The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation

Roland van Rijswijk-Deij, Kaspar Hageman, Anna Sperotto, and Aiko Pras

***Abstract*—The Domain Name System is a core Internet infrastructure that translates names to machine-readable information, such as IP addresses. Security flaws in DNS led to a major overhaul, with the introduction of the DNS Security Extensions. DNSSEC adds integrity and authenticity to the DNS using digital signatures. DNSSEC, however, has its own concerns. It suffers from availability problems due to packet fragmentation and is a potent source of distributed denial-of-service attacks.**

In earlier work we argued that many issues with DNSSEC stem from the choice of RSA as default signature algorithm. A switch to alternatives based on elliptic curve cryptography (ECC) can resolve these issues. Yet switching to ECC introduces a new problem: ECC signature validation is much slower than RSA validation. Thus, switching DNSSEC to ECC imposes a significant additional burden on DNS resolvers, pushing load toward the edges of the network. Therefore, in this paper we study the question: will switching DNSSEC to ECC lead to problems for DNS resolvers, or can they handle the extra load?

To answer this question, we developed a model that accurately predicts how many signature validations DNS resolvers have to perform. This allows us to calculate the additional CPU load ECC imposes on a resolver. Using real-world measurements from four DNS resolvers and with two open source DNS implementations, we evaluate future scenarios where DNSSEC is universally deployed. Our results conclusively show that switching DNSSEC to ECC signature schemes does not impose an insurmountable load on DNS resolvers, even in worst-case scenarios.

***Index Terms*—DNS; DNSSEC; elliptic curve cryptography; ECDSA; EdDSA; ECC**

I. Introduction

THE Domain Name System (DNS) is arguably one of the most crucial protocols on the Internet. Its main task is to translate human-readable names (such as 'www.utwente.nl') to machine readable information (such as IP addresses). Over the past decade, the DNS has been undergoing a major overhaul with the introduction of the DNS Security Extensions (DNSSEC). DNSSEC addresses a critical flaw in the DNS protocol: a lack of authenticity and integrity. This is done using digital signatures. While DNSSEC effectively addresses the lack of trust in the original DNS protocol, it is not without its own flaws. In earlier work, we have shown that:

- DNSSEC suffers from IP fragmentation. As DNSSEC responses are larger than 'classic' DNS, due to the inclusion of digital signatures, they may be fragmented at the IP level. Up to 10% of DNS resolvers on the Internet may not be able to deal with fragmented responses [1]. This can have consequences for users of these resolvers. Resolving popular DNSSEC-signed domains, such as `paypal.com`, may incur a performance penalty and in the worst case the domain may even become unreachable.
- DNSSEC can be abused for potent distributed denial-of-service attacks. Because DNS is susceptible to IP address spoofing, it can be abused in so-called amplification attacks. For 'classic' DNS the average amplification factor is around 6×, but DNSSEC makes things much worse, increasing the average amplification to around 50× [2]. This means that by sending 100 Mbit/s, attackers can mount an attack of 5 Gbit/s.

The root cause of these issues is the choice of RSA as default signature algorithm for DNSSEC. We showed that alternative signature schemes based on elliptic curve cryptography (ECC) effectively address the major issues in DNSSEC described above [3]. This is because ECC signatures are significantly smaller in size, leading to smaller DNS responses.

While switching DNSSEC to ECC-based signature algorithms is highly beneficial and solves serious issues in DNSSEC, it introduces a new problem: validation of ECC signatures is an order of magnitude slower than validation of the RSA signatures currently in widespread use in DNSSEC. This may have consequences for the global DNS infrastructure. Currently, using RSA, the most CPU intensive operation in DNSSEC is the signing process. This is performed at regular intervals by the DNS operators of signed domains. Validation of signatures is performed by recursive caching name servers ('DNS resolvers'). Thus, a switch from RSA to ECC-based signatures imposes a significant additional burden on DNS resolvers, effectively pushing the cost of cryptographic operations in DNSSEC to the edges of the network.

This paper addresses this new problem by answering the question: *What is the performance impact on DNSSEC validation of switching from RSA- to ECC-based signature algorithms?* We break this down into the following subquestions:

- *What is an upper bound on the number of signatures a resolver can validate on current hardware?*
- *How many signatures does a typical resolver validate at present?*
- *How would the number of signatures to validate increase for a growing global DNSSEC deployment?*
- *Based on these figures, can a resolver cope with the switch from RSA to ECC in current and future scenarios, where DNSSEC deployment becomes universal?*

R. van Rijswijk-Deij, K. Hageman, A. Sperotto and A. Pras are with the Design and Analysis of Communications (DACS) group at the faculty for Electrical Engineering, Mathematics and Computer Science of the University of Twente, Enschede, the Netherlands

R. van Rijswijk-Deij is also with SURFnet bv, the National Research and Education Network in Utrecht, the Netherlands

Manuscript received March 24, 2016; revised August 31, 2016.

A. Contribution

The main contribution of this paper is that we show whether validating DNS resolvers can handle the additional CPU load imposed by the validation of elliptic curve-based signatures. We introduce a novel model that predicts the number of signature validations a resolver needs to perform, given the number of queries it sends upstream to authoritative name servers. This model is then used to extrapolate how future growth in DNSSEC deployment will change the number of validations a resolver will need to be able to process. Based on measurements on four production resolvers, three for a major network operator and one for a medium-size university, and based on two popular open source DNS resolver implementations, we show that even if DNSSEC deployment grows to 100% in the future, the workload due to signature validations can be handled on a single modern CPU core. In the worst-case scenario, where the most CPU intensive ECC algorithm is used, the single core load due to signature validations would be less than 50% for full DNSSEC deployment at current workloads for a busy DNS resolver.

B. Related Work

Numerous past studies have looked at performance aspects of the DNS. Jung et al. [4] study the performance of DNS resolution from a client perspective, based on trace analyses and simulations. In particular, they study the effect of the Time-to-Live (TTL) of DNS records on cache effectiveness. Gao et al. [5] have more recently revisited the DNS from a resolver perspective. Compared to Jung, they found that the TTL for address (`A`) records had decreased significantly in the intervening ten years since Jung's study, but that the TTL of name server (`NS`) records remained stable. Wessels et al. [6] performed a measurement and simulation study with the purpose of studying how DNS resolver implementations impact high level DNS servers (i.e. at the root and TLD level). Koç et al. [7], finally, create and validate a model of the DNS. According to their paper, the purpose of their model is to study ongoing and future changes of the DNS, such as the introduction of DNSSEC or the growing deployment of IPv6.

More closely related to this work are past studies that have looked at the impact of DNSSEC on the domain name system. Wijngaards and Overeinder [8] were the first to study the impact of DNSSEC on DNS resolvers. In their work, they use simulations to quantify the computational overhead of validating digital signatures on DNS resolvers. Wijngaards and Overeinder's study is limited to signatures created using the RSA cryptosystem (which was the only viable option for DNSSEC at the time of their study). Where their work used a simulated DNS environment, our study uses a model that we feed with real world data from validating DNS resolvers. Migault et al. [9] carried out a number of performance tests related to DNSSEC, looking both at the impact on authoritative name servers as well as DNS resolvers. The focus of their work is to assess the impact of DNSSEC deployment for operators. Finally, Lian et al. [10] study if clients are protected by DNSSEC validation, and what problems clients can experience due to name resolution errors related to DNSSEC.

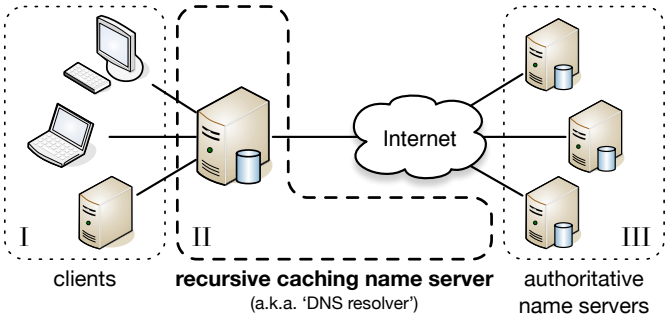

Fig. 1. Architecture of the DNS

C. Paper Organization

The remainder of this paper is organized as follows. Section II provides the necessary background information on DNS and DNSSEC. Section III describes the approach we took and introduces the model we developed to estimate the impact of ECC signature validation on future DNSSEC deployment scenarios. Section IV discusses the results obtained when applying the model to current and future DNSSEC deployment scenarios, based on real-world measurement data. Section V discusses open issues that may influence the adoption of ECC-based signature algorithms in DNSSEC. Section VI contains conclusions based on our findings with the model.

II. Background

A. DNS Architecture

The architecture of the DNS can be divided into three parts as shown in Figure 1. The first part consists of *clients*, shown on the left (I). Clients generally have what is called a *stub resolver* as part of the operating system or an application such as a web browser. The stub resolver performs DNS lookups on behalf of applications on the client. Stub resolvers are simple pieces of software that outsource DNS lookups to a *recursive caching name server*, shown in the middle (II). The DNS name space is a tree structure, starting with the root zone, followed by top-level domains (such as `.com`, `.net`, ...) one level down from the root, and second-level domains (such as `example.com`) below that, and so on. Recursive caching name servers perform the actual DNS lookup through a process called *recursion*. During recursion, they traverse the name space from top to bottom, communicating with *authoritative name servers*, shown on the right (III). Figure 2 shows a schematic example of a recursion. Recursive caching name servers also cache DNS responses, according to the *Time-To-Live* (TTL) field of the DNS record. Subsequent clients sending the same query will receive the cached response until the TTL expires. Caching ensures that the expensive process of recursion (in terms of network round trips) does not have to be performed for every query.

This paper studies the impact of DNSSEC on recursive caching name servers. These servers are often referred to as '*DNS resolvers*'. A DNS resolver that validates the digital signatures used in DNSSEC is then referred to as a '*validating DNS resolver*'. This terminology is used throughout the paper.

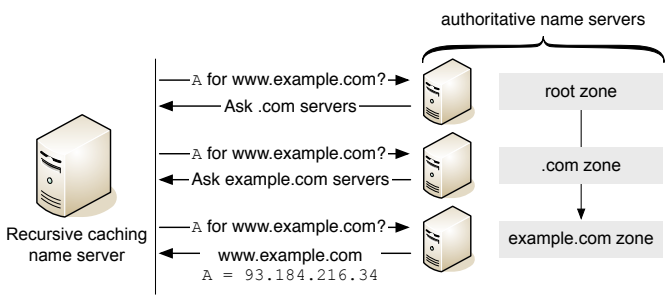

Fig. 2. DNS recursion for `www.example.com`

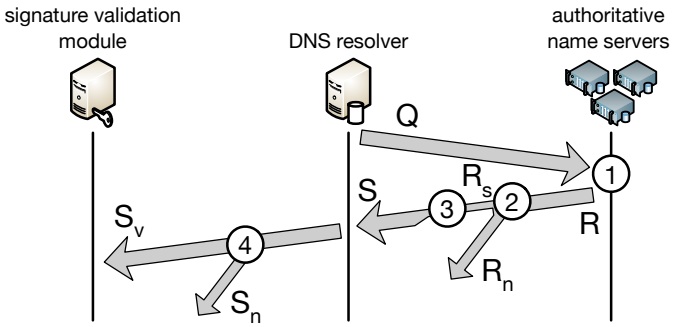

Fig. 3. Relation between outgoing queries and signature validation

B. DNSSEC and Signature Validation

DNSSEC is an extension to the DNS protocol. Its goals are to add authenticity and integrity to the DNS through the introduction of digital signatures. In DNSSEC, signatures are computed over coherent sets of resource records, called *RRsets*. An RRset consists of all records of a certain type and class for a certain DNS label in a DNS zone. For instance, consider the DNS zone snippet shown in Example 1.

label	*class*	*type*	*value*	
`example.com.`	IN	A	93.184.216.34	} RRset #1
	IN	RRSIG	*...signature data...*	← Signature #1
	IN	NS	a.iana-servers.net.	} RRset #2
	IN	NS	b.iana-servers.net.	
	IN	RRSIG	*...signature data...*	← Signature #2

Example 1. Signed DNS zone snippet

The snippet shows two RRsets. The first RRset contains a single `A` record that maps the label '`example.com`' to an IPv4 address. The second RRset contains two `NS` records that indicate the authoritative name servers for `example.com`. It also shows the two signatures that cover each of the RRsets. These signatures are contained in the `RRSIG` record type.

Validating DNS resolvers verify the signatures in the `RRSIG` records that accompany RRsets in a DNS response. In order to do this, they need to know the public key required to verify the signatures. DNSSEC has a special resource record type for public keys, called `DNSKEY`. When it first verifies signatures for a domain, a validating resolver will thus need to query for the `DNSKEY`. In most cases, DNSSEC-signed zones will contain `DNSKEY` records for two keys, a Key Signing Key (KSK) and a Zone Signing Key (ZSK). The KSK is only used to sign the `DNSKEY` RRset, the ZSK is used to create signatures on all other RRsets in the zone[1]. But one final piece is missing. How does the validating DNS resolver know it can trust this public key? That problem is solved by DNSSEC's chain of trust. The parent zone of each domain contains a *Delegation Signer* (`DS`) record that references the KSK of a domain. If the parent zone is signed, then this `DS` is signed, and thus the validating DNS resolver will only have to trust the parent zone's KSK. In DNSSEC, this chain of trust ends at the root zone of the DNS. Thus, validating DNS resolvers only have to trust the KSK of the root zone in order to validate signatures along the whole chain of trust.

[1]For a detailed discussion of the rationale behind this key model and its advantages and disadvantages, see [3], [11].

III. Approach

This section discusses the approach taken to predict the impact of ECC validation on validating DNS resolvers. It starts by analysing what factors play a role in the number of signature validations a DNS resolver has to perform. Then, a model is introduced that describes the relationship between the tasks of a DNS resolver and the number of signature validations. The section ends with a validation of the model against real world data measured on four validating DNS resolvers for two popular open source DNS resolver implementations.

A. Validation by DNS resolvers

To accurately model validating DNS resolvers, we first need to examine the factors that determine the number of signatures that a resolver needs to validate. Intuitively, one might assume that the prime determinant is the number of incoming queries received from clients. But in actual practice, this is not the case. A validating DNS resolver validates signatures in responses to queries it initiates. And while there is a relationship between the number of incoming queries from clients and the number of outgoing queries that the DNS resolver sends, in order to estimate the number of signatures that need to be validated it is sufficient only to consider the number of outgoing queries initiated by the DNS resolver.

Given the number of outgoing queries, Q, that a validating DNS resolver sends, there are four factors that determine the number of signatures, S_v, that it needs to validate. Figure 3 shows these factors, and they are described below:

1) Not all queries (Q) initiated by a DNS resolver result in a response (R) from an authoritative name server.
2) DNSSEC is not yet universally deployed. Currently, around 3% of second-level domains on the Internet have deployed DNSSEC[2]. Thus, not all responses will contain signatures. We designate responses that contain signatures with R_s and responses without signatures with R_n.
3) The number of signatures in a response varies, because:
   a) Responses may contain RRsets for multiple types. Consider, e.g., a `CNAME` response (an alias), that is returned to an `A` query; this response may also contain the `A` record(s) that the `CNAME` alias expands to if the authoritative name server knows about these `A` records.

[2]See http://www.internetsociety.org/deploy360/dnssec/statistics/

b) Next to the *answer* section, which contains the answers to a query, DNS responses may also have an *authority* section (for information about authoritative name servers for a domain) and *additional* section (for additional information, such as the addresses for name servers listed in the authority section). These two sections of a DNS response can also contain signatures.
c) DNSSEC has authenticated denial-of-existence to prove that a queried name and type do not exist. Such a proof may require multiple so-called `NSEC` or `NSEC3` records that are each accompanied by a signature.

We designate the number of signatures from responses S.

4) Finally, not all signatures need to, or can be validated. Signature validations can be cached by a resolver, or validation may be impossible because no full chain of trust to that particular signature exists. Also, signatures in the authority and additional sections of a response are not always validated. We refer to signatures that are validated as S_v and signatures that are not validated as S_n.

B. Model

The previous subsection discussed the factors that determine the number of signatures a DNS resolver needs to validate. The next step is to create a model of a validating DNS resolver that accurately predicts the number of signature validations (S_v) it needs to perform given a certain workload in terms of the number of queries it sends to authoritative name servers (Q). Thus, we want to find a function f, such that:

$$f : Q \rightarrow S_v \tag{1}$$

The factors discussed in the previous subsection each play a role in defining f. We hypothesize that each factor can independently be described using a function, and that a combination of these four functions approximates f. In other words:

$$\exists f_1, f_2, f_3, f_4 :$$
$$f_1 : Q \rightarrow R \qquad f_3 : R_s \rightarrow S$$
$$f_2 : R \rightarrow R_s \qquad f_4 : S \rightarrow S_v$$
$$f \approxeq f_4 \circ f_3 \circ f_2 \circ f_1$$

To gain an intuition about $f_1 \ldots f_4$, we examined empirical data collected on three validating DNS resolvers ($r_1 \ldots r_3$) operated by SURFnet[3]. We performed a live capture of traffic from clients to these DNS resolvers and replayed this traffic against an instrumented DNS resolver. A schematic overview of our measurement setup is shown in Figure 4. As the figure shows, traffic is captured live on the link between clients and the production DNS resolver. This traffic is instantly replayed to the instrumented DNS resolver. The number of queries from clients (Q_c) and responses to clients (R_c) as well as the distribution of DNS response codes is measured for both the production resolver and the instrumented resolver. These measurements are used to verify correct functioning of the instrumented resolver, by checking if the measurements of Q_c and R_c correspond within a small error margin.

[3]The National Research and Education Network in the Netherlands.

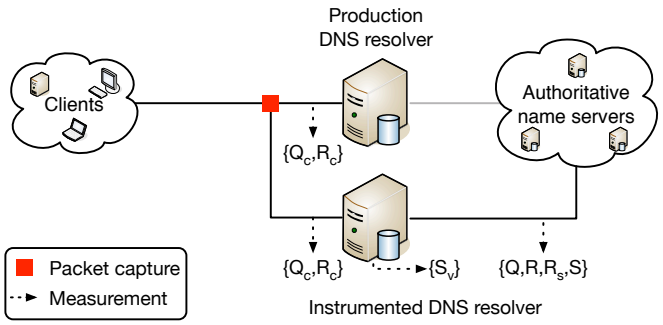

Fig. 4. Schematic overview of the measurement setup

To characterise $f_1 \ldots f_4$, the variables Q, R, R_s and S are measured on the network link between the instrumented resolver and upstream authoritative name servers. The variable S_v (the actual number of signatures verified) is measured through instrumented code in the DNS resolver software. Figure 5 shows four scatterplots that graph measurement data collected on the three resolvers r_1, r_2 and r_3 over a one week period. The axes show the average parameter value per second over 2-minute time slots. From top-left to bottom-right, plot (a) shows the relation between Q and R. Plot (b) shows the relation between R and R_s. Plot (c) shows the values for R_s and S. Finally, plot (d) shows the data for S and S_v.

The plots suggest a linear relationship between each pair of variables. In other words: they suggest that each function is of the form $f_n = ax + b$. The plots also illustrate that this relationship is weakest between R and R_s (f_2). This can be explained by three intuitions based on the fact that only a fraction of domains worldwide are DNSSEC-signed:

1) Query name popularity among clients influences this relationship; if more popular names are DNSSEC-signed, then the fraction of responses that contain signatures (R_s) will be higher. We expect this to vary between resolvers that have different client populations, and thus different query name popularity distributions. The resolvers used to develop our model, $r_1 \ldots r_3$ have different (albeit partially overlapping) client populations. As Figure 5b shows, they have differing values for R_s versus R.

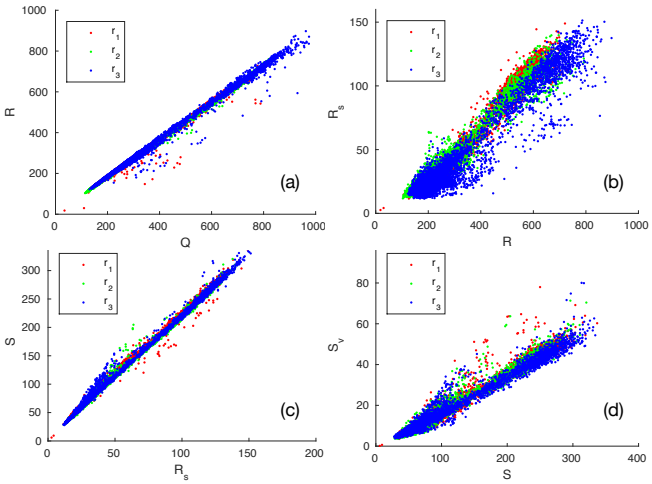

Fig. 5. Scatter plots showing the relationships between measured variables

2) DNSSEC-deployment across the Internet changes over time. This means that R_s will vary over time. Given current DNSSEC deployment trends, R_s will tend to grow over time. We will exploit this fact later when evaluating future DNSSEC deployment scenarios.
3) Query name popularity among clients varies over time; this can be explained in two ways. First, user behaviour varies during the day (with different interests at different times of day). Second, the distribution of client types varies during the day; automated systems tend to be active all day long, whereas human users tend to show diurnal behaviour (more activity during the day, less during the night). This can be seen in Figure 5b as a larger variability in R_s versus R than for the other measured relations.

Given that the plots suggest linear relationships between the variables, we define our model to be a set of parametrised linear functions $f_1 \dots f_4$ specified below:

$$f_1 : R = \bar{r}Q + \beta_1 \quad f_3 : S = \bar{s}R_s + \beta_3$$
$$f_2 : R_s = \alpha_s R + \beta_2 \quad f_4 : S_v = \alpha_v S + \beta_4$$

with:

$\bar{r}$ - the average number of responses per query
α_s - the fraction of responses with signatures
$\bar{s}$ - the average number of signatures per response
α_v - the fraction of signatures that is validated

These functions can then be combined to give f:

$$f : S_v = aQ + b$$
$$a = \alpha_v \bar{s} \alpha_s \bar{r}$$
$$b = \alpha_v(\bar{s}(\alpha_s \beta_1 + \beta_2) + \beta_3) + \beta_4$$

Finally, to use the model, the four parameters $\bar{r}$, α_s, $\bar{s}$ and α_v need to be estimated. We do this by performing linear regression on the measurement data obtained for each parameter. Two approaches for linear regression were considered. The first, Simple Linear Regression (SLR), fits a straight line through a set of points, such that the sum of the squared residuals (the distance between a point and the fitted line) is minimised. Although SLR has the smallest overall error, it is susceptible to outliers. As Figure 5 shows, all four variables have some outliers. For this reason, we also considered a second approach, the Theil-Sen Estimator [12], [13], which is robust in the presence of outliers. Comparison of the fit for both approaches shows negligible differences. Therefore, we chose to use the simplest approach, SLR, for the final model.

C. Model Validation

Before the model is used to analyse the impact of ECC signature validation on DNS resolvers, the predictive qualities of the model need to be validated first. In order to do this, we evaluate four criteria:

I. The model works for different DNS resolver implementations.
II. The model has stable properties over time; in particular, the values of $\bar{r}$, $\bar{s}$ and α_v remain relatively stable over longer periods of time and only α_s varies significantly as time progresses (as explained in the previous subsection).

Resolver	Operator	#Clients	Workload (queries/second) Average (24h)	Peak (1h)	Minimum (1h)
r_1	SURFnet	±125k	2623 qps	6062 qps	625 qps
r_2	SURFnet	±58k	781 qps	1441 qps	373 qps
r_3	SURFnet	±48k	568 qps	888 qps	223 qps
r_4	University	±11k	281 qps	520 qps	127 qps

TABLE I
RESOLVER CHARACTERISTICS

III. The model works for different client populations (i.e. for different operational DNS resolvers).
IV. The model is a good predictor of observed data.

Only if all four criteria are met can the model be used to make meaningful predictions about the number of signature validations required in future scenarios (where DNSSEC deployment grows). Each criterion is evaluated separately in the paragraphs below. Live data from four production DNS resolvers was used for the evaluation. Table I characterises each resolver in terms of estimated client population size and average, peak and minimum workload. Resolvers $r_1 \dots r_3$ (also used for the initial model development discussed in the previous subsection) are operated by SURFnet[3]. These resolvers are open for use by around 200 organisations (universities, research institutes, ...) connected to the SURFnet network. Resolver r_4 is operated by a medium-size university in the Netherlands. It serves the networks in the university buildings as well as the network in student dormitories on campus.

1) Resolver implementations: To test whether the model works for different DNS resolver implementations, we compared two popular open source packages. The first is Unbound[4], developed by NLnet Labs. Unbound is a resolver-only implementation, designed from the ground up to support DNSSEC validation, and optimised for speed. The second is BIND[5], the oldest and most popular[6] open source DNS implementation. BIND implements both resolver and authoritative name server functionality in a single application. Based on the measurement setup shown in Figure 4, two instrumented resolvers were deployed, one running Unbound, the other running BIND. Both resolvers ran simultaneously for a day and were fed live client data from production resolver r_1.

Figure 6 shows the measurement data and resulting parameter estimation based on simple linear regression. Three things stand out. First, as subfigures (a) and (c) show, parameters $\bar{r}$ and $\bar{s}$ are almost identical for the two resolver implementations. Given that both resolvers were sent the same query stream, this is as expected. Second, subfigure (b) shows a difference in the fraction of responses that contain signatures (α_s). This is due to implementation differences between Unbound and BIND. Third, as subfigure (d) shows, the most significant implementation difference between Unbound and BIND immediately becomes apparent when we perform the parameter estimation. BIND validates significantly more signatures given the same input queries (almost 3× more). The main takeaway is that the model works for the two different resolver implementations. As we will show in more detail

[4]http://unbound.net/, version 1.5.6 was used.
[5]https://www.isc.org/downloads/bind/, version 9.10.3 was used.
[6]Recent work suggests BIND has a 55% market share (https://indico.dns-oarc.net/event/24/session/11/contribution/11/material/slides/0.pdf).

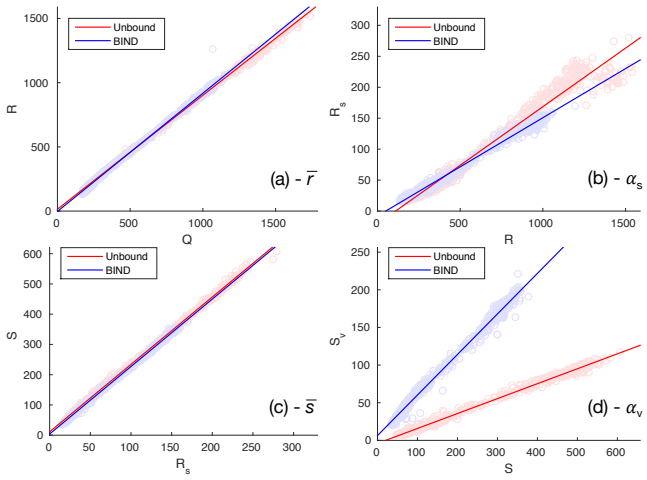

Fig. 6. Modelling two different open source resolver implementations

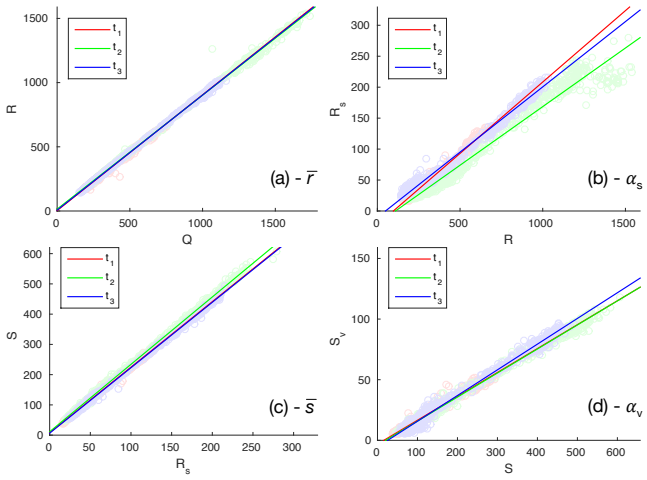

Fig. 7. Evaluating model parameters over time for r_1

when discussing criterion IV, the parameter estimation through linear regression leads to a good fit for both implementations.

2) Stability over time: As we wrote in the introduction to this section, we want to use the model to evaluate future DNSSEC-deployment scenarios. Predictions are only meaningful if the parameters of the model remain stable over time. In particular, $\bar{r}$, $\bar{s}$ and α_v should not change much over time. To evaluate if this is the case, we performed measurements for r_1 at three different times over a four month period.

Figure 7 shows the resulting scatter plots and parameter estimations through linear regression. Time t_1 is early October 2015, t_2 is early December 2015 and t_3 is late January 2016. In all three cases, data was captured over a 24 hour period on a working day. As the figure shows, the parameters we are particularly interested in vary little, thus the model is stable over time. The only noticeable fluctuations occur for $\bar{s}$ (c) and α_v (d) at t_2. We note that this fluctuation is self-canceling, because if $\bar{s}$ rises while α_v decreases the net effect on a prediction for the total model is negligible. A likely explanation for this fluctuation is that at t_2 slightly more responses from authoritative name servers were observed that had signatures in the optional *authority* and *additional* sections

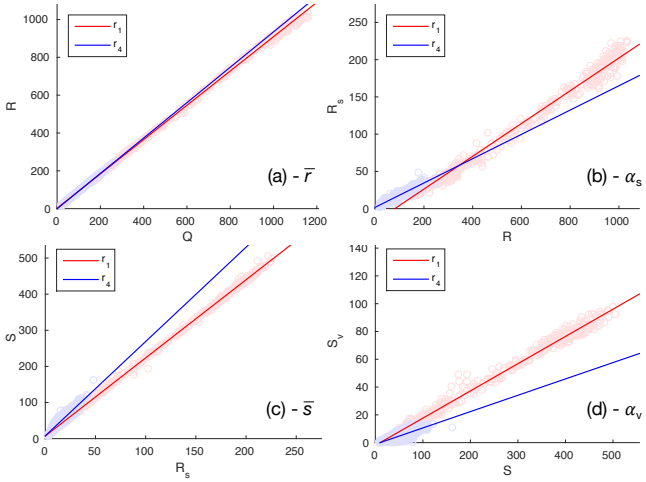

Fig. 8. Modelling resolvers with different client populations (r_1, r_4)

of the DNS response (see also Section III-A). Because these signatures are less likely to be validated, such a change would lead to $\bar{s}$ rising and α_v falling. Finally, note that α_s (b) differs significantly for $t_1 \ldots t_3$. This was expected, as this parameter is a function of DNSSEC deployment over time as well as query name popularity. We will be varying α_s in Section IV to simulate changes in global DNSSEC deployment.

3) Different client populations: To evaluate how well the model works for differing client populations, we performed parameter estimations based on measurements for all four resolvers $r_1 \ldots r_4$ described in Table I. Despite having different client populations of different sizes, as can already be seen in Figure 5, the parameter estimations for $r_1 \ldots r_3$ lead to almost the same values for $\bar{r}$, $\bar{s}$ and α_v. The only variation is observed for α_s, which, as mentioned above, does not influence the predictive capabilities of the model.

While we see few differences between $r_1 \ldots r_3$, there is a noticeable difference between these three resolvers and r_4. Figure 8 shows a comparison between the parameter estimation for r_1 and r_4. As the figure shows, only $\bar{r}$ is roughly the same, while the other two important parameters, $\bar{s}$ and α_v differ significantly. There are two explanations for this. First, the query name popularity for r_4 differs from that for r_1; just like the difference between times t_1,t_3 and t_2, this most likely means that r_4 receives more responses with signatures in the *additional* and *authority* sections. Second, and more importantly, the client population and query load for r_4 are much smaller than for the other three resolvers. This leads to a much smaller distribution in observed values. This is reflected in the scatter plots for measurement results in Figure 8, which show that the blue scatter points for r_4 are bunched much more tightly together towards the bottom left of each of the four subplots. As we will show when evaluating criterion IV next, this leads to a less accurate parameter estimation. This then, is a shortcoming of the model: it will tend to be less accurate for DNS resolvers with a lower query load.

4) Predictive qualities: Finally, we evaluate if the model is a good predictor of observed data. We do this by performing a goodness of fit test that checks how well the prediction from

Resolver	Time	DNS software	R^2	Criteria
r_1	t_1	Unbound	0.981	II (Fig. 7)
r_1	t_2	Unbound	0.966	I & II (Fig. 6 & 7)
r_1	t_2	BIND	0.976	II (Fig. 7)
r_1	t_3	Unbound	0.987	I (Fig. 6)
r_1	t_5	Unbound	0.976	III (Fig. 8)
r_4	t_4	Unbound	0.842	n/a
r_4	t_4	BIND	0.772	n/a
r_4	t_5	Unbound	0.801	III (Fig. 8)

TABLE II
R^2 FOR EVALUATION SCENARIOS

the model fits the observed data. In particular, we compare the number of signature validations predicted by the model to the observed number of signature validations and then compute the *coefficient of determination* (Equation 2).

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \qquad 0 \leq R^2 \leq 1 \qquad (2)$$

The value of R^2 is a measure for the fraction of the variance in the observed data that can be explained by the model. In general a higher value for R^2, closer to 1, indicates a better fit, and thus a better model. For each of the evaluations of the previous three criteria, we performed parameter estimation and input the resulting values into the model. Then, using the observed value for Q (the number of outgoing queries from the resolver), we used the model to predict how many signatures would need to be validated ($S_{v_{predicted}}$). We compared this to the number of signatures that were actually validated ($S_{v_{observed}}$) and computed R^2. All of these evaluations were performed over 24 hour periods on working days.

Table II shows the resulting R^2 values. The table includes one additional scenario, in which the performance of the model for Unbound and BIND is compared for queries to r_4 at t_4. The takeaway from the table is that the model is a good predictor in most cases, but as already observed during the evaluation of criterion III and reflected in the value of R^2, its predictive capabilities are diminished for r_4 because of its smaller population size and lower query load.

Summarising, based on the evaluation of the four criteria we conclude that the model is a good predictor of the number of signature validations (S_v) that need to be performed given a certain number of outgoing queries (Q) from a DNS resolver. We note, however, that the DNS resolver to which the model is applied must have a sufficiently large client population and a sufficiently high query load. Given that a large client population and high query load constitute a worst-case scenario in terms of the expected number of signature validations, this makes the model well-suited to analyse the impact of ECC signature validation on validating DNS resolvers.

IV. RESULTS

Based on the model introduced in the previous section, this section studies current and future DNSSEC deployment scenarios in order to quantify the impact a DNS-wide switch to elliptic curve-based signature algorithms will have on the global DNS. The section starts by describing the scenarios to be evaluated. Next, baseline benchmarks for the performance of elliptic curve-based signature algorithms are established, which will be used together with the scenario predictions to quantify the impact of a switch to ECC. Finally, the scenarios introduced at the beginning of the section are evaluated.

A. Scenarios

Our goal is to quantify the impact a DNSSEC-wide switch to ECC-based algorithms will have on the global DNS, and in particular what the performance impact is on validating DNS resolvers. To do this, we will evaluate two scenarios for current and future DNSSEC-deployment, described below:

I. *Current DNSSEC deployment* – this scenario evaluates what the performance impact would be if all domains that currently deploy DNSSEC would switch to an ECC-based signature algorithm overnight.
II. *Popular-domains-first growth to 100% DNSSEC deployment* – this scenario evaluates the performance impact of a growing DNSSEC-deployment in which the most popular domains (in terms of outgoing queries from the resolver) are the first to deploy DNSSEC. Effectively, this is the worst-case scenario as it requires the most signature validations at the shortest possible notice.

When the scenarios are evaluated, the model will be used to measure (for scenario I) or predict (for scenario II) the number of signature validations required in that particular scenario. This number is then compared against a benchmark figure indicating the number of signature validations that can be performed on a single modern CPU core for specific elliptic curve digital signature schemes. Just as in our earlier study on the use of ECC in DNSSEC [3], we examine multiple signature schemes. We include the two signature schemes currently standardised for use in DNSSEC, ECDSA P-256 and ECDSA P-384 [14], [15]. Next, we include the Ed25519 signature scheme based on twisted Edwards curves [16], [17]. Finally, new in this paper, we include a more recently introduced twisted Edwards curve-based scheme that is cryptographically stronger, Ed448 [18]. Both Ed25519 and Ed448 are currently being considered for standardisation by the IETF [19].

B. ECC Benchmarks

In earlier work [3] we relied on benchmarks from the eBACS project[7] to compare RSA and elliptic curve implementations. For this paper, we performed new benchmark tests. We did this because we explicitly wanted to incorporate recent performance improvements in ECC implementations for both ECDSA and EdDSA. Second, we wanted to standardise benchmarks to a single common CPU architecture, that is representative of modern server systems on which validating DNS resolvers are typically deployed.

The benchmarks were performed for five ECC implementations: three versions of OpenSSL and two independent high-performance implementations of Ed25519 and Ed448 respectively. OpenSSL versions were selected based on the following criteria: the first (0.9.8zh) we consider a 'legacy' implementation, the second (1.0.1f) is the mainstream implementation

[7] http://bench.cr.yp.to/index.html

	RSA				Signature algorithm and curve							
	1024-bit		2048-bit		ECDSA P-256		ECDSA P-384		Ed25519		Ed448	
Implementation	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ
OpenSSL 0.9.8zh	74221.3	508.2	22632.1	248.4	2694.8	29.0	1285.2	13.7	-	-	-	-
OpenSSL 1.0.1f	95909.5	721.1	28948.7	235.9	3684.8	26.7	1236.2	12.6	-	-	-	-
OpenSSL 1.0.2e	112516.0	903.5	35078.8	507.4	9786.6	75.7	1288.9	16.3	-	-	-	-
ed25519-donna	-	-	-	-	-	-	-	-	14162.4	212.2	-	-
ed448-goldilocks	-	-	-	-	-	-	-	-	-	-	4816.9	48.3

TABLE III
ECC BENCHMARKS (SIGNATURE VALIDATIONS PER SECOND, SINGLE CORE)

		Compared to*			
		RSA		ECDSA	
ECC algorithm	OpenSSL version†	1024	2048	P-256	P-384
ECDSA P-256	0.9.8zh	27.5	8.4	-	-
	1.0.1f	26.0	7.9	-	-
	1.0.2e	11.5	3.6	-	-
ECDSA P-384	0.9.8zh	57.7	17.6	-	-
	1.0.1f	77.6	23.4	-	-
	1.0.2e	87.3	27.2	-	-
Ed25519	(1.0.2e)‡	7.9	2.5	0.7	0.1
Ed448	(1.0.2e)‡	23.4	7.3	2.0	0.3

*the number means that the ECC algorithm is x times *slower*
†comparison of the ECC and RSA primitives for this OpenSSL version
‡independent implementations compared to this OpenSSL version

TABLE IV
COMPARISON OF RSA AND ECC SIGNATURE VALIDATION SPEED

that, for instance, ships with current Ubuntu and Debian Linux distributions and the third implementation (1.0.2e) is the newest stable release branch that incorporates significant performance improvements for ECDSA P-256. Benchmark data was collected by performing 100 independent speed tests for each of the five implementations. A single speed test consists of a 10-second run with continuous calls to signature validation functions, from which the average number of validations per second is calculated. The benchmark tests were run on an Intel Xeon E5-2695 v3 operating at 2.3GHz.

Table III shows the average results over 100 tests together with the standard deviation. The performance of ECDSA P-256 as well as 1024- and 2048-bit RSA improved significantly between OpenSSL versions. Interestingly, there was no performance improvement for ECDSA P-384. Table IV provides a speed comparison between different implementations. Note that from a cryptographic point of view, comparing 1024-bit RSA to ECDSA P-256 is comparing apples to oranges. The cryptographic strength of ECDSA P-256 is roughly equivalent to 3072-bit RSA [20]. The reason we make this comparison is because RSA 1024-bit is the most common signature type in DNSSEC at present, while ECDSA P-256 is the most attractive candidate to replace the current RSA-based schemes [3].

C. Scenario Evaluation

Before we evaluate the two scenarios, we make explicit what assumptions we made during the evaluation. We assume that:

A1. we only consider signature validations when calculating CPU use (i.e. we do not consider CPU use for other resolver functions, as this is highly dependent on, e.g., the number of clients, how many queries these send, ...);
A2. the DNS resolver runs on a single CPU core (worst-case scenario);
A3. there are no future advances in ECC implementation performance compared to the benchmarks in Section IV-B;
A4. DNSSEC policies do not change significantly[8].

In the following paragraphs we evaluate the two scenarios.

1) Current DNSSEC deployment: To evaluate this scenario, we looked at the peak signature validation rate observed on resolver r_1 (the busiest resolver). The highest rates measured were observed in the measurement at t_2. For the Unbound resolver implementation, validation peaked at 124 signatures per second, for BIND it peaked at 224 signatures per second. Looking at Table III, this is far below the maximum signature validation rates that can be achieved with each of the benchmarked ECC signature schemes. In other words, if all of the current DNSSEC deployments on the Internet were to switch to an ECC-based signature scheme overnight, this would not pose a problem for validating DNS resolvers, and would leave ample room for growth both in terms of DNSSEC deployment as well as an increase in query load on the resolver.

2) Popular-domains-first growth to 100% DNSSEC deployment: Next, we evaluated the worst-case scenario, where the most popular domains (in terms of number of queries for that domain) enable DNSSEC first. For this evaluation we measured query name popularity for outgoing queries from a DNS resolver. The reason that the query name popularity on the outgoing side was chosen is that this represents the absolute worst-case scenario for the resolver for which the distribution is measured. On the outgoing side, popularity is not just determined by popularity of the name among the client population of the resolver, but is also determined by the time-to-live (TTL) of records for certain names. Moderate popularity on the client side combined with a low TTL for DNS records will lead to a high number of outgoing queries (to refresh the cache). For the evaluation of this scenario, we measured the query name popularity for outgoing queries from the busiest DNS resolver r_1. Figure 9 shows the distribution of the query name popularity observed at t_4[9]. On the x-axis are domain names ranked in order of popularity (from highest- to lowest-ranked). The y-axis shows the number of queries for each domain name. Both axes were plotted using a logarithmic scale. The shape of Figure 9 resembles a Zipf distribution, commonly seen for many phenomena on the Internet [21]. In essence, in a Zipf distribution few entities (in this case domain names) account for the majority of observations (in this case queries). Jung et al. [4] also observed that query name popularity follows a Zipf distribution.

[8]As we note in [3], a switch to ECC-based signature schemes warrants simpler key management schemes with a single key per zone; this would result in fewer signature validations.

[9]The distribution is almost identical for other time periods and resolvers.

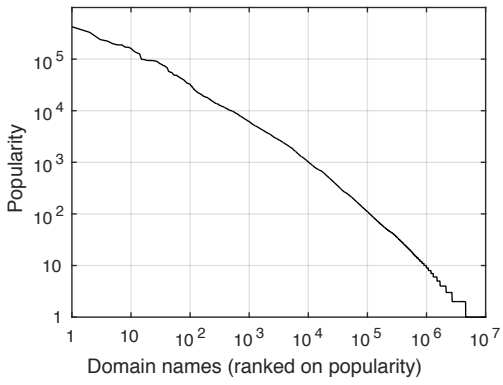

Fig. 9. Query name popularity (outgoing queries from r_1 at t_4)

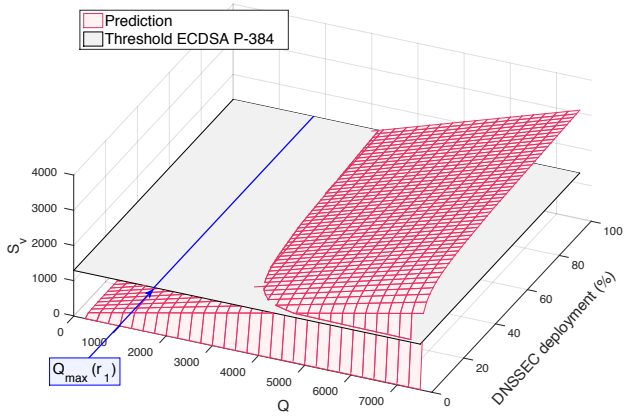

Fig. 10. Predicted validation requirements for r_1 running Unbound, compared to peak ECDSA P-384 performance

Using the observed distribution, and assuming that DNSSEC is deployed according to popularity rank from highest- to lowest-ranked, we calculate which fraction of queries would contain signatures under this assumption. In other words: we vary model parameter α_s based on the observed distribution. To calculate α_s for $x\%$ of domains deploying DNSSEC, we take the first n domain names that constitute $x\%$ of the total number of observed domains (d_{obs}). Then, with q_i being the number of queries observed for domain i and q_{obs} the total number of queries observed, Equation 3 gives the value for α_s. Using the estimated model parameters for the busiest resolver, r_1 at time t_4, we then predict the number of signature validations required (S_v) for Q outgoing queries.

$$n = x\% \cdot d_{obs} \qquad \alpha_s = \sum_{i=0}^{n} \frac{q_i}{q_{obs}} \tag{3}$$

Figure 10 shows this prediction for r_1 when running the Unbound DNS resolver implementation. The x-axis shows the number of outgoing queries (Q), the y-axis the required number of signature validations (S_v) and the z-axis the percentage of domains (ranked by popularity) that deploy DNSSEC. The figure compares the required number of validations to the worst-performing ECC signature scheme from Table III, ECDSA P-384. The intercept of the red surface (the prediction for S_v) with the gray plane represents where validations would

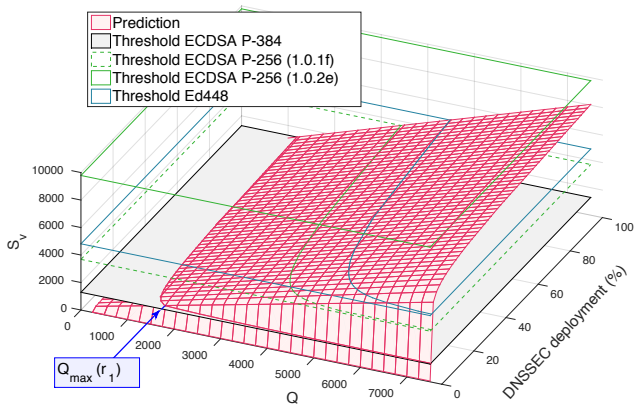

Fig. 11. Predicted validation requirements for r_1 running BIND, compared to peak ECDSA P-384 performance

account for 100% CPU saturation. The blue line indicates the maximum number of observed outgoing queries for r_1 over all measurements performed to date (1738 queries per second). As the figure clearly shows, even if 100% of domains on the Internet deploy DNSSEC using the ECDSA P-384 signature scheme (which is highly unlikely), the number of outgoing queries could almost double before signature validations account for 100% of the CPU use on a single core.

However, while the worst-case scenario indicates that for Unbound the margin for growth is generous, both in terms of DNSSEC deployment and in terms of the number of outgoing queries, the picture for BIND looks markedly different. Figure 11 shows the same plot when using BIND. As the plot shows, in this worst-case scenario BIND will quickly suffer CPU saturation, even if only a small proportion of popular domains deploy DNSSEC. Since it is unlikely that ECDSA P-384 will become the dominant implementation, however, we have also plotted lines for Ed448 (as high-security alternative) and ECDSA P-256 (as short term most likely candidate for deployment). These lines make clear that even for BIND, which clearly performs worse than Unbound in terms of the number of signatures it needs to validate for a given query load, 100% DNSSEC deployment is unlikely to lead to CPU saturation due to signature validations.

D. Summary and recommendations

Looking at the results of the scenario evaluations, it is clear that – from a performance point of view – even if DNSSEC deployment grows from the current 3% to 100%, and assuming worst-case conditions, the use of ECC-based signature schemes would not pose an insurmountable problem for validating DNS resolvers. This is a very positive result, as we have already shown in earlier work [3] that it is highly attractive to switch DNSSEC to ECC-based signature schemes.

Given these results, we strongly recommend that new DNSSEC deployments select ECC-based signature schemes and that existing implementers consider gradually switching to ECC-based signature schemes. For the short term, it is recommended to choose the ECDSA P-256 scheme. This offers excellent security properties combined with good performance

in terms of signature validation speed. Indeed, a major early adopter of ECC-based DNSSEC signing (CloudFlare) [22] has chosen to use ECDSA P-256. For the longer term we recommend considering Edwards curves-based signature schemes, in particular Ed25519 as future default algorithm and Ed448 for deployments with high security requirements.

V. Discussion

A. Representativeness of results

In this work, we have used data obtained at four separate DNS resolvers from two different operators. As we have shown in Section III-C, the simple linear model we have constructed works well in diverse environments. Nevertheless, one could argue that the set of resolvers we included in this study is far from representative of DNS resolvers worldwide. To mitigate this limitation, we have deliberately evaluated results against absolute worst-case scenarios, and have shown that even under these worst-case conditions the workload imposed on a validating DNS resolver as a result of signature validations for ECC-based signature schemes is far from prohibitive. Additionally, since we illustrated that our model is a good predictor, one can further vary parameters to simulate even worse conditions. For instance, assuming all queries lead to a response ($\bar{r} = 1.0$), all responses contain signatures ($\alpha_s = 1.0$) and that all signatures are validated ($\alpha_v = 1.0$), the average number of signatures per response ($\bar{s}$) would need to reach an average of almost 5 signatures per response (we observed an average around 2.1) before validation of ECDSA P-256 signatures would saturate a single CPU core at an outgoing query rate of 2000 queries per second.

B. Denial-of-service through CPU starvation

In discussions with a large ISP, the issue of denial-of-service attacks on validating DNS resolvers through CPU starvation was raised as a potential barrier to adoption of ECC-based signature schemes. They argued that an attacker could craft queries to a validating resolver that would lead to large numbers of validations. Given that, as Table III shows, validation of ECC-based signatures is highly CPU intensive (much more so than RSA), forcing large numbers of validations could lead to CPU starvation. In particular, an attacker could send queries for random non-existent names in a DNSSEC-signed domain, which would lead to authenticated denial-of-existence answers. Every signature over a proof of non-existence would then need to be validated by the resolver.

To assess the impact of such an attack, we performed two attack experiments:

1) Using a domain signed with ECDSA P-256 with a regular NSEC3 chain for authenticated denial-of-existence (i.e. a domain with a pre-computed fixed set of authenticated denial-of-existence records as specified in [23]).
2) Using a domain signed with ECDSA P-256 that uses 'NSEC3 White Lies' [24]. In essence, for such a domain every authenticated denial-of-existence answer is minimally enclosing and thus almost certainly unique[10].

[10]A recent draft RFC [25] suggests a similar approach the authors call 'Black Lies' that uses the `NSEC` record type and always only requires a single authenticated denial-of-existence proof.

The experimental attacks were performed against a test resolver with a single CPU core specifically set up for the experiment. Both tests were performed against an instrumented version of Unbound as well as an instrumented version of BIND. The first set of experiments did not result in a denial-of-service of any significance, neither for Unbound nor for BIND. While there is an initial peak workload, where CPU use peaks at 100%, the impact of the attack quickly diminishes as NSEC3 records are cached. Since in a regular NSEC3 chain there is a limited number of records, the attack is in essence self-limiting. The second set of experiments, however, did result in denial-of-service both for Unbound as well as for BIND. Because each authenticated denial-of-existence response is unique, caching does not help diminish the impact of the attack, requiring the resolver to expend CPU cycles validating the signatures in these responses. There was a notable difference in resilience against this type of attack. While Unbound's performance degraded, it reliably kept on serving answers from its cache for non-attack query traffic. Queries that required recursion, however, became very slow. After ceasing the attack, Unbound returned to normal operation within seconds. BIND, on the other hand, showed a significant performance degradation, also for responses to non-attack queries that it could have served from its cache. The degradation was such that this attack type can be considered a very effective denial-of-service against BIND. Worse, however, was that BIND did not recover and return to normal service after the attack was stopped. We did not investigate in detail what caused this breakdown in BIND.

We note that this attack could be much worse if domains signed with slower ECC signature schemes are abused (e.g. ECDSA P-384). While there are currently no mitigation mechanisms incorporated into validating DNS resolver implementations, we note that some form of rate limiting could be an effective countermeasure. Such a mechanism would need to keep track of clients or netblocks that require excessive numbers of signature validations and should rate limit queries from these clients or netblocks. It is likely that mechanisms currently implemented for Response Rate Limiting (RRL)[11] by authoritative name servers can be re-used.

C. Remaining hurdles for ECC adoption

While we recommend that operators switch to ECC-based signature schemes, a number of hurdles that may stand in the way of deployment still remain. Whether or not these hurdles are an actual barrier depends on many factors. In general, however, we believe these hurdles are rapidly being tackled by the Internet community. Open issues are:

- *TLD registry and registrar secure delegation support* – ECC signature schemes require changes to registry and registrar systems to support the creation of secure delegations. Many registry operators and registrars perform some form of validation on secure delegations that are submitted by domain owners; these checks will need to be updated to support ECC schemes. We note that

[11]https://kb.isc.org/article/AA-01000/0/A-Quick-Introduction-to-Response-Rate-Limiting.html

a number of large TLDs (including `.com`, `.net` and `.org`) already support both currently standardised ECC schemes, ECDSA P-256 and P-384.

- *Signer software support* – DNSSEC signing software needs to support ECC signature schemes. All mainstream implementations support ECDSA P-256 and P-384. Support for the newer algorithms currently being standardised (Ed25519 and Ed448), however, is almost non-existent. Operators may need to upgrade to newer versions of DNSSEC signer software to gain ECC support.
- *Validating DNS resolver support* – on the other side of the DNS, resolver software also needs to support validation of ECC-based signatures. Again, all mainstream implementations support validation of ECDSA P-256 and P-384 signatures, but support of newer algorithms is lacking. As shown in this paper, validation of ECC-based signatures does not require costly CPU upgrades.

One particular hurdle was raised by an operator in discussions during the research that led to this paper: algorithm rollover. For DNSSEC-signed domains that use RSA, there is a gradual upgrade path in case advances in cryptanalysis require stronger keys. RSA keys can simply be increased in size during regular key rollovers. For ECC-based signature schemes, however, this is not possible. Each ECC signature scheme has its own algorithm identifier in DNSSEC, that fixes the curve, and thus the key size. This is because in ECC signature schemes, the hashing algorithm used in signature creation is fixed and linked directly to the curve group size. Thus, for ECC signature schemes, if stronger keys are required this means an algorithm rollover will need to be performed. Algorithm rollovers (described in [26]) are considered more complex than key rollovers by operators. We note, however, that the likelihood of needing to perform an algorithm rollover because of serious advances in cryptanalysis that compromise ECC schemes such as ECDSA P-256 are small. Both European [27] as well as US [28] authorities currently recommend that 128-bit or higher cryptographic security is sufficient for the next 30 years at least. All ECC schemes discussed in this paper offer 128-bit security or more.

Finally, there has been conflicting advice from the NSA about the adoption of Suite B cryptographic algorithms[12]. In August 2015, the NSA recommended that implementers should no longer expend energy on a transition to Suite B algorithms, but should rather focus on implementing post-quantum cryptography (PQC)[13]. This led to speculation about the motivations behind this message from the NSA as well as the security of elliptic curve cryptography. Noted ECC experts Koblitz and Menezes provide a detailed analysis of the announcement by the NSA [29]. They make a strong case, based on the collective experience of the academic cryptography community over decades, that it is unlikely that there have been significant advances in cryptanalysis against ECC. Furthermore, as we showed in the introduction, the use of ECC-based signature schemes in DNSSEC offers significant benefits, tackling two major current issues with DNSSEC. The main benefits relevant in this context are smaller signatures and keys. None of the current PQC schemes offer these benefits. On the contrary; currently proposed PQC signature schemes all have key and signature sizes ranging from thousands to millions of bits [30], making them unsuitable for an application such as DNSSEC. This makes the NSA recommendation to focus on PQC implementation, rather than Suite B algorithms, impractical for DNSSEC. In light of these considerations, and taking into account the compelling arguments made by experts about ECC security, we stand by our earlier recommendation to switch to the use of ECC algorithms for DNSSEC.

[12]NIST curves P-256 and P-384 are part of Suite B.

[13]Algorithms resistant to a particular class of cryptanalysis that can be performed on a sufficiently powerful quantum computer.

VI. Conclusions

In this paper we have conclusively answered the question *can validating DNS resolvers handle the additional CPU load imposed by the validation of elliptic curve-based signatures*. We show that a set of linear relationships accurately models the behaviour of a validating DNS resolver. Using this model, we are able to reliably predict future developments in signature validation. By combining these results with benchmarks of various elliptic curve digital signature schemes we have shown that the CPU requirements for signature validations do not exceed the capacity of a single modern CPU core, even if the most CPU-intensive ECC scheme is used.

We discussed remaining hurdles that operators wishing to switch to ECC-based signature schemes may encounter, such as support for ECC keys by TLD registries and domain name registrars. We believe these problems to be transient; all are in the process of being resolved by the Internet community [31]. We also discussed one more serious concern, raised by an operator, which is the potential for denial-of-service on a validating DNS resolver through CPU starvation. This threat requires the attention of implementers of validating DNS resolver software, who may be able to implement effective countermeasures by applying some form of rate limiting.

As we have shown in earlier work [3], the use of elliptic curve digital signature schemes in DNSSEC has significant advantages. The use of ECC-based signature schemes can tackle serious issues in current DNSSEC deployments: amplification attacks and packet fragmentation. Given the findings of this paper, we strongly recommend that DNS operators considering deploying DNSSEC use ECC-based signature schemes. Additionally, existing operators should consider switching to ECC signature schemes as part of their regular upgrade cycle.

A. Future Work

As illustrated in Section IV-C, the number of outgoing queries from a resolver is one of the main determinants of the number of signature validations that a validating DNS resolver needs to perform. The outgoing query rate is a function of the number of queries from clients and query name popularity. Queries from clients will only lead to outgoing queries from the resolver if the answer is not already cached. Thus, although popular domains may be queried millions of times by clients, this does not necessarily lead to a high outgoing query rate. One development that may change this is the large scale

introduction of new generic top-level domains (gTLDs) [32]. If these new gTLDs prove to be popular, this may lead to a larger spread in names on the Internet, which may reduce the effectiveness of caching by resolvers and lead to higher numbers of outgoing queries. This should be studied in future work, as a larger number of outgoing queries will lead to a higher number of signature validations.

ACKNOWLEDGMENTS

This work was supported by the EU-FP7 FLAMINGO Network of Excellence Project (318488) and by SURF, the Netherlands collaborative organisation for ICT in higher education and research institutes.

The authors would like to thank the anonymous reviewers for their feedback. Furthermore, we thank Paul Ebersman of Comcast, Ondřej Surý and Jan Včelak of CZ.NIC, Wouter Wijngaards and Benno Overeinder of NLnet Labs, Ólafur Guðmundsson of CloudFlare, Rick van Rein of OpenFortress and Sharon Goldberg of Boston University for their input to, and feedback on our work.

REFERENCES

[1] G. van den Broek, R. M. van Rijswijk, A. Sperotto, and A. Pras, "DNSSEC Meets Real World: Dealing with Unreachability Caused by Fragmentation," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 154–160, Apr 2014.

[2] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and its potential for DDoS attacks," in *Proceedings of ACM IMC 2014*. Vancouver, BC, Canada: ACM Press, 2014.

[3] ——, "Making the Case for Elliptic Curves in DNSSEC," *ACM Computer Communication Review (CCR)*, vol. 45, no. 5, 2015.

[4] Jaeyeon Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS Performance and the Effectiveness of Caching," *IEEE/ACM Transactions on Networking*, vol. 10, no. 5, pp. 589–603, Oct 2002.

[5] H. Gao, V. Yegneswaran, J. Jiang, Y. Chen, P. Porras, S. Ghosh, and H. Duan, "Reexamining DNS From a Global Recursive Resolver Perspective," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 43–57, 2014.

[6] D. Wessels, M. Fomenkov, N. Brownlee, and K. Claffy, "Measurements and Laboratory Simulations of the Upper DNS Hierarchy," in *Passive and Active Measurement*, 2004, pp. 147–157.

[7] Y. Koç, A. Jamakovic, and B. Gijsen, "A global reference model of the dns," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 108–117, 2012.

[8] W. C. A. Wijngaards and B. J. Overeinder, "Securing DNS: Extending DNS servers with a DNSSEC validator," *IEEE Security and Privacy*, vol. 7, no. 5, pp. 36–43, 2009.

[9] D. Migault, C. Girard, and M. Laurent, "A performance view on DNSSEC migration," in *Proceedings of the 2010 International Conference on Network and Service Management, CNSM 2010*, 2010, pp. 469–474.

[10] W. Lian, E. Rescorla, H. Shacham, and S. Savage, "Measuring the Practical Impact of DNSSEC Deployment," in *Proceedings of the 22nd USENIX Security Symposium (USENIX Security 2013)*. Washington, D.C.: USENIX, 2013, pp. 573–588.

[11] H. Yang, E. Osterweil, D. Massey, S. Lu, and L. Zhang, "Deploying cryptography in internet-scale systems: A case study on DNSSEC," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 656–669, 2011.

[12] H. Theil, "A rank-invariant method of linear and polynomial regression analysis. I, II, III," *Koninklijke Nederlandse Academie van Wetenschappen*, vol. 53, pp. 386–392, 512–525, 1397–1412, 1950.

[13] P. K. Sen, "Estimates of the Regression Coefficient Based on Kendall's Tau," *Journal of the American Statistical Association*, vol. 63, no. 324, pp. 1379–1389, Dec 1968.

[14] P. Hoffman and W. Wijngaards, "RFC6605 - Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC," 2012.

[15] NIST, "FIPS PUB 186-4 - Digital Signature Standard (DSS)," *Processing Standards Publication*, 2009.

[16] D. J. Bernstein, P. Birkner, M. Joye, and T. Lange, "Twisted Edwards Curves," in *Progress in Cryptology - AFRICACRYPT 2008*. Springer Berlin Heidelberg, 2008, vol. 2, pp. 389–405.

[17] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B. Y. Yang, "High-Speed High-Security Signatures," *Journal of Cryptographic Engineering*, vol. 2, no. 2, pp. 77–89, 2012.

[18] M. Hamburg, "Ed448-Goldilocks, a new elliptic curve," *Cryptology ePrint Archive*, no. 625, 2015.

[19] O. Surý and R. Edmonds, "EdDSA for DNSSEC," 2016. [Online]. Available: https://tools.ietf.org/html/draft-ietf-curdle-dnskey-eddsa-00

[20] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for Key Management - Part 1: General (rev. 3)," *NIST SP800-57*, 2012.

[21] L. Adamic and B. Huberman, "Zipf's Law and the Internet," *Glottometrics*, vol. 3, pp. 143–150, 2002.

[22] R. van Rijswijk-Deij, M. Jonker, and A. Sperotto, "On the Adoption of the Elliptic Curve Digital Signature Algorithm (ECDSA) in DNSSEC," in *Proc. of the 12th International Conference on Network and Service Management (CNSM 2016)*. Montréal, Canada: IFIP, 2016.

[23] B. Laurie, G. Sisson, R. Arends, and D. Blacka, "RFC 5155 - DNS Security (DNSSEC) Hashed Authenticated Denial of Existence," 2008.

[24] R. Gieben and W. Mekking, "RFC7129 - Authenticated Denial of Existence in the DNS," 2014.

[25] F. Valsorda and O. Guðmundsson, "Compact DNSSEC Denial of Existence or Black Lies," 2016.

[26] O. Kolkman, W. Mekking, and R. Gieben, "RFC 6781 - DNSSEC Operational Practices, Version 2," 2012.

[27] N. Smart, "ECRYPT II Yearly Report on Algorithms and Keysizes 2011-2012," European Commission, Tech. Rep., 2012.

[28] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for Key Management - Part 1: General (rev. 4)," *NIST SP800-57*, 2016.

[29] N. Koblitz and A. J. Menezes, "A Riddle Wrapped in an Enigma," *IACR Cryptology ePrints*, no. 1018, 2015. [Online]. Available: https://eprint.iacr.org/2015/1018

[30] R. Perlner and D. Cooper, "Quantum Resistant Public Key Cryptography: a Survey," in *Proceedings of the 8th Symposium on Identity and Trust on the Internet - IDtrust '09*, 2009.

[31] D. York, O. Surý, P. Wouters, and O. Guðmundsson, "Observations on Deploying New DNSSEC Cryptographic Algorithms," 2016.

[32] T. Halvorson, M. F. Der, I. Foster, S. Savage, L. K. Saul, and G. M. Voelker, "From .academy to .zone: An Analysis of the New TLD Land Rush," in *Proceedings of ACM IMC 2015*. Tokio, Japan: ACM Press, 2015, pp. 381–394.

Roland van Rijswijk-Deij is a Ph.D. candidate at the University of Twente, the Netherlands, in the Design and Analysis of Communication Systems Group. He received an M.Sc. degree in Computer Science from the University of Twente in 2001. Roland also works for SURFnet bv, the National Research and Education Network in the Netherlands. His research interests include network security and network measurements, with a particular interest in DNS and DNSSEC.

Kaspar Hageman Kaspar Hageman received a B.Sc. in Computer Science and an M.Sc. in Telematics from the University of Twente, the Netherlands, in 2013 and 2015 respectively. He currently works at Nedap Healthcare, developing applications for healthcare professionals and clients in the Netherlands. His research interests include network measurements and modelling.

Anna Sperotto is assistant professor at the Design and Analysis of Communication Systems Group of the University of Twente, the Netherlands. She received a Ph.D. degree from the University of Twente, in 2010, with the thesis titled "Flow-based intrusion detection". Her research interests include network security, network measurements and traffic monitoring and modelling.

Aiko Pras is professor in the area of Network Operations and Management at the University of Twente, the Netherlands in the Design and Analysis of Communication Systems Group. His research interests include network management, monitoring, measurements and security. He chairs the IFIP Technical Committee on Communications Systems, and is Coordinator of the European Network of Excellence on Management of the Future Internet (FLAMINGO). He is steering committee member of several conferences, including IM/NOMS and CNSM, and series/associate editor of ComMag, IJNM and TNSM.