans, 12 2021. https://news.yahoo.com/operation-whistle-pig-inside-the-secret-cbp-unit-with-no-rules-that-investigates-americans-100000147.html.

[Wir21] WireShark authors. WireShark, 1998-2021. https://www.wireshark.org/.

[WM15] Ben Wagner and Patricia Mindus. Multistakeholder Governance and Nodal Authority–Understanding Internet Exchange Points. NoC Internet Governance Research Project: Case Studies, Case Study 7, 2015. https://publixphere.net/i/noc/page/IG_Case_Study_Multistakeholder_Governance_and_Nodal_Authority_Understanding_Internet_Exchange_Points.html.

[Wol04] Laurence J. Wolf. Preventing Failure: The Value of Performing a Single Point of Failure Analysis for Critical Applications and Systems. *Information Systems Security*, 13(1):51–54, 2004.

[Woo18] Connor Woodman. Spycops in context: A brief history of political policing in Britain. *Centre for Crime and Justice Studies*, 2018.

[Wou13] Paul Wouters. History and implementation status of Opportunistic Encryption for IPsec. https://nohats.ca/wordpress/blog/2013/09/12/history-and-implementation-status-of-opportunistic-encryption-for-ipsec/, 2013.

[WSG14] Matthias Wachs, Martin Schanzenbach, and Christian Grothoff. A censorship-resistant, privacy-enhancing and fully decentralized name system. In *13th International Conference on Cryptology and Network Security (CANS 2014)*, pages 127–142, 2014.

[Wu18] Peter Wu. Bug 15011 - Support for WireGuard VPN protocol, 2018. `https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=15011`.

[WWW15] Klaas Wierenga, Stefan Winter, and Tomasz Wolniewicz. The eduroam Architecture for Network Roaming. RFC 7593, September 2015. `https://rfc-editor.org/rfc/rfc7593.txt`.

[Yar31] Herbert O Yardley. *The American Black Chamber*, volume 40. Bobbs-Merrill, 1931.

[Yon] James Yonan. OpenVPN. `https://openvpn.net/`.

[Zal01] Michal Zalewski. Strange Attractors: TCP/IP Sequence Number Analysis, 2001. `https://lcamtuf.coredump.cx/oldtcp/tcpseq.html`.

[ZEE16] A. Zimmermann, W. Eddy, and L. Eggert. Moving Outdated TCP Extensions and TCP-Related Documents to Historic or Informational Status. RFC 7805 (Informational), April 2016.

[Ziz08] Slavov Zizek. Nature and its Discontents. *SubStance*, 37(3):37–72, 2008.

[Zub19] Shoshana Zuboff. *The age of surveillance capitalism: The fight for a human future at the new frontier of power*. Profile books, 2019.

General index

802.11, *see* Wireless LAN
802.1x authentication, *see* Authentication, Authorization, and Accounting
802.3, *see* Ethernet

AAA, *see* Authentication, Authorization, and Accounting
Accused whistleblower
  Joshua Schulte, 67
Address Resolution Protocol, 18
  Spoofing, 193, 215–217
Advanced Encryption Standard, *see* Block cipher
Adversary
  End Users, 193, 217
  Network Operator, 194, 218
  Unilateral Surveillance Adversary, 193, 217
AEAD, *see* Authenticated-Encryption with Associated-Data
AES, *see* Block cipher
Afghanistan
  Afghan War logs, 61
  American war in Afghanistan, 9, 61, 79
African National Congress, 83
AFSA, *see* United States of America
al-Awlaki, Abdulrahman Anwar, *see* Assassination victim
al-Awlaki, Anwar Nasser, *see* Assassination victim
al-Awlaki, Nawar, *see* Assassination victim
Albury, Terry J., *see* Whistleblower
ANC, *see* African National Congress
ANGRYNEIGHBOR, 108–111, 129, 130
Anonymity, 245
  Biometric surveillance, 3
  Stylometry, 3
  Unlinkability, 182
ANT, *see* ANT catalog
ANT catalog, 92–144
AOL, *see* Corporate Surveillance Partner
Apple
  in the PRISM program, 74, 77
  iPhone, 64, 123
  MacOS, 64, 66
    SeaPea (kernel rootkit), 64
    WireGuard, 172
Argon2id, *see* Hash function
ARKSTREAM, 94, 133
ARM, *see* CPU
ARP, *see* Address Resolution Protocol
ASA, *see* Cisco Systems
Ashur, Tomer, 81
Assange, Julian, *see* Journalist
Assassination victim
  Abdulrahman Anwar al-Awlaki, 58
  Anwar Nasser al-Awlaki, 58
  Nawar al-Awlaki, 58
  Patrice Lumumba, 70
Audio surveillance, 3, *see* Telephone surveillance
Authenticated-Encryption with Associated-Data, 28, 32, 173, 174, 235, 244
  ChaCha20Poly1305, 173, 174, 244, 246, 247

Authentication, Authorization, and Accounting, 189

Backdoor, 127
  BSAFE toolkit, 81
  Bugdoor, 81
  Cavium CPU backdoor, 72
  Clipper Chip, 56
  DES, 81
  DUAL_EC_DRBG, 81
  Eastcom
    760c+, 127
  MAC address covert channel, 214
  NOBUS, 81
  PX-1000cr, 83
  Samsung
    C140, 127
    E600, 127
    X450, 127
  Thuraya 2520, 126
  Unreleased programs, 84
Bahamas, 79
Bamford, James, *see* Journalist
BANANAGLEE, 99
Bandwidth
  450Mhz, 110
  45Mhz, 107
Barcelona Telephone exchange, 80
Barlow, John Perry, 56
Base64, 204
Belgium
  Belgacom, 41
Belkin, 65
Bernstein v. United States, 56
Bernstein, Daniel J., 56
Binney, Bill, *see* Whistleblower
BIOS
  DEITYBOUNCE, 94
  FEEDTROUGH, 99
  GOURMETTROUGH, 100
  IRONCHEF, 93
  SCHOOLMONTANA, 105
  SIERRAMONTANA, 104
  SOUFFLETROUGH, 97, 101
  STUCCOMONTANA, 106
Bittorrent, 222
Black bag operations, 96, 126, 128, 131, 134
Black Chamber, 48
BLARNEY
  AT&T, 73
Blaze, Matt, 56
BLINDDATE, 122
Block cipher, 27
  Advanced Encryption Standard, 232
  BATON, 12
  Data Encryption Standard, 81
  Rijndael, 27, 232
  Simon and Speck, 81
BLUEZEPHYR
  US-3277, 73
BND, *see* Germany
BOI, *see* United States of America
Braam, Conny, 83
Browder, Kalief, 67
Bugdoor, *see* Backdoor
BULLDOZER, 131
BULLRUN, 6, 42, 70, 81, 83, 193
  MAC address covert channel, 214
Bureau of Investigation, *see* United States of America

Cablegate, *see* WikiLeaks
CALEA, 42, 221
Campbell, Duncan, *see* Journalist
Canadian Hockey Hall of Fame, 63
CANDYGRAM, 113, 115
Catshadow, 248
CCN, 93
CDMA2000, 116, 118
  1900 MHz, 118
  WATERWITCH, 120
Censorship
  Resistance, 152, 249
Census data
  DEHOMAG, 48
  France, 48
  Germany, 48
  Netherlands, 48
  United States of America, 49

Central Intelligence Agency, 11, 37, 38, 41–43, 49, 50, 53, 54, 57, 60, 64–67, 73, 102, 171
  Lysergic acid diethylamide, 43
ChaCha20, *see* Stream cipher
ChaCha20Poly1305, *see* Authenticated-Encryption with Associated-Data
CHIMNEYPOOL, 112, 123, 126, 128, 140–144
Church Committee, 50, 51, 57
Church, Frank, 51
CIA, *see* Central Intelligence Agency
Circuit Switched Data, 112
Cisco Systems
  ASA 5505, 97
  ASA 5510, 97
  ASA 5520, 97
  ASA 5540, 97
  ASA 5550, 97
  ASA Firewall, 97
  OpenDNS, 149, 161
  PIX 500 Series, 97
  PIX Firewall, 97
Citizens' Commission to Investigate the FBI, 50
Clement, Andrew, 63
Cloudflare, 78
CM-I, 140
CM-II, 141
CM-III, 142
CNA, *see* Computer Network Attacks
CNE, 93, 95, 131, 144
CNSA, *see* Commercial National Security Algorithm Suite
CO-TRAVELER, 80
COBALTFALCON
  US-3354, 73
COINTELPRO, 11, 12, 50
Collateral murder, *see* WikiLeaks
Compulsion attack, 172
Confidential DNS, *see* Dmain Name System37
CONJECTURE, 135
Continous Wave
  Generator, 107, 110
Corporate Surveillance Partner
  Amazon, 191
  America Online, 74
  Apple, 74
  AT&T, 59, 78
  BellSouth, 78
  British Telecom, 39
  Broadcom, 143
  Dropbox, 74
  Facebook, 74
  FinFisher, 40
  Global Crossing, 39
  Google, 74, 78, 191
  HackingTeam, 40
  iMessage, 56
  International Telephone & Telegraph, 49
  Interoute, 39
  Juniper, 82
  Level 3, 39, 73
  Line, 56
  Microsoft, 74, 191
  Motorola, 117
  Narus, 40, 47
  Nehemiah Security, 65
  Nokia, 40
  PalTalk, 74
  Philips, 83
  QUEST, 59
  Raytheon, 66
  Raytheon (Blackbird Technologies), 66
  RCA Corporation, 49
  RSA Security, 81
  SBC, 78
  Siege Technologies, 65
  Signal, 56
  Skype, 74
  Stanford Research Institute (SRI International), 65
  Telegram, 56
  Threema, 56
  Verizon, 78
  Verizon Business, 39
  Viatel, 39
  Viber, 56
  Vodafone Cable, 39
  WeChat, 56

  - Western Union, 48, 49
  - WhatsApp, 56
  - Wickr, 56
  - Yahoo, 74
  - YouTube, 74
- COTTONMOUTH-I, 140
- COTTONMOUTH-II, 141
- COTTONMOUTH-III, 142
- Covert Channel, 140–143
- CPU
  - ARM7
    - 66 Mhz, 137
  - ARM9, 139
    - 400Mhz, 136
    - AT91SAM9G20, 136
  - Atmel
    - AT49BV322A, 137, 139
    - AT91R40008, 137
  - Cavium CPU backdoor, 72
  - PowerPC
    - 405GPR, 122
- Cracking DES, 81
- CROSSBEAM, 112
- CRUMPET COVERT NETWORK, 93
- CryptoAG, *see* Switzerland
- CSIDH, *see* Non-interactive Key Exchange
- CTX4000, 107
- Curve25519, *see* Non-interactive Key Exchange
- CW, *see* Continous Wave
- CYCLONE Hx9, 114

- DACRON, 39
- DANDERSPRITZ, 143
- Data Over Voice, 112
- DCS, 119
  - 1800 MHz, 119
- DCSNet, 221
- DDoS, *see* Distributed Denial of Service
- DEA, *see* United States of America
- DEHOMAG, 48
- DEITYBOUNCE, 94
- Dell
  - PowerEdge, 94, 95
- Democratic Republic of the Congo
  - Patrice Lumumba, *see* Assassination victim
- Deniability, 244
- DES, *see* Block cipher
- DHCP, *see* Dynamic Host Configuration Protocol
- DHT, *see* The GNU name system
- Digital signature, 31
  - Ed25519, 31–33, 194, 196, 199, 201, 205, 209, 210, 213, 215, 218
  - RSA, 157
  - SPHINCS, 4
  - SPHINCS+-128s, 205
- Direction Finding
  - ENTOURAGE, 116
  - HOLLOWPOINT, 116
  - WATERWITCH, 120
- DIRNSA, 8
- DITP, *see* Federal Bureau of Investigation
- DITU, *see* Federal Bureau of Investigation
- DLink, 65
- DNS, *see* Domain Name System
- DNS-SD, *see* Multicast DNS
- DNSCrypt, 161
- DNSCurve, 160, 162, 167
  - Cache, 160
  - Resolution example, 161
  - Server, 160
- DNSSEC, *see* Domain Name System
- DNT, 93, 97, 98, 100–102, 104–106, 132, 138, 140–143
- DOCSIS, 188
- DOJ, *see* United States of America
- Dolev-Yao, *see* Threat model
- Domain Name System, 149, 150, 154, 166–168
  - Anonymity, 151
  - Caching resolver, 150
  - Confidential DNS, 160, 161, 167
  - Distributed Denial of Service attack, 167
  - DNS Security Extensions, 19, 157, 166, 167, 199
  - DNS-over-TLS, 159, 160, 167

  - DNSKEY records, 157
  - DS records, 157, 162
  - ENCRYPT records, 161, 162
  - End-to-End security, 152
  - Example name resolution, 151
    - Confidential DNS, 162
    - DNSSEC, 157
    - QNAME minimization, 159
  - NS records, 150, 161
  - NSEC records, 157
  - NSEC5 records, 166
  - NXDOMAIN, 157, 166
  - Query name minimization, 149, 158, 159
  - Query privacy, 152
  - RRSIG records, 157
  - RSA signatures, 157
  - Security, 151
  - Surveillance, 19, 152
  - Unbound DNS server, 160
  - Zones, 150, 152
- Drake, Thomas, *see* Whistleblower
- DROPMIRE, 107, 109, 110, 129
- DROPOUTJEEP, 123
- DSpool, 249
- DTLS, 20
- DTMF, 112
- DUAL_EC, *see* Backdoors
- Dynamic Host Configuration Protocol, 199
  - Attack, 193, 215–217

- E-ink
  - WATERWITCH, 120
- East German Ministry for State Security, 43
- ECHELON, 52
- Ed25519, *see* Digital signatures
- Eduroam, 189
- EGSM, 118, 119
  - 900 MHz, 119
- Egypt
  - Technical Research Department, 40
- Elligator, 249
  - Elligator 2, 232, 234, 246
- Ellsberg, Daniel, *see* Whistleblower
- ENTOURAGE, 116
- EQGRP, *see* National Security Agnecy
- ETERNALBLUE, 92
- Ethernet, 18, 115, 143, 187, 188
  - 802.3, 187
- Ethics in Mathematics (EiM), 4–8
- Executive Order 12333, 85
- Exfiltration
  - SIM cards, 124, 125

- FAA702 operations, 73
- FAIRVIEW
  - AT&T, 73
- FBI, *see* Federal Bureau of Investigation
- Federal Bureau of Investigation, 8, 12, 37, 49, 56, 58, 75, 221
  - Citizens' Commission to Investigate the FBI, 50
  - Data Intercept Technology Program, 75
  - Data Intercept Technology Unit, 75, 77
  - Operational Technology Division, 75
- FEEDTROUGH, 99
- Fellwock, Perry, *see* Whistleblower
- File System
  - EXT2, 133
  - EXT3, 132, 133
  - FAT, 132
  - FAT32, 133
  - NTFS, 132, 133
  - UFS, 132, 133
- Find/Fix/Finish, 117
- FIREWALK, 135, 143
- First Crypto War, 56
- FISA, *see* Foreign Intelligence Surveillance Act
- FISC, *see* Foreign Intelligence Surveillance Court
- Five Eyes, 50, 52
- Flip Chip Module, 136
- FLUXBABBITT, 95
- Foreign Intelligence Surveillance Act, 9–12, 51–53, 61, 73
- Foreign Intelligence Surveillance Court, 52, 53, 60
  - Judge Roger Vinson, 53

Forward secrecy, 188, 196, 244
FOXACID, 144
FPGA
  Xilinx, 136, 137, 139
Free Software, 17, 218, 248
FreeBSD, 104–106, 133, 177
FREEFLOW, 123, 126, 128
Frequency
  10-4000 MHz, 116
  1000-2000 MHz, 107, 110
  1000-4000 MHz, 110
  1800 MHz, 113, 115, 119
  1900 MHz, 113, 115, 118, 119
  2100 MHz, 118
  850 MHz, 115, 119
  900 MHz, 113–115, 118, 119
FRIEZERAMP, 126, 128, 144
FRS, 116
FSB, *see* Russian Federation
Full-take audio surveillance, *see* Telephone surveillance
FVEY, *see* Five Eyes

GALAXY, 116
GCHQ, *see* Government Communications Headquarters
GECKO II, 93
GENESIS, 117
GENIE, 126, 128, 140–143
Germany
  Bundesnachrichtendienst, 41
GERONTIC, 39
Ghedi, Abdulaziz, 58
Gilmore, John, 56
GINSU, 131
Global Positioning System, 114, 115, 118, 119
Global System for Mobile Communications, 93, 112–116, 118, 119, 124–127
  1800 MHz, 119
  1900 MHz, 119
  850 MHz, 119
  900 MHz, 118, 119
  EGSM, 114, 118, 119
  GPRS, 112, 123, 128
  WATERWITCH, 120
GNS, *see* The GNU name system
GNU/Linux, 14, 33, 90, 122, 144, 171, 189, 198
  CAP_NET_ADMIN capability, 210
  Fedora, 121
  WireGuard, 172, 177
GODSURGE, 95
Google, 77, 191
  App Engine, 222, 229
  in the PRISM program, 74, 77
  public DNS service, 149
GOPHERSET, 124
Gottlieb, Sidney, 43
GOURMETTROUGH, 100
Government Communications Headquarters, 37, 54
  Belgacom, 41
  Deny / Disrupt / Degrade / Deceive, 46
  Honey-trap, 46
  Information Operations: disruption, 46
  Information Operations: influence, 46
  Online Covert Action, 46
  Operation Socialist, 41
  Technical disruption, 46
GPS, *see* Global Positioning System
Greenwald, Glenn, *see* Journalist
Grover's algorithm, 247
GSM, *see* Global System for Mobile Communications

Hale, Daniel, *see* Whistleblower
HALLUXWATER, 98
Hard drive firmware
  IRATEMONK, 132
Hardware, 95, 107–116, 118–122, 124, 125, 129–131, 135–137, 139, 140, 142, 143
Harrison, Sarah, *see* Journalist
Hash function, 26
  Argon2id, 231, 244, 246
  BLAKE2s, 173, 174, 181, 213, 231, 246
  SipHash2-4, 182

Hayden, Michael, 171
HC12, 136, 137
Headscale, 190, 191
HEADWATER, 102
Hewlett-Packard, 93
HIT, 102
HKDF, *see* HMAC-based Extract-and-Expand Key Derivation Function
HM, 140, 142, 143
HMAC, *see* Message Authentication Code
HMAC-based Extract-and-Expand Key Derivation Function, 29, 173, 174, 202, 231, 246
HOLLOWPOINT, 116
Hosenball, Mark, *see* Journalist
Host Protected Area, 133
HOWLERMONKEY, 135, 140, 142, 143
HPA, *see* Host Protected Area
HTTP, 20
HTTPS, 19
HTTPSlink2, 126, 128
Huawei Technologies, 98, 102
    Eudemon firewall, 98
    Eudemon firewall 1000, 98
    Eudemon firewall 200, 98
    Eudemon firewall 500, 98
    Router, 102

I2C bus, 93
IANA, *see* Internet Assigned Numbers Authority
IBM, 81
ICMP, 188
Identity intelligence, 9, 49
    Pocket litter, 33, 221, 243
IETF, *see* Internet Engineering Task Force
Implant
    Achilles, 66
    Aeris, 66
    AfterMidnight, 65
    Angelfire, 66
    Archimedes (LAN MITM), 65
    Athena, 65
    BadMFS, 66
    BANALMONKEY, 144
    BANANAGLEE, 97, 99–101, 144
    BANANALIAR, 144
    BARGLEE, 144
    BG, 144
    BIOS, 133
    BLATSTING, 144
    Boot ROM, 98, 102
    BothanSpy, 65
    Broken Promise, 65
    Brutal Kangaroo, 65
    BULLDOZER, 131
    BUZZLIGHTYEAR, 144
    Cisco Adaptive Security Appliance Firewall, 97
    Cisco PIX Firewall, 97
    CouchPotato, 66
    DarkMatter, 64
    DEITYBOUNCE, 94
    Dell PowerEdge, 95
    Dell PowerEdge 1950, 95
    Dell PowerEdge 2950, 95
    DoubleFantasy, 144
    Drifting Deadline, 65
    Dumbo, 66
    ExpressLane, 66
    FEEDTROUGH, 99
    FIREWALK, 135
    FLUXBABBITT, 95
    GODSURGE, 95
    GOURMETTROUGH, 100
    Gyrfalcon, 66
    HALLUXWATER, 98, 102
    HAMMERMILL, 144
    HEADWATER, 102
    Highrise, 66
    Host Protected Area, 133
    HOWLERMONKEY, 135
    HP Proliant 380DL G5, 93
    I2C bus, 93
    Intel JTAG, 95
    Intel Xeon 5100, 95
    Intel Xeon 5300, 95
    IRATEMONK, 132
    IRONCHEF, 93
    JETPLOW, 97, 144
    JUNOS, 104–106

Keyboard, 130
Keystone, 66
KONGUR, 131
MagicWand, 66
Marble Framework (anti-forensics), 64
Maxtor, 132
Microsoft Windows, 94, 95, 131
Microsoft Windows 2000, 134
Microsoft Windows 2003, 134
Microsoft Windows CE, 126
Microsoft Windows Mobile, 128
Microsoft Windows Transitory File system, 66
Microsoft Windows XP, 134, 138
NSA Playset, 146
OutlawCountry, 65
Pandemic, 65
PS2, 130
Samsung, 132
  F Series Smart Televisions, 64
SCHOOLMONTANA, 105
SCREAMINGPLOW, 97, 144
ScreenOS, 99–101
Scribbles (document watermarking), 65
Seagate, 132
SeaPea (MacOS kernel rootkit), 64, 66
Shadow, 65
Shattered Assurance, 65
SIERRAMONTANA, 104
SIM cards, 124, 125
Solartime, 66
SOUFFLETROUGH, 101
STRAITBIZZARE, 144
STUCCOMONTANA, 106
SURLYSPAWN, 130
SUTURESAILOR, 135
SWAP, 133
TAWDRYYARD, 111
TOTEGHOSTLY, 126, 128
UNITEDRAKE, 144
UR, 144
USB, 130
USBee, 146
VALIDATOR, 144
Weeping Angel (Samsung F Series Smart Televisions), 64
Western Digital, 132
WISTFULTOLL, 134
Wolfcreek, 66
YELLOWPIN, 135
ZESTYLEAK, 99, 144
IMSI-catcher, 113–115, 118, 119
INCENSOR, 144
Indigenous cryptography, 83
Information Brokers, 67
Innernet, 190, 191
Interception Capabilities 2000, 57
Interconnection bus
  I2C, 93
  JTAG, 95
  PCI, 131
Interdiction
  BULLDOZER, 131
  DEITYBOUNCE, 94
  FEEDTROUGH, 99
  GOPHERSET, 124
  GOURMETTROUGH, 100
  JETPLOW, 97
  JTAG bus hardware, 95
  MONKEYCALENDAR, 125
  PCI bus hardware, 131
  SCHOOLMONTANA, 105
  SIERRAMONTANA, 104
  SOUFFLETROUGH, 101
  STUCCOMONTANA, 106
Internet Assigned Numbers Authority, 150
Internet Corporation for Assigned Names and Numbers, 150
Internet Engineering Task Force, 149, 157, 160, 168
  DNSOP, 158
  DPRIVE, 158, 161
Internet Exchange Points, 63, 187
  DE-CIX, 187
  LINX, 187
Internet Protocol
  Addressing, 18, 192
  Dynamic Host Configuration Protocol, 188
  ICMP, 19

- IPsec, 189
  - Opportunistic Encryption, 190, 191
- IPv4, 18, 20, 150, 177, 189
- IPv6, 18, 20, 152, 177, 189
- Multicast, 188
- Packet marking, 203
- TCP/IP, 18, 188

IP, *see* Internet Protocol
IPsec, *see* Virtual Private Network
Iran-Contra, 58
Iraq war logs, *see* WikiLeaks
IRATEMONK, 132
IRONCHEF, 93
IXMaps, 63
IXP, *see* Internet Exchange Points

Jacobs, Bart, 41
Jefferson, Thomas, 5
JETPLOW, 97
Joint Threat Research Interest Group, 39, 43, 44, 46, 47
Journalist
- Duncan Campbell, 54, 57
- Glenn Greenwald, 43, 63, 67
- James Bamford, 54
- John Paul Oulo, 61
- Julian Assange, 47, 54, 61, 62, 70, 71
- Laura Poitras, 63, 64, 67
- Mark Hosenball, 54
- Oscar Kamau Kingara, 61
- Sarah Harrison, 71

JTAG, 95
JTRIG, *see* Joint Threat Research Interest Group
JUNIORMINT, 136
Juniper, 82
- DUAL_EC_DRBG, 82
- ISG 1000, 99, 100
- J-Series, 105
- JUNOS, 104–106
- M-Series, 104
- Netscreen, 99–101
- NOBUS, 82
- ns200, 99
- ns25, 99, 100
- ns50, 99, 100
- ns500, 99
- ns5xt, 99
- nsg5t, 100
- ScreenOS, 82
- SSG 300, 101
- SSG 320M, 101
- SSG 350M, 101
- SSG 500, 101
- SSG 520, 101
- SSG 520m, 101
- SSG 550, 101
- SSG 550m, 101
- SSG140, 100
- SSG20, 100
- SSG5, 100
- T-Series, 106

Kahn, David, 48
Kapor, Mitch, 56
Kenya, 79
Kerckhoffs, Auguste
- Kerckhoffs’ principle, 6

Keystroke recorder, 130
Kingara, Oscar Kamau, *see* Journalist
Kiriakou, John, *see* Whistleblower
Klein, Mark, *see* Whistleblower
Knot resolver, 160
Koblitz, Neal
- Another Look, 5

Kompromat, 11
KONGUR, 131

LANDSHARK, 115
LFS-2, 110
Linksys, 65
LITTLE, 39
Location tracking, 111
Loomis, Edward, *see* Whistleblower
LOUDAUTO, 108
LOVEINT, *see* Surveillance
LSD, *see* MKUltra
LTE, 118
Lumumba, Patrice, *see* Assassination victim
Luykx, Atul, 81

MAC, *see* Message Authentication Code

MACsec, 187, 189, 190
MAESTRO-II, 137
MAINWAY, 78
Mandela, Nelson, 83
Manning, Chelsea, *see* Whistleblower
Mass surveillance, 171, 187
Massive Data Repositories, 8, 9
Master Boot Record
  Substitution, 132
Maximator, 41
McLuhan, Marshall, 167
MCM, *see* Multi-Chip-Module Module
mDNS, *see* Multicast DNS
MDR, *see* Massive Data Repositories
Meinhof, Ulrike, 4
Menezes, Alfred
  Another Look, 5
Merkel, Angela, 4
Message Authentication Code, 27
  Authentication tag, 27
  Poly1305, 213, 231
Mexico, 79
Micron
  MT47H64M16, 136
  MT48LC8M32, 139
  MT48LCM32, 137
Microsoft
  Windows, 64, 65, 95, 131, 133, 134, 144
  Windows CE, 126
  Windows Mobile, 128
Misclosure, 243, 245
Mix network
  Katzenpost network, 248
    Shared Random Value, 231
MKUltra, 43
Mobile cellular phone, 126
MOCCASIN, 140
MONKEYCALENDAR, 124, 125
MONKEYROCKET
  US-3206:6T, 73
Monocypher, 232
MORECOWBELL, 149, 152, 153, 156
Motorola, 112, 117
Mullvad, *see* Virtual Private Network
Multi-Chip-Module Module, 137
Multicast DNS, 19, 198, 203
  .local., 198
  _opabinia._udp.local., 198
  DNS Service Discovery, 198, 199, 201
  Vula peer discovery, 192
MUSCULAR
  DS-200B, 73
MYSTIC, 79, 80

Nacchio, Joseph, 59
Name Service Switch module, 204
Namecoin, 163, 164, 166, 167
  Blockchain, 163
  Example name resolution, 164
NAT, *see* Network address translation
National Institute of Standards and Technology, 84
National Security Agency, 8, 20, 37, 49, 54, 56, 73, 81, 102, 149, 150, 153, 172
  Battle Damage Indication, 153
  Belgacom, 41
  Cavium CPU backdoor, 72
  Commercial National Security Algorithm Suite, 12
  Computer Network Attacks, 153
  Cryptographic Exploitation Solutions, 9
  DNS surveillance, 152
  Domestic surveillance of AT&T, 59
  Equation Group, 92
  Operation Socialist, 41
  Project MINARET, 51
  S32X, 144
  SORM compromise, 41
  Suite A cryptography, 12
  Suite B cryptography, 12
  Tailored Access Operations, 92
  Type 1 cryptography, 12
  XKeyscore, 9, 39, 40, 55, 56, 90, 151, 172, 221, 230
National Security Letter, 78
  Cloudflare, 78
National Telecommunications and Information Administration, 150
NCC, 93

NEBULA, 116, 118
Nebula, 191
Network address translation, 188, 222
NIGHTSTAND, 121, 190, 193, 211
NIGHTWATCH, 107, 109, 110, 129
NIKE, *see* Non-interactive Key Exchange
NIPRNET, 144
NIST, *see* National Institute of Standards and Technology
NOBUS, *see* Backdoors
Noise Protocol Framework, 181
  IK pattern, 21, 173, 179, 183, 198
Non-adversarial forensics, 17
Non-interactive Key Exchange, 29
  CSIDH, 192, 196, 198, 199, 202–205, 211, 213, 235, 244, 246, 247, 249
    CSIDH 512, 205
  Curve25519, 160, 172–174, 199, 201, 203, 204, 213, 232, 234, 235, 244, 246, 247, 249
  X25519, 29, 31, 196, 205, 232, 235, 246
NSA, *see* National Security Agency
NSL, *see* National Security Letter
NSS, *see* Name Service Switch module
NTIA, *see* National Telecommunications and Information Administration

OAKSTAR
  Unknown partners, 73
OE, *see* IPsec
OLYMPUS, 138
OMNIGAT, 131
Onion routing
  Tor network
    Shared Random Value, 231
Onyx interception system, *see* Switzerland
Open Hardware, 17
OpenVPN, *see* Virtual Private Network
Operation Rubikon, 41
Operation Thesaurus, 41
Operational psychology, 43
Operational security, 17
Opsec, *see* Operational security
ORANGEBLOSSOM
  US-3251, 73
ORANGECRUSH
  PRIMECANE, 73
  US-3230:0B, 73
Organization of this thesis, 13
OSS, *see* United States of America
OTD, *see* Federal Bureau of Investigation
Oulo, John Paul, *see* Journalist
OVERHEAD, 193

PACKAGEDGOODS, 153
Packet injection, 121
Paglen, Trevor, 63
PAKE, *see* Password Authenticated Key Exchange
PANDA, 222, 223, 244, 249
  PANDA′, 223, 228, 229, 249
  Verifpal model, 223, 226
Partnership
  BND, 41
  CIA, 41, 102
  GCHQ, 110
  NSA, 41
  Schweizer Nachrichtendienste, 41
Password Authenticated Key Exchange, 222
  Magic Wormhole, 222
  PANDA, 222
  SPAKE2, 222
PBD, *see* Persistent Backdoor
PCB, *see* Printed Circuit Board
PCS, 119
  1900 MHz, 119
Peck, Winslow, *see* Whistleblower
People’s Republic of China
  Cyberspace Administration of China, 37
  Office of the Central Cyberspace Affairs Commission, 37
  State Internet Information Office, 37
Persistent Backdoor
  GOURMETTROUGH, 100
  HALLUXWATER, 98
  HEADWATER, 102

  - IRATEMONK, 132
  - JETPLOW, 97
  - SCHOOLMONTANA, 105
  - SIERRAMONTANA, 104
  - SOUFFLETROUGH, 101
  - STUCCOMONTANA, 106
- PGP, *see* Pretty Good Privacy
- Philippines, 79
- Philips
  - PX-1000, 83
  - PX-1000cr, 83
- PHOTOANGLO, 107, 109, 110, 129
- PICASSO, 127
- Pike Committee, 50, 57
- PINNAGE, 39
- PIT, *see* TURBOPANDA, 98
- Pocket litter, 33, 221, 243
  - Threat model, 243
- Poitras, Laura, *see* Journalist
- Poly1305, *see* Message Authentication Code
- Pompeo, Mike, 66, 67, 70
- Post-quantum, 249
  - Rainbow, 84
  - Rainbow break, 84
  - Transitional security, 171
- Post-Quantum MACsec Key Agreement, 190
- Postal mail surveillance, 3
- Pournelle, Jerry
  - Pournelle's Iron Law of Bureaucracy, 49
- Power
  - 1 Watt, 115
  - 1 kWatt, 107
  - 2 Watt, 107, 110
- Pre-computation
  - Rainbow tables, 244
- Pre-shared Key, 171, 198
- Pretty Good Privacy, 56, 221
- Printed Circuit Board, 136
- PRISM, 77
  - America Online, 74
  - Apple, 74
  - Dropbox, 74
  - Facebook, 74
  - Google, 74, 77
  - Microsoft, 74
  - PalTalk, 74
  - Skype, 74
  - Tibet.net, 77
  - US-984XN, 9, 73, 74
  - Yahoo, 74
  - YouTube, 74
- Privacy-by-design, 221
- Project SHAMROCK, 49
- Protest, 4
- Protocols, 33
- PROTOSS GUI, 93
- PRP, *see* Pseudorandom permutation
- Pseudorandom permutation, 246, 247
- PSK, *see* Pre-shared Key
- Psychological operations, 43
- Python, 144, 192, 202, 211
  - PANDA offline attack, 229
  - REUNION-on-a-LAN, 249
  - SIBC, 211

- QFIRE, 144
- QUANTUM, 144
  - DS-300, 144
  - INCENSOR, 144
  - USJ-759, 144
  - USJ-759A, 144
  - USJ-799, 144
- Quantum computer, 8, 9
- Quantum-annoying, 249
- QUANTUMBISQUIT, 144
- QUANTUMBOT, 144, 154
- QUANTUMBOT2, 144
- QUANTUMCOOKIE, 144
- QUANTUMDNS, 144, 152–154, 156
- QUANTUMHAND, 144
- QUANTUMINSERT, 144
- QUANTUMMUSH, 144
- QUANTUMNATION, 144
- QUANTUMPHANTOM, 144
- QUANTUMSMACKDOWN, 144
- QUANTUMSPIM, 144
- QUANTUMSQUIRREL, 144
- QUANTUMTHEORY, 153, 154

- Radar, 107–111, 129, 130

Radio, 107, 110, 112–115, 118, 119, 121, 122, 135, 140–143
  802.11, *see* Wireless LAN
  WATERWITCH, 120
RAGEMASTER, 107, 109, 110, 129
Range
  12KM, 121
  30 KM, 119
  32 KM, 114
  Medium Range, 93
REMEDY, 39
Resistance, 4
RETURNSPRING, 132
REUNION, 249
  Curve25519, 246
  Elligator 2, 232, 234, 246
  Security goals, 245
  Space complexity, 248
  Threat model, 192, 243
  Verifpal model, 236, 237, 240
RF retro-reflector
  AM, 111, 130
  CTX4000, 107
  FM, 108
  FSK, 111, 130
  LOUDAUTO, 108
  NIGHTWATCH, 109
  PHOTOANGLO, 110
  RAGEMASTER, 129
  SURLYSPAWN, 130
  TAWDRYYARD, 111
  VGA, 109, 129
Ricochet, 222
Rijndael, *see* Block cipher
Roark, Diane, *see* Whistleblower
ROC, 93, 97, 101, 104–106, 126, 128, 132, 138
ROCKYKNOB, 112
Rogaway, Phillip
  Moral Character of Cryptographic Work, 6
Room 641A, 59
Rootkit, 93–95, 97, 99–101, 104–106, 124–126, 128, 131, 132, 134, 138
RSA, 157
Russian Federation
  Federal Security Services, 37
  System for Operative Investigative Activities, 41

Sabotage, *see* BULLRUN
SARATOGA, 144
Satellite phone
  Thuraya 2520, 126
Scapy, 172, 174
SCHOOLMONTANA, 105
Schulte, Joshua, *see* Accused whistleblower
SDR, *see* Software Defined Radio
SEAGULLFARO, 132
Secure Socket Layer, 20
Serge, Victor, 12
SERUM, 132
SEXINT, *see* Surveillance
Shadowbrokers, 92
Shannon, Claude
  Shannon entropy, 9
  Shannon's maxim, 6
Shared Random Value, 230, 231, 244
SHIFTINGSHADOW
  US-3217:MU, 73
Shor's algorithm, 180, 181, 183, 246
  and WireGuard, 180, 181, 183
  WireGuard, 179
SIERRAMONTANA, 104
SigAd, 9
SIGINT
  disabling, 214
  enabling, 214
SILVERZEPHYR
  STEELKNIGHT, 73
  US-3273:SK, 73
SIM Toolkit, 124, 125
Single Point of Failure, 191, 244, 248
  Availability, 191
  Bandwidth, 191
  Confidentiality, 191
SIS, *see* United States of America
SLICKERVICAR, 132
SMM
  DEITYBOUNCE, 94
  IRONCHEF, 93
  SCHOOLMONTANA, 105

  - SIERRAMONTANA, 104
  - SOUFFLETROUGH, 97, 101
  - STUCCOMONTANA, 106
- SMOKEYSINK, 144
- SMS, 113, 115, 123, 128
- SND, *see* Switzerland
- SNEAKERNET, 94, 133
- Snowden, Edward, *see* Whistleblower
- Software, 93–95, 97, 99–101, 104–106, 116, 123–126, 128, 131, 132, 134, 138
- Software Defined Radio, 116, 117
- Solaris, 133
- SOMALGET, 79, 80
- SOMBERKNAVE, 138
- SORM, *see* Russian Federation, 221
- SOUFFLETROUGH, 101
- SPARROW II, 122, 193, 211
- SPECULATION, 135, 140, 142
- SPHINCS+-128s, 205
- Sphinx
  - Single Use Reply Block, 248
- SPOF, *see* Single Point of Failure
- Spy Files, *see* WikiLeaks
- SRV, *see* Shared Random Value
- SSG, 132
- SSL, *see* Secure Socket Layer
- Stasi, 43
- STK, 124, 125
- STORMBREW
  - Verizon, 73
- STRAITBAZZARE, 132
- STRAITBIZARRE, 93, 123, 126, 128, 140–143
- Stream cipher, 27
  - ChaCha20, 27, 213, 231
- STREETCAR, 39
- STRIKEZONE, 135
- STRONGMITE, 93
- STUCCOMONTANA, 106
- SURB, *see* Sphinx
- SURLYSPAWN, 107, 130
- Surveillance, 3
  - Early history of, 48
  - Fusion Centers, 3
  - LOVEINT, 7
  - SEXINT, 7
- SUTURESAILOR, 135
- SWAP, 133
- Switzerland
  - CryptoAG, 41
  - Onyx interception system, 41
  - Schweizer Nachrichtendienste, 41
  - Secret Files Scandal, 41
- Symmetric encryption, 27
- Systemd, 202
  - System integration, 208
  - Vula discover service, 209
  - Vula organize service, 209, 210
  - Vula publish service, 209
  - Vula slice, 209

- Tailscale, 190, 191
- Taliban, 9
- Tamm, Thomas, *see* Whistleblower
- TAO, *see* National Security Agency
- TAWDRYYARD, 111
- TCP/IP, 6, 19, 160, 188
- Tcpcrypt, 191
- Telephone surveillance, 3
  - Content surveillance, 79, 80
  - Metadata surveillance, 79, 80
- The Codebreakers, 48
- The GNU name system, 164, 166, 167
  - Distributed hash table, 164–166
  - Example name resolution, 165
  - Privacy, 166
  - Zones, 166
- ThinThread, 78
- Threat model
  - Dolev-Yao, 194
  - Pocket litter, 243
  - REUNION, 243
  - Vula, 192
- Thunderbolt, 188
- Thuraya, *see* Satellite phone
- Tibet.net, 78
- Tice, Russel D., *see* Whistleblower
- TLS, 20, *see* Transport Layer Security
- TOFU, *see* Trust-on-first-use
- TOTECHASER, 126
- TOTEGHOSTLY, 126
- TOTEGHOSTLY 2.0, 128
- Transitionally post-quantum, 181, 188

Transport Layer Security, 20, 159, 160
TRINITY, 139, 140, 142, 143
Trust-on-first-use, 196
Tsarist Russia
  Okhrana, 12
TUNING FORK, 94, 132, 133
TURBINE, 144
TURBOPANDA, 98, 102
  Insertion Tool, 98, 102
TURBULENCE, 123, 126, 128
TURMOIL, 144
TWISTEDKILT, 133
TYPHON, 114, 118, 119
  WATERWITCH, 120
TYPHON HX, 119

UDP, 19, 20, 160, 188, 249
UMTS, 116, 118
  2100 MHz, 118
  WATERWITCH, 120
Unbound DNS server, 160
UniFi USG3, 72
United Kingdom
  Government Communications Headquarters, *see* Government Communications Headquarters
United States of America
  Armed Forces Security Agency, 49
  Black Chamber, 49
  Bureau of Investigation, 49
  Central Intelligence Agency, *see* Central Intelligence Agency
  Department of Justice, 12
  Drug Enforcement Agency, 37, 61
  Federal Bureau of Investigation, *see* Federal Bureau of Investigation
  National Security Agency, *see* National Security Agency
  Office of Strategic Services, 49
  Signal Intelligence Service, 49
UNITEDRAKE, 93, 132
Unlinkability, 245
USB, 140–143, 188

VAGRANT, 107–110, 129
VALIDATOR, 138
Vault 7, *see* WikiLeaks
VCARD, 222
VDSL2, 188
Verifpal, 205, 214, 223, 226
  PANDA model, 226
  REUNION model, 236, 237, 240
  Vula model, 205
Verizon, 51
VIEWPLATE, 110
Vinson, Roger, *see* Foreign Intelligence Surveillance Court
Virtual Private Network, 20, 171, 172
  Bypass, 203
  IPsec, 20, 21, 171
  Mullvad, 171
  OpenVPN, 20, 21, 171, 172, 181
  Sabotage, 21
  UDP, 172
  WireGuard, 20, 21, 171–174, 179, 181–183, 187, 190, 191, 198, 202, 203, 213
    and Shor’s algorithm, 180, 181, 183
    Android, 172
    Authenticated-Encryption with Associated-Data, 173, 174
    BLAKE2s, 173, 174
    ChaCha20Poly1305, 173, 174
    Curve25519, 173, 174
    FreeBSD, 177
    GNU/Linux, 172, 177
    Golang, 172, 182
    HKDF, 173, 174
    Informal protocol narration, 173
    iOS, 172
    MacOS, 172
    Message Authentication Code, 179
    Network performance, 207, 212, 213
    Noise Protocol Framework, 182, 183
    Packet formats, 176, 177
    Performance, 206
    Post-quantum WireGuard, 204

      - Pre-shared Key, 182, 183, 192, 201
      - Python, 172
      - Rust, 172, 182
      - Shor's algorithm, 179
      - SipHash2-4, 182
      - Traffic analysis, 174, 181
      - Transitionally post-quantum, 181
      - UDP, 177
      - Wireshark protocol dissector plugin, 172
- VITREOUS, 39
- VPN, *see* Virtual Private Network
- Vula, 190
  - Adversary evaluation, 217
  - Censorship, 216
  - Denial-of-Service, 197
  - discover, 203
  - organize, 203
  - Peer states, 194
  - Peer verification, 201, 216
  - Petnames, 203, 215
  - Policy State Engine, 199
  - Post-quantum protection, 204
  - Pre-shared Key, 204
  - Protocol, 201
  - publish, 203
  - Threat model, 194
  - Traffic analysis, 216
  - Usability, 197
  - Verifpal model, 205, 214

- WAGONBED, 93, 112
- WAGONBED 2, 112
- WASABI GSM Module, 112
- WATERWITCH, 120
- Whistleblower
  - Bill Binney, 60, 64
  - Chelsea Manning, 59, 61
  - Daniel Ellsberg, 53, 54
  - Daniel Hale, 59
  - Diane Roark, 60
  - Edward Loomis, 60
  - Edward Snowden, 5, 8, 41, 70, 83
    - Snowden archive, 7, 8, 21, 41, 55, 60, 64, 70–72, 79, 82
    - Summer of Snowden, 70
  - John Kiriakou, 60
  - Kirk Wiebe, 60
  - Mark Klein, 59, 60
  - Perry Fellwock, 52, 54
  - Reality Winner, 62
  - Russel D. Tice, 60
  - Terry J. Albury, 63
  - Thomas Drake, 60
  - Thomas Tamm, 60
  - Winslow Peck, 52, 54
- Whitney Museum, 64
- Wiebe, Kirk, *see* Whistleblower
- WiFi, *see* Wireless LAN
- WikiLeaks, 61, 62, 64–67, 70, 79
  - publications
    - Afghan War logs, 61
    - Cablegate, 61
    - Collateral Murder, 61
    - Iraq War logs, 61
    - Spy Files, 62
    - Vault 7:AfterMidnight, 65
    - Vault 7:Angelfire, 66
    - Vault 7:Archimedes, 65
    - Vault 7:Athena, 65
    - Vault 7:BothanSpy, 65
    - Vault 7:Brutal Kangaroo, 65
    - Vault 7:Cherry Blossom, 65
    - Vault 7:CouchPotato, 66
    - Vault 7:Dark Matter, 64
    - Vault 7:Dumbo, 66
    - Vault 7:Elsa, 65
    - Vault 7:ExpressLane, 66
    - Vault 7:Grasshopper, 64
    - Vault 7:Highrise, 66
    - Vault 7:Hive, 64
    - Vault 7:Imperial, 66
    - Vault 7:Marble Framework, 64
    - Vault 7:OutlawCountry, 65
    - Vault 7:Pandemic, 65
    - Vault 7:Protego, 66
    - Vault 7:Scribbles, 64
    - Vault 7:UCL_Raytheon, 66
    - Vault 7:Weeping Angel, 64
    - Vault 7:Year Zero, 66
- Winner, Reality, *see* Whistleblower
- WireGuard, *see* Virtual Private Network

Wireless LAN, 138, 188
  802.11, 114, 115, 118, 119, 121, 122, 138, 187
  Surveillance, 116, 121, 122
  WPA, 187, 193
  WPA1-Personal, 190
  WPA2, 187, 193
  WPA2-Enterprise, 189
  WPA2-Personal, 190
WISTFULTOLL, 132, 134
WPA, *see* Wireless LAN
WPA1, *see* Wireless LAN
WPA1-Personal, *see* Wireless LAN
WPA2, *see* Wireless LAN
WPA2-Enterprise, *see* Wireless LAN

X25519, *see* Non-interactive Key Exchange
Xilinx
  XC2V1000, 139
  XC2V500, 137
  XC4VLX25, 136
XKeyscore, *see* National Security Agency

YACHTSHOP
  BLUEANCHOR, 73
  US-3247:PJ, 73
Yardley, Herbert O., 48
YELLOWPIN, 135

Zersetzung, 43
ZESTYLEAK, 99
Zimmermann, Philip R., 56

Glossary

AFSA Armed Forces Security Agency of the United States of America. 49
ANT catalog NSA Advanced Network Technologies (ANT) catalog. 37

CAC Cyberspace Administration of China. 40
CES Cryptographic Exploitation Services department of the National Security Agency of the United States of America. 9, 143
CIA Central Intelligence Agency of the United States of America. 11, 37, 38, 41–43, 49, 50, 53, 54, 57, 60, 64–67, 72, 76, 171
CIA Center for Cyber Intelligence Center for Cyber Intelligence is a part of the Directorate of Digital Innovation of the Central Intelligence Agency of the United States of America. 66
CIA Directorate for Digital Innovation Directorate of Digital Innovation of the Central Intelligence Agency of the United States of America. 66
CIA Embedded Devices Branch Embedded Devices Branch is part of the CIA Center for Cyber Intelligence. 66
CIA Engineering Development Group Engineering Development Group is part of the CIA Center for Cyber Intelligence. 66
COMINT Communications intelligence. 54
Commercial National Security Algorithm Suite Replacement name for NSA Suite-B cryptography. 12
Computer Network Exploitation NSA Computer Network Exploitation. 143
Computer Network Operations NSA Computer Network Operations. 37

Data Network Technologies Data Network Technologies (S323) is a part of the NSA TAO. Their motto is "Nothing but net". 143
DEA Drug Enforcement Agency of the United States of America. 37, 60
DEHOMAG Deutsche Hollerith Maschinen Gesellschaft (DEHOMAG) company. Originally a German IBM subsidiary which became IBM after the end of the Second World War. 8
Department of Justice Department of Justice of the United States of America. 60
DITP FBI Data Intercept Technology Program. 74
DITU FBI Data Intercept Technology Unit. 74, 76

ECHELON Mid-to-Late 20th century worldwide intelligence gathering network. 52
ELINT Electronic intelligence gathering. 54

F6 Codename for Special Collection Service program. 72

FBI Federal Bureau of Investigation of the United States of America. 11, 37, 44, 50, 73, 76

FISA United States of America's Foreign Intelligence Surveillance Act. 9, 10, 12, 51–53, 60, 72

FISC United States of America's Foreign Intelligence Surveillance Court. 52, 53, 59

Five Eyes SIGINT agreement between United States of America, the United Kingdom, Canada, New Zealand, and Australia. Based on the UKUSA agreement of 1947. 52

FOIA United States of America's Freedom of Information Act. 54

FORNSAT Interception capabilities, programs, and systems relating to collections on uplinks and downlinks in outer space; operated by NSA GAO and their international partners. 42

FSB Federal Security Service of the Russian Federation. 40

GCHQ Government Communication Headquarters of the United Kingdom. 21, 39, 51, 54

HUMINT Human intelligence such as the use of confidential human sources. 43

IRA Internet Research Agency; a Russian Federation contractor. 46

IXP Internet Exchange Point. 63

JTRIG Joint Threat Research Intelligence Group of the GCHQ. 39, 43, 44, 46, 47

NIST National Institute of Standards and Technology of the United States of America. 83

NSA National Security Agency of the United States of America. 8, 9, 12, 20, 37, 39, 40, 49, 50, 54–56, 59, 71, 72, 76, 80, 89, 101, 149–153, 172, 221

NSL FBI National Security Letter. 77

OTD FBI Operational Technology Division. 74

Personal Security Products Personal Security Products such as anti-virus, anti-malware, and other security focused software. 143

Producer Designator Digraph Code for entity performing the actual surveillance for a given SIGAD. 72

RADINT Radar signals intelligence. 54

SIGAD SIGINT Activity Designator. 72

SIGINT Signals intelligence. 54

SIIO State Internet Information Office of China. 40

SORM System for Operative Investigative Activities Russian Federation interception system. 40

Special Collection Service NSA and CIA joint Special Collection Service program. 72

TAO NSA Tailored Access Operations. 37, 143

UTT NSA Unified Targeting Tool. 76

XKeyscore Global distributed interception and injection system. Surveillance collection and injection system with programmable search engine interface run by the NSA. 9, 39, 40, 89, 151, 221

Summary

Communication in a world of pervasive surveillance

Pervasive surveillance is a part of modern life. Cryptography is intentionally sabotaged by governments and other parties to be vulnerable to targeted and mass surveillance adversaries. Most commonly used network protocols do not provide meaningful protection against surveillance at all and anticipated advances in quantum computing are rarely addressed. There are adversaries recording network traffic today at a planetary scale and storing that traffic for later analysis. This unique window of time calls for the creation of transitionally-secure post-quantum cryptographic constructions for use in network protocols that will continue to protect data after a quantum computer is available.

This thesis presents new protocols and their analyses in terms of anonymity, privacy, security, and performance. The thesis assumes protocol operation in the context of either a targeted or a mass surveillance adversary, or both. This thesis analyzes how those adversaries will use access to an universal quantum computer in the future to attack today's network traffic including the cryptographic content of surveillance data.

This thesis draws on public-interest journalistic reporting to understand adversary capabilities and to provide representative example adversaries. The thesis proves properties of its designs using formal verification methods and the new software described in the thesis is released as Free Software for practical use by all.

Curriculum Vitae

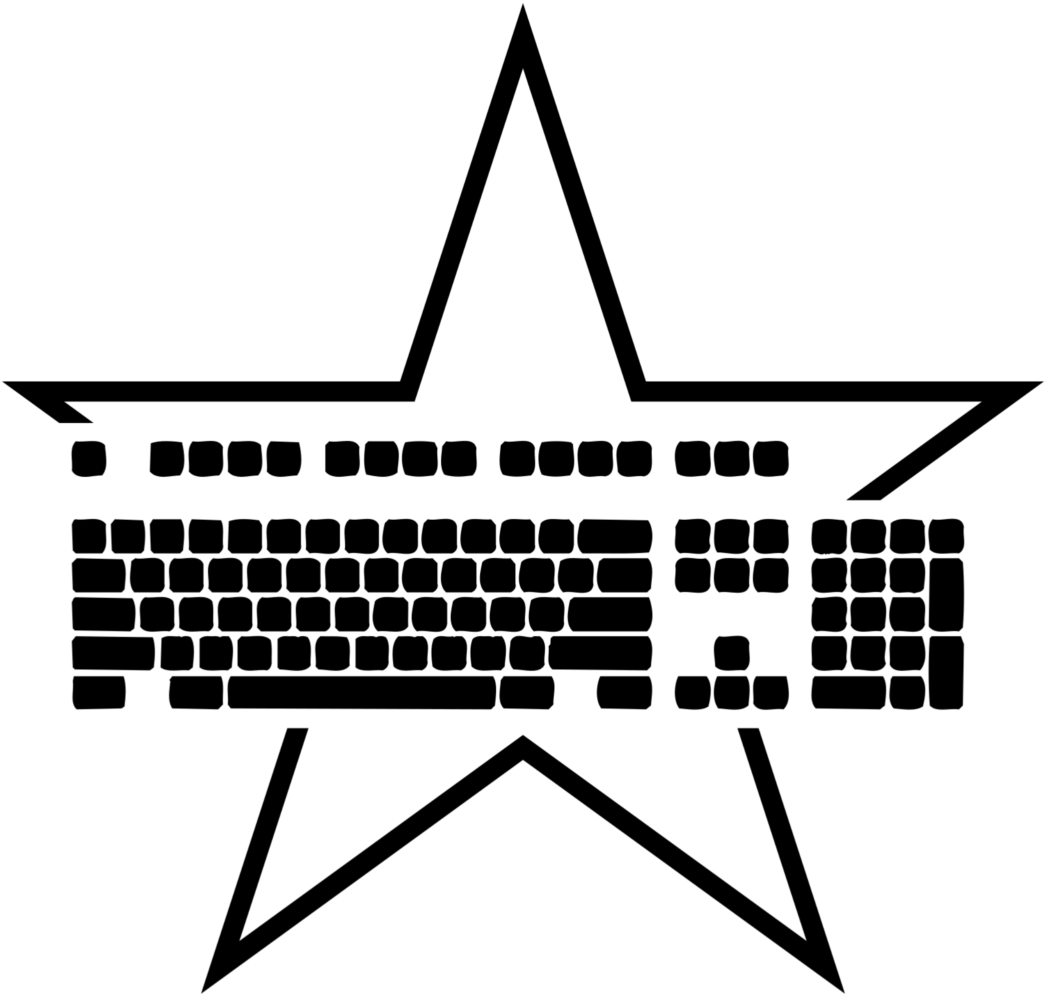
Jacob Raven Appelbaum was born on April 1st, 1983, in California, United States of America. He has worked as an artist, a journalist, and an academic researcher.

In the fall of 2015, he started his PhD project in the Cryptographic Implementations group at the Eindhoven University of Technology (TU/e) under the supervision of Daniel J. Bernstein and Tanja Lange. His research was funded by Daniel J. Bernstein's NWO Vici grant in computational number theory in cryptography and cryptanalysis. Jacob's work focused on post-quantum cryptographic protocol design as well as deployable Free Software cryptographic implementations to thwart surveillance. He additionally mentored students at the Masters level.

During the end of his PhD research time in the fall of 2021, he joined the Bern University of Applied Sciences (BFH) as a Lecturer in Bern, Switzerland, teaching and mentoring students about applied cryptography.

Jacob lives and works in Berlin, Germany with his wife and their three cats.

COURAGE IS CONTAGIOUS



LEAK MORE DOCUMENTS