iOS-Sicherheit

iOS 8.3 oder neuer

Juni 2015

Inhalt

Einleitung

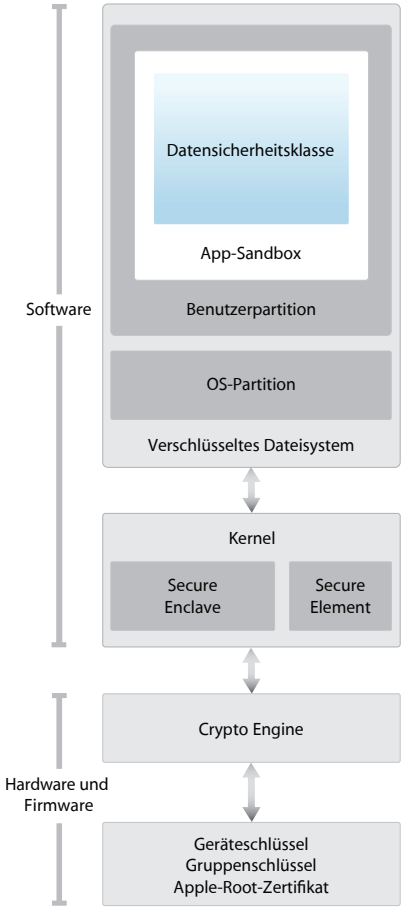


Diagramm der Sicherheitsarchitektur von iOS mit einem grafischen Überblick über die verschiedenen Technologien, auf die in diesem Dokument näher eingegangen wird

Apple hat bei der Entwicklung der iOS-Plattform die Sicherheit in den Mittelpunkt gestellt. Während der Entwicklung der besten mobilen Plattform aller Zeiten konnten wir auf jahrzehntelange Erfahrungen zurückgreifen, um eine völlig neue Architektur zu erstellen. Wir haben dabei auch an die Sicherheitsrisiken der Desktopumgebung gedacht und uns bei iOS für ein vollkommen neues Sicherheitskonzept entschieden. Wir haben innovative Funktionen entwickelt und eingebaut, mit denen die mobile Sicherheit optimiert und das gesamte System standardmäßig geschützt ist. iOS ist deshalb ein großer Entwicklungssprung für die Sicherheit bei Mobilgeräten.

Alle iOS-Geräte verbinden Software, Hardware und Dienste, die gezielt für die Zusammenarbeit entwickelt wurden und so maximale Sicherheit und eine transparente Benutzererfahrung bieten. iOS schützt nicht nur das Gerät und die gespeicherten Daten, sondern das gesamte Ökosystem, also alles, was Benutzer lokal, in Netzwerken und mit wichtigen Internetdiensten machen.

iOS und iOS-Geräte bieten modernste Sicherheitsfunktionen, die außerdem sehr benutzerfreundlich sind. Viele dieser Funktionen sind standardmäßig aktiviert, sodass IT-Abteilungen keine umfangreiche Konfiguration durchführen müssen. Zentrale Sicherheitsfunktionen, z. B. die Geräteverschlüsselung, können nicht konfiguriert werden, sodass Benutzer sie nicht versehentlich deaktivieren können. Andere Funktionen, z. B. Touch ID, erhöhen die Benutzerfreundlichkeit, da mit ihnen das Gerät einfacher und intuitiver geschützt werden kann.

In diesem Dokument finden Sie ausführliche Informationen darüber, wie unsere Sicherheitstechniken und -funktionen in der iOS-Plattform implementiert sind. Darüber hinaus hilft es Organisationen dabei, die Sicherheitstechniken und -funktionen der iOS-Plattform mit ihren eigenen Richtlinien und Verfahren zu kombinieren, damit ihre spezifischen Sicherheitsanforderungen erfüllt werden.

Dieses Dokument ist in die folgenden Themenbereiche unterteilt:

- **Systemsicherheit:** Die integrierte und sichere Software und Hardware, die die Plattform von iPhone, iPad und iPod touch bilden
- **Verschlüsselung und Datensicherheit:** Architektur und Design, die Benutzerdaten schützen, wenn das Gerät verloren oder gestohlen wird oder unbefugte Personen versuchen, es zu verwenden oder zu modifizieren
- **Sicherheit in Apps:** Die Systeme, die dafür sorgen, dass Apps sicher und ohne Gefährdung der Plattformintegrität ausgeführt werden können
- **Netzwerksicherheit:** Industriestandard-Netzwerkprotokolle die eine sichere Authentifizierung und die Verschlüsselung von Daten bei der Übertragung ermöglichen
- **Internetdienste:** Die netzwerkbasierte Infrastruktur von Apple für Nachrichten, Synchronisation und Backup
- **Gerätesteuerungen:** Methoden, die die unbefugte Verwendung des Geräts verhindern und es ermöglichen, das Gerät per Fernzugriff zu löschen, wenn es verloren oder gestohlen wurde
- **Datenschutzeinstellungen:** Möglichkeiten von iOS zum Steuern des Zugriffs auf Ortungsdienste und Benutzerdaten

Systemsicherheit

DFU-Modus aufrufen (Device Firmware Upgrade)
Die Wiederherstellung eines Geräts, das sich im DFU-Modus befindet, stellt wieder einen bekannten sicheren Zustand nur mit unverändertem, von Apple signiertem Code her. Der DFU-Modus kann manuell aufgerufen werden: Verbinden Sie das Gerät zunächst über ein USB-Kabel mit einem Computer und halten Sie die Home- und die Standby-Taste gedrückt. Lassen Sie die Standby-Taste nach 8 Sekunden wieder los, während Sie die Home-Taste weiter gedrückt halten. Hinweis: Wenn sich das Gerät im DFU-Modus befindet, bleibt der Bildschirm schwarz. Wenn das Apple-Logo erscheint, wurde die Standby-Taste zu lange gedrückt gehalten.

Die Systemsicherheit wurde so konzipiert, dass alle Kernkomponenten, sowohl Software als auch Hardware aller iOS-Geräte, sicher sind. Dazu gehören der Startvorgang, Softwareaktualisierungen und die Architektur „Secure Enclave". Diese Architektur ist von zentraler Bedeutung für die Sicherheit in iOS, ohne dabei die Benutzerfreundlichkeit zu beeinträchtigen.

Die enge Integration von Hard- und Software auf iOS-Geräten sorgt dafür, dass alle Systemkomponenten vertrauenswürdig sind und dass das System als Ganzes validiert wird. Vom ersten Systemstart über die Softwareaktualisierungen für iOS bis zu Apps anderer Anbieter wird jeder einzelne Schritt analysiert und sehr genau geprüft, damit Hardware und Software perfekt zusammenarbeiten und die verfügbaren Ressourcen optimal genutzt werden.

Sicherer Startvorgang

Jeder einzelne Schritt des Startvorgangs enthält kryptografisch von Apple signierte Komponenten, um die Integrität zu gewährleisten. Erst nach Verifizierung der „Chain of Trust" wird mit dem nächsten Schritt fortgefahren. Zu den signierten Komponenten gehören Bootloader, Kernel, Kernel-Erweiterungen und Baseband-Firmware.

Wenn ein iOS-Gerät eingeschaltet wird, führt der Anwendungsprozessor sofort Code aus dem Festspeicher, Boot-ROM genannt, aus. Dieser unveränderliche Code, Hardware-Vertrauensanker genannt, wird bei der Herstellung des Chips festgelegt und ist implizit vertrauenswürdig. Der Boot-ROM-Code enthält den öffentlichen Schlüssel der Apple-Root-Zertifizierungsstelle, mit dem überprüft wird, ob der Low-Level-Bootloader (LLB) von Apple signiert wurde, bevor er geladen werden darf. Das ist der erste Schritt in der Chain of Trust, bei der jeder Schritt überprüft, ob der nächste Schritt von Apple signiert wurde. Wenn der LLB seine Aufgabe abgeschlossen hat, wird der Bootloader der nächsten Stufe verifiziert und ausgeführt, der wiederum den iOS Kernel überprüft und ausführt.

Dieser sichere Startvorgang sorgt dafür, dass die unteren Software-Ebenen nicht unbefugt manipuliert werden können und iOS nur auf von Apple geprüften Geräten läuft.

Bei Geräten mit Zugriff auf das Mobilfunknetz verwendet das Baseband-Subsystem einen ähnlichen Prozess für sicheres Booten mit signierter Software und Schlüsseln, die vom Baseband-Prozessor überprüft wurden.

Bei Geräten mit einem A7-Prozessor (oder neuer) nutzt der „Secure Enclave"-Coprozessor ebenfalls einen sicheren Startvorgang, um zu überprüfen, ob seine Software von Apple überprüft und signiert wurde.

Kann ein Schritt in diesem Bootvorgang nicht geladen werden oder den nächsten Prozess überprüfen, dann wird der Startvorgang abgebrochen, und auf dem Display wird „Mit iTunes verbinden" angezeigt. Dies wird als Wartungsmodus bezeichnet. Wenn das Boot-ROM den LLB nicht laden oder überprüfen kann, wird der DFU-Modus (Device Firmware Upgrade) aufgerufen. In beiden Fällen muss das Gerät per USB mit iTunes verbunden und auf die Werksvoreinstellungen zurückgesetzt werden. Weitere Informationen zum manuellen Aufrufen des Wartungsmodus finden Sie unter https://support.apple.com/kb/HT1808?viewlocale=de_DE.

Autorisierung der Systemsoftware

Apple veröffentlicht in regelmäßigen Abständen Softwareaktualisierungen, die neu auftretende Sicherheitsbedenken behandeln oder neue Funktionen enthalten. Diese Aktualisierungen werden gleichzeitig für alle unterstützten Geräte bereitgestellt. Benutzer erhalten auf ihrem iOS-Gerät und über iTunes eine Mitteilung zu der iOS-Aktualisierung. Die Aktualisierungen werden drahtlos bereitgestellt, um Sicherheitslücken so schnell wie möglich schließen zu können.

Der oben beschriebene Startvorgang sorgt mit dafür, dass nur von Apple signierter Code auf den Geräten installiert werden kann. Damit Geräte nicht auf alte Betriebssystemversionen ohne neuere Sicherheitsaktualisierungen zurückgesetzt werden können, verwendet iOS einen Prozess namens *Systemsoftwareautorisierung*. Falls iOS auf eine ältere Version zurückgesetzt werden könnte, könnte ein Angreifer, der in den Besitz eines iOS-Geräts gelangt, es auf eine ältere Version von iOS zurücksetzen und Sicherheitslücken ausnutzen, die in neueren Versionen geschlossen wurden.

Bei Geräten mit einem A7-Prozessor (oder neuer) nutzt der „Secure Enclave"-Coprozessor ebenfalls die Autorisierung der Systemsoftware, um die Herkunft und Unverfälschtheit der Software sicherzustellen und die Installation eines älteren Betriebssystems zu verhindern. Siehe Abschnitt „Secure Enclave".

iOS-Softwareaktualisierungen können über iTunes oder drahtlos „Over The Air" (OTA) auf dem Gerät installiert werden. Über iTunes wird eine vollständige Kopie von iOS geladen und installiert. OTA-Softwareaktualisierungen laden anstelle des vollständigen Betriebssystems nur die für die Aktualisierungen benötigten Komponenten, wodurch die Netzwerkeffizienz verbessert wird. Außerdem können Softwareaktualisierungen auf einem lokalen Netzwerkserver, der den Caching-Dienst von OS X Server nutzt, bereitgehalten werden, sodass iOS-Geräte nicht auf die Apple-Server zugreifen müssen, um die für die Aktualisierung notwendigen Daten zu erhalten.

Bei der Aktualisierung von iOS stellt iTunes (bzw. bei einer OTA-Softwareaktualisierung das Gerät selbst) eine Verbindung mit dem Apple-Server für die Installationsautorisierung her und sendet eine Liste kryptografischer Kennzahlen für jeden Teil des Installationspakets, der installiert werden soll (z. B. LLB, iBoot, der Kernel und das OS-Image), einen Anti-Replay-Zufallswert (Nonce) und die eindeutige Kennung des Geräts (ECID).

Der Autorisierungsserver vergleicht die Liste der Kennzahlen mit den Versionen, deren Installation erlaubt ist, und fügt bei einem Treffer die ECID zu den Kennzahlen hinzu und signiert das Ergebnis. Der Server überträgt beim Aktualisierungsvorgang einen kompletten signierten Datensatz an das Gerät. Mit der hinzugefügten ECID wird die Autorisierung für das anfragende Gerät „personalisiert". Indem nur bekannte Kennzahlen autorisiert und signiert werden, stellt der Server sicher, dass die Aktualisierung genau wie von Apple bereitgestellt durchgeführt wird.

Die „Chain-of-Trust"-Evaluierung beim Start überprüft, ob die Signatur von Apple stammt und ob die Kennzahl des von der Festplatte geladenen Objekts in Kombination mit der ECID des Geräts von der Signatur abgedeckt wird.

Mit diesen Schritten wird sichergestellt, dass die Autorisierung gerätespezifisch ist und dass keine alte iOS-Version von einem Gerät auf ein anderes kopiert werden kann. Die Nonce verhindert, dass ein Angreifer die Antwort des Servers sichern und sie dafür verwenden kann, ein Gerät zu manipulieren oder die Systemsoftware anderweitig zu verändern.

Secure Enclave

Die Sicherheitsarchitektur „Secure Enclave“ befindet sich in einem Coprozessor, der in den Prozessoren der A-Reihe seit Apple A7 integriert ist. Die Secure Enclave-Architektur verwendet einen eigenen sicheren Startvorgang und eine personalisierte Softwareaktualisierung, die vom Anwendungsprozessor unabhängig sind. Diese Architektur stellt sämtliche kryptografischen Verfahren für die Schlüsselverwaltung zum Schutz von Daten bereit und garantiert den Schutz der Daten, selbst wenn der Kernel kompromittiert wurde.

Secure Enclave nutzt einen verschlüsselten Speicher und verfügt über einen Hardwarezufallszahlengenerator. Die Architektur verwendet einen von Apple modifizierter L4-Microkernel. Die Kommunikation zwischen der Secure Enclave und dem Anwendungsprozessor findet isoliert in einem interrupt-gesteuerten Postfach und Shared-Memory-Datenpuffern statt.

Jedem Secure Enclave wird bei der Herstellung eine eigene eindeutige Kennung (UID, Unique ID) zugewiesen, auf die andere Teile des Systems nicht zugreifen können und die Apple nicht bekannt ist. Wenn das Gerät startet, wird ein temporärer Schlüssel erstellt, der mit der UID verknüpft ist und verwendet wird, um den Teil des Speicherplatzes des Geräts für die Secure Enclave zu verschlüsseln.

Zusätzlich werden Daten, die von Secure Enclave im Dateisystem gespeichert werden, mit einem Schlüssel verschlüsselt, der mit der UID und einem Anti-Replay-Zähler verknüpft ist.

Die Secure Enclave ist zuständig für die Verarbeitung der Fingerabdruckdaten des Touch ID-Sensors. Sie bestimmt, ob der Fingerabdruck mit einem der registrierten übereinstimmt, und ermöglicht anschließend den Zugriff oder Kauf im Namen des Benutzers. Die Kommunikation zwischen dem Prozessor und dem Touch ID-Sensor findet über einen SPI-Bus (Serial Peripheral Interface) statt. Der Prozessor leitet die Daten an die Secure Enclave weiter, kann sie aber nicht auslesen. Sie werden verschlüsselt und mit einem Sitzungsschlüssel authentifiziert, der mit dem gemeinsamen Schlüssel des Geräts erstellt wird. Dieser wird für den Touch ID-Sensor und die Secure Enclave bereitgestellt. Beim Austausch des Sitzungsschlüssels wird AES Key Wrapping verwendet, bei dem beide Seiten einen zufälligen Schlüssel bereitstellen, aus denen der Sitzungsschlüssel erstellt wird und der die AES-CCM-Transportverschlüsselung nutzt.

Touch ID

Touch ID ist das Fingerabdrucksensorsystem, mit dem ein sicherer Zugriff auf das Gerät schneller und einfacher möglich ist. Diese Technologie liest Fingerabdruckdaten aus jedem beliebigen Winkel und erfasst den Fingerabdruck des Benutzers nach und nach immer genauer. Dabei erweitert der Sensor die Fingerabdruckdarstellung, wenn bei jeder Nutzung zusätzliche überlappende Knoten identifiziert werden.

Touch ID vereinfacht die Verwendung längerer, komplexerer Codes, da die Benutzer diese weniger häufig eingeben müssen. Touch ID macht die codebasierte Sperre benutzerfreundlicher. Sie wird zwar nicht vollständig ersetzt, kommt aber seltener zum Einsatz. Der sichere Zugriff auf das Gerät erfolgt innerhalb sinnvoller Grenzen und zeitlicher Beschränkungen stattdessen über Touch ID.

Touch ID und Codes

Um Touch ID zu verwenden, müssen Benutzer ihr Gerät so einrichten, dass zum Entsperren ein Code benötigt wird. Wenn Touch ID beim Scannen einen registrierten Fingerabdruck erkennt, wird das Gerät entsperrt, ohne dass nach dem Code für das Gerät gefragt wird. Der Code kann immer anstelle von Touch ID verwendet werden und ist unter den folgenden Umständen nach wie vor erforderlich.

- Das Gerät wurde gerade eingeschaltet oder neu gestartet.
- Das Gerät wurde seit über 48 Stunden nicht mehr entsperrt.
- Das Gerät wurde per Fernzugriff gesperrt.
- Es wurde fünf mal kein registrierter Fingerabdruck erkannt.
- Es sollen neue Fingerabdrücke für Touch ID eingerichtet oder registriert werden.

Ist Touch ID aktiviert, wird das Gerät sofort gesperrt, wenn die Standby-Taste gedrückt wird. Basiert die Sicherheit nur auf dem Code, stellen viele Benutzer eine Nachfrist für die Gerätesperre ein, damit sie den Code nicht bei jeder Benutzung des Geräts eingeben müssen. Mit Touch ID wird das Gerät im Ruhezustand stets gesperrt und zur Benutzung ist jedes Mal ein Fingerabdruck bzw. der Code erforderlich.

Touch ID kann lernen, bis zu fünf verschiedene Fingerabdrücke zu erkennen. Wenn nur ein Finger registriert wurde, liegt die Wahrscheinlichkeit, dass das Gerät mit einem anderen Finger entsperrt werden kann, bei 1 zu 50.000. Touch ID erlaubt aber nur bis zu fünf fehlgeschlagene Versuche, das Gerät mit einem Fingerabdruck zu entsperren, bevor der Benutzer den Code eingeben muss, um auf das Gerät zugreifen zu können.

Andere Anwendungsmöglichkeiten für Touch ID

Touch ID kann auch so konfiguriert werden, dass damit Einkäufe im iTunes Store, App Store oder iBooks Store bestätigt werden können, damit Benutzer das Passwort für ihre Apple-ID nicht eingeben müssen. Wenn Sie einen Einkauf autorisieren möchten, tauschen das Gerät und der Store ein Token für die Authentifizierung aus. Das Token und die kryptografische Nonce werden in der Secure Enclave gespeichert. Die Nonce wird mit einem Schlüssel der Secure Enclave signiert, den alle Geräte und der iTunes Store teilen.

Außerdem können Apps anderer Anbieter vom System angebotene APIs verwenden, um den Benutzer zur Authentifizierung per Touch ID oder Code aufzufordern. Die App wird nur benachrichtigt, ob die Authentifizierung erfolgreich war, sie kann nicht auf Touch ID oder die mit dem registrierten Fingerabdruck verbundenen Daten zugreifen.

Objekte im Schlüsselbund können ebenfalls mit Touch ID so geschützt werden, dass sie über die Secure Enclave nur mit einem passenden Fingerabdruck oder dem Code für das Gerät freigegeben werden können. App-Entwickler können ebenfalls über APIs überprüfen, ob vom Benutzer ein Code festgelegt wurde und so Objekte im Schlüsselbund mit Touch ID authentifiziert oder entsperrt werden können.

Sicherheit mit Touch ID

Der Fingerabdrucksensor ist nur dann aktiv, wenn der kapazitive Berührungssensor in dem Edelstahlring, der die Home-Taste umgibt, eine Fingerberührung erkennt, wodurch wiederum der Fingerabdruckscanner ausgelöst und das Ergebnis anschließend an die Secure Enclave gesendet wird.

Das 88 x 88 Pixel große Rasterscanbild mit 600 ppi wird vorübergehend im verschlüsselten Speicher innerhalb der Secure Enclave gespeichert und für die Analyse vektorisiert und anschließend wieder verworfen. Bei der Analyse wird der Verlauf der subkutanen Papillarleisten abgebildet. Dabei handelt es sich um ein verlustbehaftetes Verfahren, bei dem Details, die zur Rekonstruktion des Fingerabdruck des Benutzers benötigt würden, nicht gespeichert werden. Man erhält ein Abbild miteinander verbundener Knoten ohne personenbezogene Daten in verschlüsselter Form, das nur von der Secure Enclave gelesen werden kann und nie an Apple gesendet oder in iCloud oder iTunes gesichert wird.

Entsperren eines iOS-Geräts mit Touch ID

Wenn Touch ID deaktiviert ist und ein Gerät gesperrt wird, werden die Schlüssel für die Datensicherheitsklasse „Vollständiger Schutz“, die in der Secure Enclave gespeichert sind, verworfen. Die Dateien und die Objekte des Schlüsselbundes dieser Klasse sind nicht verfügbar, bis der Benutzer das Gerät über die Eingabe des Codes wieder entsperrt.

Bei aktivierter Touch ID werden die Schlüssel nicht verworfen, wenn das Gerät gesperrt wird, sondern stattdessen mit einem Schlüssel verpackt, der dem Touch ID-Teilsystem in der Secure Enclave übergeben wird. Versucht ein Benutzer, das Gerät zu entsperren, und erkennt Touch ID den Fingerabdruck des Benutzers, dann stellt es den Schlüssel zum Entpacken der Datensicherheitsschlüssel bereit und das Gerät wird entsperrt. Dieses Verfahren bietet zusätzlichen Schutz, da die Teilsysteme „Datensicherheit“ und „Touch ID“ zusammenarbeiten müssen, damit das Gerät entsperrt werden kann.

Die Schlüssel, die benötigt werden, damit Touch ID das Gerät entsperren kann, gehen verloren, wenn das Gerät neu startet. Sie werden von der Secure Enclave nach 48 Stunden oder fünf fehlgeschlagenen Versuchen, das Gerät mit Touch ID zu entsperren, verworfen.

Verschlüsselung und Datensicherheit

Der sichere Startvorgang, die Code-Signierung und Sicherheit für Laufzeitprozesse tragen alle dazu bei, dass nur vertrauenswürdige Codes und Apps auf dem Gerät ausgeführt werden können. iOS besitzt weitere Funktionen zur Verschlüsselung und Datensicherheit, die selbst dann die Benutzerdaten schützen, wenn andere Teile der Sicherheitsinfrastruktur kompromittiert wurden (zum Beispiel auf einem Gerät mit nicht autorisierten Veränderungen). Das hat wichtige Vorteile für Benutzer und IT-Administratoren, schützt zu jeder Zeit persönliche und Firmendaten und bietet die Möglichkeit, Geräte bei Diebstahl oder Verlust per Fernzugriff vollständig zu löschen.

Funktionen für die Hardwaresicherheit

Auf mobilen Geräten sind Geschwindigkeit und Energieeffizienz von entscheidender Bedeutung. Verschlüsselungsvorgänge sind komplex und können zu Problemen bei der Leistung oder Batterielebensdauer führen, wenn bei der Entwicklung und Implementation diese Prioritäten nicht berücksichtigt werden.

In jedem iOS-Gerät ist eine dedizierte AES 256 Crypto Engine in dem DMA-Pfad zwischen dem Flash-Speicher und dem Hauptspeicher vorhanden, was eine höchst effiziente Dateiverschlüsselung ermöglicht.

Die eindeutige ID des Geräts (UID) und eine Gerätegruppen-ID (GID) bestehen aus AES-256-Bit-Schlüsseln, die während der Herstellung in den Anwendungsprozessor und die Secure Enclave eingebrannt (UID) bzw. kompiliert (GID) werden. Keine Software oder Firmware kann diese direkt auslesen. Lediglich die Ergebnisse der Verschlüsselungs- oder Entschlüsselungsoperationen von den dedizierten AES-Engines können gelesen werden. Die AES-Engines wurden mit der UID oder GID als Schlüssel im Silizium implementiert. Außerdem können UID und GID der Secure Enclave nur von der dedizierten AES-Engine für die Secure Enclave verwendet werden. Die UID ist für jedes Gerät eindeutig und wird durch Apple oder eine Lieferanten nicht aufgezeichnet. Die GID wird bei allen Prozessoren in einer Geräteklasse gemeinsam verwendet (zum Beispiel bei allen Geräten mit dem Apple-A8-Prozessor) und sie wird für nicht sicherheitskritische Aufgaben verwendet, z. B. beim Bereitstellen von Systemsoftware während der Installation und Wiederherstellung. Die Integration dieser Schlüssel in Silizium trägt dazu bei sicherzustellen, dass Manipulationen und Umgehungsversuche erfolglos bleiben und sie nur für die AES-Engine zugänglich sind. Auf die UID und die GID kann auch nicht über JTAG oder andere Debugging-Schnittstellen zugegriffen werden.

Durch die UID können Daten kryptografisch an ein bestimmtes Gerät gebunden werden. So enthält beispielsweise die Schlüsselhierarchie, die das Dateisystem schützt, die UID. Werden die Speicherchips physisch von einem Gerät auf ein anderes bewegt, kann nicht auf die Dateien zugegriffen werden. Die UID hat keinerlei Verbindung zu anderen Kennungen auf dem Gerät.

Außer UID und GID werden alle anderen kryptografischen Schlüssel vom Zufallszahlengenerator (RNG) des Systems mit einem auf CTR_DRGB basierenden Algorithmus generiert. Die dafür erforderliche Entropie wird beim Starten aus Zeitabweichungen und zusätzlich nach abgeschlossenem Startvorgang aus dem Interrupt-Timing erzeugt. In Secure Enclave erzeugte Schlüssel verwenden den echten Hardwarezufallszahlengenerator, der auf mehreren Ringoszillatoren basiert, und mit CTR_DRBG nachbearbeitet wird.

Inhalte & Einstellungen löschen
Die Option „Inhalte & Einstellungen löschen" entfernt alle Schlüssel im Effaceable Storage, wodurch alle Benutzerdaten auf dem Gerät kryptografisch unzugänglich werden. Sie eignet sich daher ideal dafür sicherzustellen, dass sämtliche persönlichen Daten von einem Gerät gelöscht werden, bevor man es jemand anderem oder zur Reparatur gibt. Achtung: Verwenden Sie die Option „Inhalte & Einstellungen löschen" nur, wenn das Gerät gesichert wurde, da es ansonsten keine Möglichkeit gibt, die gelöschten Daten wiederherzustellen.

Das sichere Löschen gespeicherter Schlüssel ist genauso wichtig wie deren Erstellung. Dies ist bei Flash-Speicher eine besondere Herausforderung, da aufgrund der auf Abnutzungsverteilung ausgelegten Architektur möglicherweise mehrere Kopien der Daten gelöscht werden müssen. Um dieses Problem zu beheben, bieten iOS-Geräte eine Funktion zum sicheren Löschen von Daten namens „Effaceable Storage".

Mit dieser Funktion erfolgt ein Zugriff auf die zugrunde liegende Speichertechnologie (beispielsweise NAND), um direkt eine kleine Anzahl von Blöcken auf einer sehr niedrigen Ebene anzusteuern und zu löschen.

Sicherheit von Dateidaten

Zusätzlich zu den in iOS-Geräten eingebauten Funktionen zur Hardwareverschlüsselung verwendet Apple Funktionen für die Datensicherheit, um die im Flash-Speicher des Geräts abgelegten Daten noch effektiver zu schützen. Die Datensicherheit ermöglicht es einem Gerät, auf übliche Ereignisse, wie eingehende Telefonanrufe, zu reagieren und erlaubt zugleich einen hohen Verschlüsselungsstandard für die Benutzerdaten. Wichtige systemeigene Apps, wie Nachrichten, Mail, Kalender, Kontakte, Fotos oder Daten aus der App „Health" verwenden standardmäßig Datensicherheitsfunktionen. Apps anderer Anbieter, die unter iOS 7 oder neuer installiert wurden, erhalten diesen Schutz automatisch.

Die Datensicherheit wird durch die Erzeugung und Verwaltung einer Hierarchie von Schlüsseln implementiert. Sie baut auf den Technologien zur Hardwareverschlüsselung auf, die in jedes iOS-Gerät integriert sind. Die Datensicherheit wird mit einem pro Datei erzeugten Schlüssel gesteuert, wobei jede Datei einer Klasse zugeordnet wird; der Zugriff wird dadurch bestimmt, ob die Klassenschlüssel entsperrt wurden.

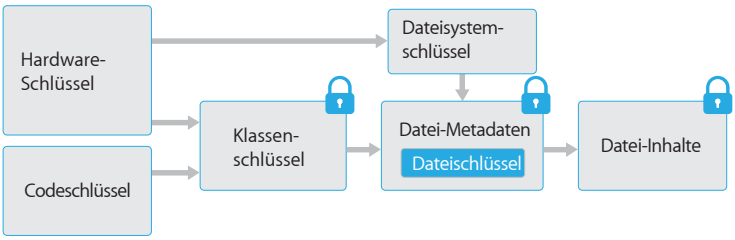
Architektur – Überblick

Jedes Mal, wenn eine Datei in der Datenpartition erstellt wird, erzeugt die Datensicherheit einen neuen 256-Bit-Schlüssel (den pro Datei erzeugten Schlüssel) und übergibt diesen an die Hardware-AES-Engine. Diese verwendet den Schlüssel zum Verschlüsseln der Datei, wenn diese mit dem AES-CBC-Modus in den Flash-Speicher geschrieben wird. Der Initialisierungsvektor (IV) wird anhand des Block-Offsets in der Datei berechnet und mit dem SHA-1 Hash des pro Datei erzeugten Schlüssels verschlüsselt.

Der pro Datei erzeugte Schlüssel wird mit einem von mehreren Klassenschlüsseln sicher verpackt. Der ausgewählte Schlüssel richtet sich nach den Umständen, unter denen die Datei zugänglich sein soll. Das sichere Verpacken wird immer mit NIST AES Key Wrapping, gemäß RFC 3394, durchgeführt. Dieser geschützte Schlüssel wird wiederum in den Metadaten der Datei gespeichert.

Wird eine Datei geöffnet, werden ihre Metadaten mit dem Dateisystemschlüssel entschlüsselt, wodurch der pro Datei erzeugte Schlüssel und ein Vermerk, mit welcher Klasse sie geschützt ist, entpackt werden. Der pro Datei erzeugte Schlüssel wird mit dem Klassenschlüssel entpackt und dann an die Hardware-AES-Engine gesendet, die die Datei beim Lesen aus dem Flash-Speicher entschlüsselt.

Die Metadaten für alle Dateien des Dateisystems werden mit einem Zufallsschlüssel verschlüsselt, der erzeugt wird, wenn iOS das erste Mal installiert oder das Gerät von einem Benutzer vollständig gelöscht wird. Der Schlüssel für das Dateisystem wird im Effaceable Storage gespeichert. Aufgrund der Speicherung auf dem Gerät wird dieser Schlüssel nicht verwendet, um die Vertraulichkeit von Daten aufrechtzuerhalten. Stattdessen ist er für das schnelle Löschen auf Anforderung konzipiert (durch den Benutzer mit der Option „Inhalte & Einstellungen löschen" oder durch einen Benutzer oder Administrator, der einen Befehl zum Fernlöschen von einem MDM-Server, Exchange ActiveSync oder iCloud absendet). Wird der Schlüssel auf diese Weise gelöscht, werden alle Dateien kryptografisch unzugänglich gemacht.



Der Inhalt einer Datei wird mit einem pro Datei erzeugten Schlüssel verschlüsselt, der mit einem Klassenschlüssel verpackt und in den Metadaten der Datei gespeichert wird, die wiederum mit dem Dateisystemschlüssel verschlüsselt sind. Der Klassenschlüssel wird mit der UID der Hardware und bei manchen Klassen mit dem Code des Benutzers geschützt. Diese Hierarchie bietet gleichzeitig Flexibilität und Effizienz. Wird beispielsweise die Klasse einer Datei geändert, muss nur der pro Datei erzeugten Schlüssel neu verpackt werden. Bei Änderung des Codes wird nur der Klassenschlüssel neu verpackt.

Gerätecodes

Erwägungen zum Gerätecode
Wenn ein langer Gerätecode, der nur aus Ziffern besteht, eingegeben wird, wird auf dem Sperrbildschirm anstelle der kompletten Tastatur nur eine numerische Tastatur angezeigt. Ein längerer Gerätecode, der nur aus Ziffern besteht, kann einfacher einzugeben sein als ein kürzerer alphanumerischer Gerätecode und bietet ähnlich hohe Sicherheit.

Durch das Einrichten eines Gerätecodes aktiviert der Benutzer automatisch die Datensicherheit. iOS unterstützt Gerätecodes, die aus vier Ziffern bestehen, und alphanumerische Gerätecodes beliebiger Länge. Zusätzlich zum Entsperren des Geräts stellt der Gerätecode die Entropie für bestimmte Verschlüsselungsschlüssel zur Verfügung. Das bedeutet, dass ein Angreifer, der ein Gerät in seinem Besitz hat, ohne den Gerätecode nicht auf Daten bestimmter Sicherheitsklassen zugreifen kann.

Der Gerätecode ist mit der UID des Geräts verknüpft, sodass Brute-Force-Angriffe direkt auf dem anvisierten Gerät durchgeführt werden müssen. Ein Zähler für die Anzahl der Wiederholungen sorgt dafür, dass mehr Zeit für jeden einzelnen Versuch benötigt wird. Dieser Zähler wurde so kalibriert, dass für einen Versuch etwa 80 Millisekunden benötigt werden. Das bedeutet, dass es über fünfeinhalb Jahre dauern würde, alle sechsstelligen alphanumerischen Gerätecodes, die aus Kleinbuchstaben und Ziffern bestehen, auszuprobieren.

Je sicherer ein Gerätecode ist, desto sicherer ist auch der Verschlüsselungscode. Touch ID kann dazu verwendet werden, diese Situation zu verbessern, da der Benutzer so einen sehr viel stärkeren Gerätecode einrichten kann, als normalerweise praktisch wäre. Dadurch wird effektiv die Entropie erhöht, mit der die für die Datensicherheit verwendeten Verschlüsselungsschlüssel geschützt werden, ohne dass die Benutzerfreundlichkeit leidet, wenn iOS-Geräte mehrmals täglich entsperrt werden müssen.

Um Brute-Force-Codeangriffe noch besser abzuwehren, setzt die Benutzeroberfläche von iOS steigende zeitliche Verzögerungen durch, wenn ein ungültiger Gerätecode auf dem Sperrbildschirm eingegeben wurde. Benutzer können festlegen, dass das Gerät nach zehn fehlgeschlagenen Code-Eingaben automatisch gelöscht werden soll. Diese Einstellung ist auch als Verwaltungsrichtlinie über Mobile Device Management (MDM) und Exchange ActiveSync verfügbar, und die maximal zulässige Anzahl von Fehleingaben kann zudem verringert werden.

Bei Geräten mit einem A7 oder neueren Prozessor werden die entscheidenden Schritte von der Secure Enclave ausgeführt, die zudem eine Verzögerung von 5 Sekunden nach einem fehlgeschlagenen Entsperrversuch erzwingt. Damit fungiert sie als Geschwindigkeitsbegrenzer, der zusätzlich zu den Sicherungsmaßnahmen von iOS vor Brute-Force-Angriffen schützt.

Datensicherheitsklassen

Wird eine neue Datei auf einem iOS-Gerät erzeugt, so wird dieser Datei von der App, die sie erzeugt, eine Klasse zugewiesen. Jede Klasse verwendet unterschiedliche Richtlinien, um zu bestimmen, wann auf die Daten zugegriffen werden kann. Die grundlegenden Klassen und Richtlinien werden im Folgenden beschrieben.

Vollständiger Schutz

(`NSFileProtectionComplete`): Der Klassenschlüssel wird mit einem Schlüssel geschützt, der aus dem Code des Benutzers und der UID des Geräts abgeleitet wird. Kurz nachdem der Benutzer das Gerät gesperrt hat (10 Sekunden, wenn „Sofort" für „Code anfordern" eingestellt wurde), wird der entschlüsselte Klassenschlüssel verworfen, sodass sämtliche Daten dieser Klasse unzugänglich sind, bis der Benutzer den Code erneut eingibt bzw. das Gerät mit Touch ID entsperrt.

Geschützt, außer wenn offen

(`NSFileProtectionCompleteUnlessOpen`): Manche Dateien müssen geschrieben werden, während das Gerät gesperrt ist. Ein gutes Beispiel sind E-Mail-Anhänge, die im Hintergrund geladen werden. Dieses Verhalten wird durch die Verwendung der asymmetrischen Elliptic Curve Cryptography (ECDH over Curve25519) erreicht. Der normale pro Datei erzeugte Schlüssel wird mit einem Schlüssel geschützt, der per One-Pass-Diffie-Hellmann-Schlüsselaustausch wie in NIST SP 800-56A beschrieben erzeugt wird.

Der temporäre öffentliche Schlüssel für den Austausch wird zusammen mit dem verpackten pro Datei erzeugten Schlüssel gespeichert. Es wird die KDF (Concatenation Key Derivation Function, anerkannte Alternative 1) wie unter 5.8.1 in der NIST SP 800-56A beschrieben verwendet. Die Algorithmus-ID wird weggelassen. Als temporärer bzw. statischer öffentlicher Schlüssel werden PartyUInfo und PartyVInfo verwendet. Als Hash-Funktion wird SHA-256 verwendet. Sobald die Datei geschlossen wird, wird der pro Datei erzeugte Schlüssel aus dem Speicher gelöscht. Um die Datei erneut öffnen zu können, wird das Shared Secret mit dem privaten Schlüssel der Klasse „Geschützt, außer wenn offen" und dem temporären öffentlichen Schlüssel der Datei neu erstellt; der Hash wird verwendet, um den pro Datei erzeugten Schlüssel zu entpacken, mit dem wiederum die Datei entschlüsselt wird.

Geschützt bis zur ersten Benutzerauthentifizierung

(`NSFileProtectionCompleteUntilFirstUserAuthentication`): Diese Klasse verhält sich wie der vollständige Schutz, mit dem Unterschied, dass der entschlüsselte Klassenschlüssel beim Sperren des Geräts nicht aus dem Speicher gelöscht wird. Der Schutz dieser Klasse ist mit der vollständigen Festplattenverschlüsselung auf Desktopcomputern vergleichbar. Daten werden so vor Angriffen geschützt, die einen Neustart beinhalten. Dies ist die Standardklasse für Apps von Drittanbietern, die keiner anderen Datensicherheitsklasse zugewiesen wurden.

Kein Schutz

(`NSFileProtectionNone`): Dieser Klassenschlüssel wird nur mit der UID geschützt und im Effaceable Storage gespeichert. Da alle Schlüssel zum Entschlüsseln von Dateien dieser Klasse auf dem Gerät gespeichert werden, bietet diese Verschlüsselung nur den Vorteil einer schnellen Fernlöschung. Wenn einer Datei keine Datensicherheitsklasse zugewiesen wird, wird sie dennoch in verschlüsselter Form gespeichert (wie alle Daten auf einem iOS-Gerät).

Komponenten eines Objekts im Schlüsselbund

Neben der Zugangsgruppe enthält jedes Schlüsselbundelement administrative Metadaten (z. B. Zeitstempel wie „Erstellt" und „Zuletzt Aktualisiert").

Außerdem enthalten sie SHA-1 Hashwerte der bei der Abfrage eines Objekts verwendeten Attribute (z. B. Name des Accounts oder Servers), damit eine Suche auch ohne Entschlüsselung der einzelnen Objekte möglich ist. Schließlich enthalten sie Verschlüsselungsdaten, darunter:

- Versionsnummer
- Daten der Zugriffssteuerungslisten (ACL)
- Einen Wert, der die Sicherheitsklasse des Objekts angibt
- Den pro Datei erzeugten Schlüssel, der mit dem Sicherheitsklassenschlüssel verschlüsselt wurde
- Das Verzeichnis der Attribute, die das Objekt beschreiben (wie an „SecItemAdd" weitergegeben), als binäre plist codiert und mit dem pro Datei erzeugten Schlüssel verschlüsselt werden

Als Verschlüsselung wird AES 128 im Galois/Counter Mode (GCN) verwendet; die Zugriffsgruppe ist in den Attributen enthalten und wird mit dem GMAC-Tag geschützt, der bei der Verschlüsselung errechnet wird.

Sicherheit von Schlüsselbunddaten

Viele Apps müssen Passwörter und andere kurze, aber vertrauliche Datensätze, wie Schlüssel und Anmelde-Tokens, verarbeiten. Der iOS-Schlüsselbund stellt eine sichere Methode zum Speichern dieser Elemente zur Verfügung.

Der Schlüsselbund ist als SQLite-Datenbank implementiert, die im Dateisystem gespeichert wird. Die Datenbank existiert nur einmal im System. Der securityd-Daemon legt fest, auf welche Schlüsselbundelemente ein Prozess oder eine App zugreifen kann. Zugriffe auf die APIs des Schlüsselbunds resultieren in Anfragen an den Daemon, der wiederum die „keychain-access-groups"- und die „application-identifier"-Berechtigung abfragt. Anstatt den Zugriff auf einen einzelnen Prozess einzuschränken, ermöglichen die Zugriffsgruppen es, Schlüsselbundeinträge zwischen Apps zu teilen.

Schlüsselbundeinträge können nur zwischen Apps desselben Entwicklers gemeinsam genutzt werden. Die Verwaltung dieser Vorgabe erfolgt dadurch, dass Apps anderer Anbieter Zugriffsgruppen mit einem Präfix verwenden, das ihnen durch das iOS-Entwicklerprogramm oder in iOS 8 über Anwendungsgruppen zugewiesen wurde. Das Präfix-Erfordernis und die Einzigartigkeit der Anwendungsgruppe werden über Codesignierung, Bereitstellungsprofile und das iOS-Entwicklerprogramm sichergestellt.

Die Schlüsselbunddaten werden mit einer Klassenstruktur geschützt, die der Klassenstruktur bei der Dateidatensicherheit ähnelt. Diese Klassen weisen ähnliche Verhaltensweisen wie die Datei-Datensicherheitsklassen auf, verwenden aber separate Schlüssel und sind Bestandteil von APIs, die unterschiedlich benannt sind.

Verfügbarkeit	Schutz von Dateien	Schutz von Schlüsselbunddaten
Wenn entsperrt	NSFileProtectionComplete	kSecAttrAccessibleWhenUnlocked
Wenn gesperrt	NSFileProtectionCompleteUnlessOpen	Nicht verfügbar
Nach erstem Entsperren	NSFileProtectionCompleteUntilFirstUserAuthentication	kSecAttrAccessibleAfterFirstUnlock
Immer	NSFileProtectionNone	kSecAttrAccessibleAlways
Code aktiviert	n. a.	kSecAttrAccessible-WhenPasscodeSetThisDeviceOnly

Apps, die Hintergrundaktualisierungsdienste nutzen, können `kSecAttrAccessibleAfterFirstUnlock` für Schlüsselbundobjekte nutzen, auf die bei Hintergrundaktualisierungen zugegriffen werden muss.

Die Klasse `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly` ist nur verfügbar, wenn das Gerät mit einem Gerätecode konfiguriert wurde. Die Objekte in dieser Klasse existieren nur im System-Keybag, sie werden nicht mit dem iCloud-Schlüsselbund synchronisiert, werden nicht gesichert und sind nicht in Escrow-Keybags enthalten. Wenn der Gerätecode entfernt oder zurückgesetzt wird, werden diese Objekte unbrauchbar, da die Klassenschlüssel verworfen werden.

Andere Schlüsselbundklassen besitzen ein „Nur dieses Gerät"-Gegenstück, das immer mit der UID geschützt wird, wenn eine Kopie bei der Sicherung des Gerät erstellt wird. Dadurch wird es bei der Wiederherstellung auf einem anderen Gerät nutzlos.

Apple hat Sicherheit und Benutzerfreundlichkeit sorgfältig abgewogen und Schlüsselbundklassen gewählt, die von der Art der zu sichernden Information abhängen und davon, wann iOS auf sie zugreifen muss. Ein VPN-Zertifikat muss zum Beispiel immer verfügbar sein, damit das Gerät eine permanente Verbindung besitzt, wird aber als „nicht-migrierend" klassifiziert, kann also nicht auf ein anderes Gerät übertragen werden.

Bei von iOS erstellten Schlüsselbundobjekten wird der folgende Klassenschutz erzwungen:

Objekt	Zugänglich
WLAN-Passwörter	Nach erstem Entsperren
Mail-Accounts	Nach erstem Entsperren
Exchange-Accounts	Nach erstem Entsperren
VPN-Passwörter	Nach erstem Entsperren
LDAP, CalDAV, CardDAV	Nach erstem Entsperren
Account-Token für soziale Netzwerke	Nach erstem Entsperren
Schlüssel für Handoff-Ankündigungen	Nach erstem Entsperren
iCloud Token	Nach erstem Entsperren
Homeshare-Passwörter	Wenn entsperrt
Token für „Mein iPhone suchen“	Immer
Voicemail	Immer
iTunes-Backup	Wenn entsperrt, nicht-migrierend
Safari-Passwörter	Wenn entsperrt
VPN-Zertifikate	Immer, nicht-migrierend
Bluetooth®-Schlüssel	Immer, nicht-migrierend
Token für Apple-Push-Benachrichtigungsdienst	Immer, nicht-migrierend
iCloud-Zertifikate und privater Schlüssel	Immer, nicht-migrierend
iMessage-Schlüssel	Immer, nicht-migrierend
Vom Konfigurationsprofil installierte Zertifikate und private Schlüssel	Immer, nicht-migrierend
SIM-PIN	Immer, nicht-migrierend

Schlüsselbundzugriffssteuerung

Schlüsselbunde können Zugriffssteuerungslisten (ACLs) verwenden, um Richtlinien für Zugriffs- und Authentifizierungsanforderungen festzulegen. Objekte können Bedingungen festlegen, bei denen der Benutzer sich per Touch ID oder Eingabe des Gerätecodes authentifizieren muss, um auf sie zugreifen zu können. ACL werden in der Secure Enclave evaluiert und nur dann an den Kernel weitergegeben, wenn die angegebenen Einschränkungen erfüllt sind.

Zugriff auf gesicherte Passwörter in Safari

Apps können in iOS mit den Schlüsselbundobjekten, die Safari zum automatischen Ausfüllen von Passwörtern speichert, über die folgenden beiden APIs zugreifen:

- `SecRequestSharedWebCredential`
- `SecAddSharedWebCredential`

Zugriff wird nur gewährt, wenn sowohl der Entwickler der App als auch der Administrator der Website dies erlauben und der Benutzer zugestimmt hat. App-Entwickler können angeben, dass sie auf die von Safari gesicherten Passwörter zugreifen möchten, indem sie in ihre App eine Berechtigung einfügen. Diese Berechtigung beinhaltet den vollständig qualifizierten Domain-Namen verknüpfter Websites. Auf dem Server der Website muss sich eine signierte CMS-Datei befinden, die die eindeutige Kennung der Apps, die zugelassen sind, enthält. Wird eine App mit der Berechtigung „com.apple.developer.associated-domains“ installiert, stellt iOS 8 eine TLS-Anfrage für die Datei „/apple-app-site-association“ an alle aufgeführten Websites. Wenn die Signatur von einer für die Domain gültigen und von iOS als vertrauenswürdig eingestuften Identität stammt und die Datei die Kennung der zu installierenden App auflistet, markiert iOS die Beziehung zwischen Website und App als vertrauenswürdig. Nur bei vertrauenswürdigen Beziehungen führen Aufrufe dieser beiden APIs zu einer Eingabeaufforderung für den Benutzer, der zustimmen muss, bevor Passwörter für die App freigegeben, aktualisiert oder gelöscht werden.

Keybags

Die Schlüssel sowohl für Datei- als auch für Schlüsselbund-Datensicherheitsklassen werden in Keybags gesammelt und verwaltet. iOS verwendet die folgenden vier Keybags: System, Backup, Escrow und iCloud-Backup.

Im **System-Keybag** werden die verpackten Klassenschlüssel, die im normalen Betrieb des Geräts verwendet werden, gespeichert. Wenn beispielsweise ein Gerätecode eingegeben wird, wird der Schlüssel `NSFileProtectionComplete` aus dem System-Keybag geladen und entpackt. Es handelt sich um eine binäre plist, die in der Klasse „Kein Schutz" gespeichert ist, deren Inhalte aber mit einem Schlüssel aus dem Effaceable Storage verschlüsselt werden. Dieser Schlüssel wird jedes Mal, wenn der Besitzer seinen Code ändert, gelöscht und neu erzeugt, um den Keybags Folgenlosigkeit (Forward Secrecy) zu verleihen. Die Kernel Extension `AppleKeyStore` verwaltet den System-Keybag und über sie kann der Status für die Sperre des Geräts abgerufen werden. Sie meldet nur dann, dass das Gerät entsperrt ist, wenn auf alle Klassenschlüssel im System-Keybag zugegriffen werden kann und sie erfolgreich entpackt wurden.

Der **Backup-Keybag** wird erstellt, wenn iTunes ein verschlüsseltes Backup erstellt und auf dem Computer speichert, auf dem das Gerät gesichert wird. Es wird ein neuer Keybag mit neuen Schlüsseln erstellt und die gesicherten Daten werden mit diesen neuen Schlüsseln erneut verschlüsselt. Wie oben beschrieben bleiben nicht-migrierende Schlüsselbundobjekte mit dem von der UID abgeleiteten Schlüssel verpackt, sodass sie auf dem Gerät, von dem sie ursprünglich gesichert wurden, wiederhergestellt werden können, von anderen Geräten aus aber nicht auf sie zugegriffen werden kann.

Der Keybag wird mit dem in iTunes festgelegten Passwort geschützt, das 10.000 Iterationen der PBKDF2 durchläuft. Trotz dieses Zählers wird keine Verknüpfung mit einem bestimmten Gerät hergestellt, sodass theoretisch ein Brute-Force-Angriff von mehreren Computern gleichzeitig aus auf den Backup-Keybag ausgeführt werden könnte. Dieser Bedrohung kann mit einem hinreichend sicheren Passwort entgegengewirkt werden.

Wenn ein Benutzer ein iTunes-Backup nicht verschlüsseln lässt, werden die Backupdateien unabhängig von ihrer Datensicherheitsklasse nicht verschlüsselt, der Schlüsselbund wird aber weiterhin mit einem von der UID abgeleiteten Schlüssel geschützt. Aus diesem Grund können Schlüsselbundobjekte nur auf ein neues Gerät migriert werden, wenn ein Backup-Passwort festgelegt wurde.

Der **Escrow-Keybag** wird zum Synchronisieren von iTunes und für MDM verwendet. Mit diesem Keybag kann iTunes sichern und synchronisieren, ohne dass der Benutzer einen Gerätecode eingeben muss, außerdem ermöglicht er es einem MDM-Server, den Code eines Benutzers fernzulöschen. Er wird auf dem Computer gespeichert, der zum Synchronisieren von iTunes verwendet wird, oder auf dem MDM-Server, der das Gerät verwaltet.

Der Escrow-Keybag verbessert die Benutzerfreundlichkeit beim Synchronisieren von Geräten, wobei potenziell auf Daten aller Klassen zugegriffen werden muss. Wenn ein mit einem Code gesperrtes Gerät das erste Mal mit iTunes verbunden wird, muss der Benutzer einen Code eingeben. Das Gerät erstellt daraufhin einen Escrow-Keybag, der mit einem neu erzeugten Schlüssel geschützt wird und dieselben Klassenschlüssel enthält, die auf dem Gerät verwendet werden. Der Escrow-Keybag und der Schlüssel, mit dem er geschützt wird, werden zwischen dem Gerät und dem Host/Server aufgeteilt, wobei den auf dem Gerät gespeicherten Daten die Klasse „Geschützt bis zur ersten Benutzerauthentifizierung" zugewiesen wird. Aus diesem Grund muss der Code für das Gerät das erste Mal nach einem Neustart eingegeben werden, bevor der Benutzer ein iTunes-Backup erstellen kann.

Eine bestimmte Instanz eines Escrow-Keybags (ein so genannter *Stash-Keybag*) wird bei einer Softwareaktualisierung verwendet, damit der Aktualisierungsprozess auf Dateien und Schlüsselbundobjekte aller Datensicherheitsklassen zugreifen kann. Während der Aktualisierung wird nach dem Neustart Zugriff benötigt, um Datenmigrationen ausführen zu können, die Aufgaben wie die Aktualisierung der Datenbankschemata durchführen, die Vorschau für neue Objekte erzeugen oder auch die Datensicherheitsklassen erhöhen.

Bei OTA-Softwareaktualisierungen wird der Benutzer vor dem Starten der Aktualisierung nach seinem Code gefragt. Dadurch erhält man eine sichere Markierung im System-Keybag, durch die ein Stash-Keybag im Speicher erstellt wird, während das Gerät gesperrt ist. Wenn der Benutzer zum Durchführen der Aktualisierung bereit ist, was nur nach dem Entsperren des Geräts möglich ist, wird der Stash-Keybag auf das Laufwerk geschrieben und im Effaceable Storage mit einem Schlüssel geschützt. Bei der Aktualisierung mit iTunes und einem verbundenem Host mit einem gültigen Escrow-Keybag müssen Benutzer ihr Gerät entsperren, bevor die Aktualisierung beginnt, damit der Stash-Keybag bei entsperrtem Gerät auf das Laufwerk geschrieben werden kann.

Während die Datenmigration ausgeführt wird, wird der Stash-Keybag geladen, sodass der Zugriff auf die Schlüssel der Datensicherheitsklassen möglich wird. Der gespeicherte Stash-Keybag wird dann gelöscht und der Schlüssel, mit dem er geschützt wird, aus dem Effaceable Storage entfernt, damit er nicht erneut verwendet werden kann. Wenn der Datenmigrationsprozess beendet wird, werden die Schlüssel, die normalerweise nur bei entsperrtem Gerät existieren, verworfen und das Gerät wird in den Zustand „Nach dem ersten Entsperren“ versetzt.

Wenn vor der Aktualisierung kein Stash-Keybag erstellt werden konnte, zeigt das Gerät nach dem Neustart die Eingabeaufforderung „Zum Aktualisieren streichen“ und fragt nach dem Code, um die Aktualisierung abzuschließen.

Der **iCloud-Backup-Keybag** ist dem Backup-Keybag ähnlich. Alle Klassenschlüssel in diesem Keybag sind asymmetrisch (es wird Curve25519 verwendet, wie bei der Datensicherheitsklasse „Geschützt außer wenn offen“), sodass iCloud-Backups im Hintergrund durchgeführt werden können. Bei allen Datensicherheitsklassen außer „Kein Schutz“ werden die verschlüsselten Daten des Geräts gelesen und an iCloud gesendet. Die entsprechenden Klassenschlüssel werden mit den iCloud-Schlüsseln geschützt. Die Klassenschlüssel des Schlüsselbunds werden mit einem Schlüssel verpackt, der von der UID abgeleitet wird, wie bei einem nicht verschlüsselten iTunes-Backup. Ein asymmetrischer Keybag wird ebenfalls für das Backup in der Schlüsselbundwiederherstellung des iCloud-Schlüsselbundes verwendet.

FIPS 140-2

Die Verschlüsselungsmodule in iOS 8 werden auf Erfüllung des U.S. Federal Information Processing Standard (FIPS) 140-2 Level 1 geprüft. Dadurch wird die Integrität bei Verschlüsselungsvorgängen in Apple-Apps und Apps anderer Anbieter validiert, die die Verschlüsselungsdienste von iOS ordnungsgemäß nutzen. Weitere Informationen zu früheren Validierungen und dem Status von iOS 8 finden Sie unter https://support.apple.com/kb/HT5808?viewlocale=de_DE.

Sicherheit in Apps

Apps sind die kritischen Elemente einer moderne Sicherheitsarchitektur für Mobilgeräte. Apps bieten dem Benutzer fantastische Produktivitätsgewinne, haben aber auch das Potenzial, die Systemsicherheit, die Stabilität und die Benutzerdaten zu gefährden, wenn mit ihnen nicht richtig umgegangen wird.

Aus diesem Grund bietet iOS mehrere Sicherheitsebenen, mit denen sichergestellt wird, dass Apps signiert und überprüft wurden und zum Schutz der Benutzerdaten Sandboxing verwenden. Diese Elemente bieten eine stabile, sichere Plattform für Apps und ermöglichen es Tausenden von Entwicklern, hunderttausende Apps für iOS zu entwickeln, ohne dass die Systemintegrität beeinträchtigt wird. Benutzer können mit ihrem iOS-Gerät auf diese Apps zugreifen, ohne unnötig Angst vor Viren, Malware oder nicht autorisierten Attacken haben zu müssen.

App-Codesignierung

Nachdem der iOS Kernel gestartet wurde, bestimmt er, welche Benutzerprozesse und Apps ausgeführt werden dürfen. Um sicherzustellen, dass alle Apps von bekannten und genehmigten Quellen stammen und nicht manipuliert wurden, muss der gesamte ausführbare Code für iOS mit einem von Apple ausgegebenem Zertifikat signiert worden sein. Die ab Werk auf dem Gerät vorhandenen Apps (z. B. Mail und Safari) wurden von Apple signiert. Apps von anderen Anbietern müssen ebenfalls vom Entwickler mit einem von Apple ausgegebenen Zertifikat validiert und signiert werden. Die zwingende Codesignierung weitet das „Chain of Trust"-Konzept vom Betriebssystem auf Apps aus und verhindert, dass Apps anderer Anbieter nicht signierten Code ausführen oder Code verwenden, der sich selbst ändert.

Um Apps auf iOS-Geräten entwickeln und installieren zu können, müssen sich Entwickler bei Apple registrieren und dem iOS-Entwicklerprogramm beitreten. Vor der Ausgabe des Zertifikats überprüft Apple die Identität jedes Entwicklers (Einzelpersonen oder Unternehmen). Mit diesem Zertifikat können Entwickler Apps signieren und sie zur Verteilung an den App Store senden. Alle Apps im App Store wurden also von identifizierbaren Personen oder Organisationen eingereicht, was für Entwickler schädlicher Apps als Abschreckung dient. Außerdem werden sie von Apple auf eine korrekte Funktionsweise überprüft, um sicherzustellen, dass sie keine offensichtlichen Bugs oder andere Probleme enthalten. Zusätzlich zu den bereits beschriebenen Technologien können Benutzer dank diesem Kurationsverfahren auf die Qualität der gekauften Apps vertrauen.

Mit iOS 8 können Entwickler in ihre Apps Frameworks integrieren, die von der App selbst oder von in der App integrierten Erweiterungen genutzt werden können. Um das System und andere Apps davor zu schützen, dass Code aus anderen Apps in ihrem Adressbereich ausgeführt wird, führt das System eine Validierung der Codesignatur für alle dynamischen Bibliotheken durch, auf die ein Prozess beim Start zugreift. Diese Überprüfung erfolgt über die Team-ID, die aus einem von Apple ausgegebenem Zertifikat extrahiert wird. Bei einer Team-ID handelt es sich um eine zehnstellige alphanumerische Zeichenfolge, z. B. 1A2B3C4D5E. Ein Programm kann auf jede Plattformbibliothek, die auf dem System vorinstalliert ist, und auf jede Bibliothek mit derselben Team-ID in der Codesignatur wie der eigentliche ausführbare Code zugreifen. Da der auf dem System vorinstallierte Code keine Team-ID besitzt, kann er nur auf Bibliotheken zugreifen, die ebenfalls auf dem System vorinstalliert sind.

Unternehmen haben auch die Möglichkeit, firmeninterne Apps zur Verwendung im Unternehmen zu entwickeln und sie an die Mitarbeiter zu verteilen. Unternehmen und Entwickler können sich mit einer D-U-N-S-Nummer für das iOS Developer Enterprise Program (iDEP) bewerben. Apple genehmigt Bewerbungsanträge nach Prüfung der Identität und Eignung der Bewerber. Tritt eine Organisation dem iDEP bei, kann sie sich für die Ausstellung eines Bereitstellungsprofils registrieren, mit dem firmeninterne Apps auf autorisierten Geräten ausgeführt werden können. Benutzer müssen das Bereitstellungsprofil installieren, um firmeninterne Apps ausführen zu können. Dadurch wird sichergestellt, dass nur die Benutzer in einer Organisation die Apps auf ihre iOS-Geräte laden können, die dazu berechtigt sein sollen. Firmeninterne Apps überprüfen auch während der Laufzeit, ob die Signatur gültig ist. Apps mit einem abgelaufenen oder annullierten Zertifikat werden nicht ausgeführt.

Anders als andere mobile Plattformen erlaubt iOS es seinen Benutzern nicht, potenziell schädliche, unsignierte Apps von Websites zu installieren oder nicht vertrauenswürdigen Code auszuführen. Während der Laufzeit wird die Codesignatur aller ausführbaren Speicherseiten überprüft, wenn sie geladen werden, um sicherzustellen, dass die App seit der Installation oder letzten Aktualisierung nicht modifiziert wurde.

Sicherheit von Laufzeitprozessen

Wenn überprüft wurde, ob die App aus einer vertrauenswürdigen Quelle stammt, setzt iOS Sicherheitsmaßnahmen durch, die verhindern sollen, dass andere Apps oder der Rest des Systems gefährdet werden.

Apps anderer Anbieter werden in einer Sandbox ausgeführt, damit sie keine Änderungen am Gerät vornehmen oder auf Dateien zugreifen können, die von anderen Apps gespeichert wurden. Dadurch können Apps keine von anderen Apps gespeicherten Informationen abrufen oder verändern. Jede App verfügt über ein eigenes Heimatverzeichnis für die dazugehörigen Dateien, der bei der Installation der App zufällig ausgewählt wird. Wenn eine App eines anderen Anbieters auf Informationen zugreifen muss, die ihr nicht zugeordnet sind, kann sie das nur über Dienste tun, die explizit von iOS bereitgestellt werden.

Systemdateien und Ressourcen werden ebenfalls von den Apps des Benutzers abgeschirmt. Der Großteil von iOS und alle Apps anderer Anbieter werden über den nicht privilegierten Benutzer „mobile“ ausgeführt. Die gesamte Betriebssystempartition ist nur für den Lesezugriff aktiviert. Nicht notwendige Tools, wie etwa Dienste für die entfernte Anmeldung gehören nicht zur Systemsoftware und die APIs lassen es nicht zu, dass Apps ihre eigenen Privilegien erhöhen, um andere Apps oder iOS zu verändern.

Der Zugriff auf Benutzerinformationen und Funktionen wie iCloud und die Erweiterbarkeit durch Apps anderer Anbieter wird über festgelegte Berechtigungen gesteuert. Berechtigungen sind Schlüssel/Wert-Paare, die zusammen mit einer App signiert werden und die eine Authentifizierung über Laufzeitfaktoren wie die UNIX-Benutzerkennung hinaus ermöglichen. Da die Berechtigungen digital signiert sind, können sie nicht verändert werden. Berechtigungen werden in großem Umfang von System-Apps und Daemons zur Durchführung bestimmter privilegierter Vorgänge verwendet, die andernfalls als Root ausgeführt werden müssten. Dadurch wird die Gefahr einer Privilegienerhöhung durch eine beschädigte Systemanwendung oder einen Daemon reduziert.

Außerdem können Apps nur über vom System bereitgestellt APIs die Hintergrundverarbeitung verwenden. Dadurch können Apps ohne Leistungseinbußen oder dramatische Beeinträchtigung der Batterielebensdauer weiterarbeiten.

Die Speicherverwürfelung (Address Space Layout Randomization, ASLR) schützt davor, dass Fehler, die den Speicher modifizieren, ausgenutzt werden können. Integrierte Apps nutzen ASLR, um sicherzustellen, dass beim Start alle Speicherbereiche zufällig vergeben werden. Die zufällige Anordnung der Speicheradressen von ausführbarem Code, den Systembibliotheken (System Libraries) und den zugehörigen Programmelementen verringert die Wahrscheinlichkeit vieler komplexer Exploits. Ein „return-to-libc"-Angriff versucht beispielsweise, durch die Manipulation der Speicheradressen von Stack- und System Libraries ein Gerät zum Ausführen von Schadcode zu zwingen. Werden diese Bibliotheken zufällig platziert, erschwert dies den Angriff deutlich, insbesondere wenn sich dieser gegen mehrere Geräte richtet. Xcode, die Entwicklerumgebung für iOS, kompiliert Programme anderer Anbieter automatisch mit aktivierter ASLR-Unterstützung.

Zusätzlichen Schutz bietet die ARM-Funktion „Execute Never" (XN), die Speicherseiten als nicht-ausführbar kennzeichnet. Speicherseiten, die als beschreibbar und ausführbar gekennzeichnet sind, können von Apps nur unter streng kontrollierten Bedingungen verwendet werden: Der Kernel überprüft, ob die Apple vorbehaltene Berechtigung „dynamic code signing" vorhanden ist. Selbst dann kann nur ein einziger mmap-Aufruf genutzt werden, um eine ausführbare und beschreibbare Seite anzufordern, die eine zufällige Adresse erhält. Safari verwendet diese Funktion für seinen JavaScript JIT Compiler.

Erweiterungen

In iOS können Apps mithilfe von *Erweiterungen* Funktionen für andere Apps bereitstellen. Erweiterungen sind signierte, ausführbare Binärdateien für einen speziellen Zweck, die in eine App verpackt wurden. Das System erkennt Erweiterungen automatisch bei der Installation und stellt sie anderen Apps über ein Abgleichsystem zur Verfügung.

Ein Systembereich, der Erweiterungen unterstützt, wird *Erweiterungspunkt* (Extension Point) genannt. Jeder Erweiterungspunkt stellt APIs bereit und setzt die Richtlinien für den Bereich durch. Das System legt anhand der jeweiligen Abgleichregeln des Erweiterungspunkts fest, welche Erweiterungen zur Verfügung stehen. Das System startet Erweiterungsprozesse bei Bedarf und verwaltet ihren Lebenszyklus automatisch. Berechtigungen können dafür verwendet werden, die Verfügbarkeit von Erweiterungen für bestimmte Systemanwendungen einzuschränken. Das Widget für die Tagesansicht erscheint beispielsweise nur in der Mitteilungszentrale und eine Freigabeerweiterung ist nur im Bereich „Freigabe" verfügbar. Die Erweiterungspunkte sind das Widget „Tagesansicht", die Freigabe, Eigene Aktionen, Bildbearbeitung, Document Provider und eigene Tastatur.

Erweiterungen werden in ihrem eigenen Adressbereich ausgeführt. Die Kommunikation zwischen Erweiterung und der App, die sie aktiviert hat, verwendet Interprozesskommunikation, die vom System-Framework vermittelt wird. Sie können nicht auf die Dateien oder Speicherbereiche der anderen Seite zugreifen. Erweiterungen sind voneinander, von der App, die sie enthält, und von anderen Apps, die sie verwenden, abgeschirmt. Sie werden genau wie alle anderen Apps anderer Anbieter in einer Sandbox ausgeführt und ihr Container ist nicht derselbe Container wie der der App. Sie teilen sich jedoch den Zugriff auf die Datenschutzeinstellungen mit der App, die sie enthält. Wenn also ein Benutzer einer App Zugriff auf die Kontakte erlaubt, so wird dieser Zugriff auch an die in der App integrierten Erweiterungen weitergereicht, aber nicht an Erweiterungen, die in der App aktiviert wurden.

Eigene Tastaturen stellen eine Sonderform der Erweiterungen dar, da sie vom Benutzer für das gesamte System aktiviert werden. Nach der Aktivierung wird die Erweiterung für alle Textfelder verwendet, außer für die Eingabe des Codes und verschlüsselte Textfelder. Aus Gründen des Datenschutzes werden eigene Tastaturen standardmäßig in einer besonders eingeschränkten Sandbox ausgeführt, welche den Zugriff auf das Netzwerk, auf Dienste, welche Netzwerkoperationen stellvertretend für den Prozess

ausführen und APIs, welche es der Erweiterung erlauben würden Eingabedaten aufzuzeichnen, blockiert. Entwickler eigener Tastaturen können anfordern, dass ihre Erweiterung Open Access erhält, mit dem das System nach Bestätigung durch den Benutzer die Erweiterung in der Standardsandbox ausführt.

Bei Geräten, die für Mobile Device Management registriert sind, folgen Dokument- und Tastaturerweiterungen der Regel „Verwaltetes Öffnen" (Managed Open In). Der MDM-Server kann beispielsweise verhindern, dass ein Benutzer ein Dokument aus einer verwalteten App in einen nicht verwalteten Document Provider exportiert oder eine nicht verwaltete Tastatur in einer verwalteten App verwendet. Außerdem können Entwickler festlegen, dass in ihrer App keine Tastaturerweiterungen anderer Anbieter verwendet werden dürfen.

App-Gruppen

Apps und Erweiterungen, die zum selben Entwickleraccount gehören, können Inhalte teilen, wenn sie als Teil einer App-Gruppe konfiguriert wurden. Es liegt am Entwickler, auf dem Apple-Entwicklerportal geeignete Gruppen zu erstellen, die die gewünschten Apps und Erweiterungen enthalten. Apps, die einer App-Gruppe zugewiesen wurden, haben Zugriff auf:

- Einen geteilten On-Disk-Container als Speicher, der auf dem Gerät bleibt, solange mindestens eine App aus der Gruppe installiert ist
- Freigegebene Einstellungen
- Freigegebene Schlüsselbundobjekte

Das Apple-Entwicklerportal garantiert, dass App-Gruppen-IDs im gesamten Ökosystem eindeutig sind.

Sicherheit von Daten in Apps

Das iOS Software Development Kit (SDK) bietet ein API-Komplettpaket, mit dem externe und interne Entwickler Datensicherheit ganz einfach übernehmen können und das für maximale Sicherheit in ihren Apps sorgt. Diese Datensicherheit ist für Datei- und Datenbank-APIs verfügbar, wie z. B. NSFileManager, CoreData, NSData oder SQLite.

Mail (inklusive Anhänge), verwaltete Bücher, App-Start-Images und Ortsdaten werden ebenfalls in verschlüsselter Form gesichert; mit Schlüsseln die mit dem Code des Benutzers für das Gerät geschützt werden. Kalender (ohne Anhänge), Kontakte, Erinnerungen, Notizen, Nachrichten und Fotos verwenden „Geschützt bis zur ersten Benutzerauthentifizierung".

Vom Benutzer installierte Apps, die keine bestimmte Datensicherheitsklasse besitzen, werden standardmäßig der Klasse „Geschützt bis zur ersten Benutzerauthentifizierung" zugeordnet.

Zubehör

Das Lizenzprogramm „Made for iPhone, iPod touch and iPad" (MFi) bietet geprüften Zubehörherstellern Zugriff auf das „iPod Accessories Protocol" (iAP) und die notwendigen unterstützten Hardwarekomponenten.

Wenn ein MFi-Zubehör über einen Lightning-Anschluss oder Bluetooth eine Verbindung zu einem iOS-Gerät herstellen will, muss das Zubehör beweisen, dass es von Apple autorisiert wurde, indem es mit einem Zertifikat von Apple, das vom Gerät überprüft wird, antwortet. Das Gerät sendet anschließend eine Anfrage, auf die das Zubehör eine signierte Antwort senden muss. Dieses Verfahren wird vollständig von einem maßgeschneiderten integrierten Schaltkreis durchgeführt, den Apple zugelassenen Zubehörherstellern zur Verfügung stellt, und ist für das Zubehör selbst transparent.

Zubehör kann Zugriff auf unterschiedliche Übertragungsarten und Funktionen anfordern, zum Beispiel Zugriff auf digitale Audiostreams über das Lightning-Kabel oder Ortsinformationen über Bluetooth. Ein IC (integrierter Schaltkreis) für die Authentifizierung sorgt dafür, dass nur genehmigtes Zubehör vollständigen Zugriff auf das Gerät erhält. Wenn ein Zubehör keine Authentifizierung bietet, erhält es nur auf analoge Audiosignale und begrenzt auf serielle Audiowiedergabesteuerung (UART) Zugriff.

AirPlay verwendet ebenfalls den IC für die Authentifizierung, um zu überprüfen, ob der Empfänger von Apple zugelassen wurde. AirPlay-Audiostreams und CarPlay-Videostreams nutzen das MFi-SAP (Sicherheitsverbindungsprotokoll, *Secure Association Protocol*), mit dem die Kommunikation zwischen dem Zubehör und dem Gerät mit AES-128 im CTR-Modus verschlüsselt wird. Temporäre Schlüssel werden mit dem ECDH-Schlüsselaustausch (Curve25519) ausgetauscht und mit dem 1024-Bit RSA-Schlüssel des ICs für die Authentifizierung als Teil des Station-To-Station-Protokolls signiert.

HomeKit

HomeKit bietet eine Infrastruktur zur Hausautomatisierung, die Sicherheitsmerkmale von iCloud und von iOS nutzt, um Ihre privaten Daten zu schützen und zu synchronisieren, ohne dass Apple darauf zugreifen kann.

HomeKit-Identität

Die HomeKit-Identität und die Sicherheit basieren auf einem öffentlich/privaten Ed25519-Schlüsselpaar. Auf dem iOS-Gerät wird für jeden HomeKit-Benutzer ein Ed25519-Schlüsselpaar erzeugt, das seine HomeKit-Identität darstellt. Es wird für die Kommunikation zwischen iOS-Geräten und/oder Zubehör verwendet.

Die Schlüssel werden im Schlüsselbund gespeichert und sind nur in verschlüsselten Backups des Schlüsselbunds enthalten. Die Schlüssel werden mit dem iCloud-Schlüsselbund geräteübergreifend synchronisiert.

Kommunikation mit HomeKit-Zubehör

HomeKit-Zubehörgeräte erstellen ihr eigenes Ed25519-Schlüsselpaar für die Kommunikation mit iOS-Geräten. Wenn das Zubehör auf die Werkseinstellungen zurückgesetzt wird, wird ein neues Schlüsselpaar erzeugt.

Um eine Verbindung zwischen einem iOS-Gerät und HomeKit-Zubehör herzustellen, werden die Schlüssel über das sichere 3072-Bit-Protokoll „Secure Remote Password“ ausgetauscht, wofür ein achtstelliger Code des Zubehörherstellers verwendet wird, der vom Benutzer auf dem iOS-Gerät eingegeben und anschließend per ChaCha20-Poly1305 AEAD mit von HKDF-SHA-512 abgeleiteten Schlüsseln verschlüsselt wird. Die MFi-Zertifizierung des Zubehörs wird bei der Konfiguration ebenfalls überprüft.

Wenn das iOS-Gerät und das HomeKit-Zubehör bei der Verwendung kommunizieren, authentifizieren sie sich gegenseitig über die zuvor ausgetauschten Schlüssel. Jede Sitzung wird über ein Station-to-Station-Protokoll hergestellt und mit von HKDF-SHA-512 abgeleiteten Schlüsseln, die auf für diese Sitzung erzeugten Curve25519-Schlüsseln basieren, verschlüsselt. Dies gilt für sowohl für IP-basiertes Zubehör als auch für Bluetooth Low Energy-Zubehör.

Lokaler Datenspeicher

HomeKit speichert Daten über Haus, Zubehör, Szenen und Benutzer auf dem iOS-Gerät des Benutzers. Diese gespeicherten Daten werden mit Schlüsseln verschlüsselt, die von den Schlüsseln der HomeKit-Identität und einer zufälligen Nonce abgeleitet werden. Zudem werden HomeKit-Daten mit der Datensicherheitsklasse „Geschützt bis zur ersten Benutzerauthentifizierung“ geschützt. HomeKit-Daten werden nur in verschlüsselten Backups gesichert, sodass sie z. B. nicht in unverschlüsselten iTunes-Backups enthalten sind.

Geräte- und benutzerübergreifende Datensynchronisierung

Die HomeKit-Daten können mit iCloud und dem iCloud-Schlüsselbund für die iOS-Geräte eines Benutzers synchronisiert werden. Die HomeKit-Daten werden bei der Synchronisation mit Schlüsseln verschlüsselt, die von der HomeKit-Identität und einer zufälligen Nonce abgeleitet werden. Diese Daten werden bei der Synchronisation als nicht einsehbare Daten synchronisiert. Die aktuellen Daten werden für die Synchronisation in iCloud gespeichert, aber dort nicht verwendet. Da sie mit Schlüsseln, die nur auf den iOS-Geräten des Benutzers verfügbar sind, verschlüsselt sind, kann auf den Inhalt weder bei der Übertragung noch in iCloud zugegriffen werden.

Die HomeKit-Daten werden auch für mehrere Benutzer im selben Haus synchronisiert. Dieses Verfahren verwendet dieselbe Authentifizierung und Verschlüsselung wie zwischen iOS-Geräten und HomeKit-Zubehör. Die Authentifizierung basiert auf öffentlichen Ed25519-Schlüsseln, die zwischen den Geräten ausgetauscht werden, wenn einem Haus ein Benutzer hinzugefügt wird. Wurde dem Haus ein neuer Benutzer hinzugefügt, wird jede weitere Kommunikation mit dem Station-to-Station-Protokoll und für die Sitzung erzeugten Schlüsseln authentifiziert und verschlüsselt.

Nur der Benutzer, der das Haus in HomeKit angelegt hat, kann neue Benutzer hinzufügen. Sein Gerät konfiguriert das Zubehör mit dem öffentlichen Schlüssel des neuen Benutzers, sodass das Zubehör den neuen Benutzer authentifizieren und Befehle von ihm empfangen kann. Das Verfahren für die Konfiguration von Apple TV für die Verwendung mit HomeKit verwendet dieselbe Authentifizierung wie das Hinzufügen neuer Benutzer, wird aber automatisch durchgeführt, wenn der Benutzer, der das Haus angelegt hat, auf dem Apple TV, das sich im Haus befinden muss, bei iCloud angemeldet ist.

Wenn ein Benutzer nicht mehrere Geräte verwendet und keinen weiteren Benutzern Zugriff auf sein Haus gewährt, werden die HomeKit-Daten nicht in iCloud gesichert.

Hausdaten und Apps

Der Zugriff von Apps auf die Hausdaten wird in den Datenschutzeinstellungen des Benutzers festgelegt. Benutzer werden gefragt, ob Zugriff gewährt werden soll, wenn Apps Hausdaten abfragen wollen, ähnlich wie bei Kontakten, Fotos und anderen iOS-Datenquellen. Stimmt der Benutzer zu, können Apps auf Zimmernamen, Zubehörnamen, Zubehörstandorte und weitere Informationen zugreifen, die in der HomeKit-Entwicklerdokumentation beschrieben werden.

Siri

Siri kann verwendet werden, um Zubehör abzufragen und zu steuern und Szenen zu aktivieren. Es werden so wenig Informationen wie möglich anonym an Siri gesendet, wie im Abschnitt „Siri“ dieses Dokuments beschrieben, sodass die Namen von Zimmern, Zubehör und Szenen für die Erkennung der Befehle verfügbar sind.

HealthKit

Das HealthKit-Framework bietet eine gemeinsame Datenbank, die von Apps mit Zustimmung des Benutzers genutzt werden kann, um Daten zu Fitness und Gesundheit speichern und darauf zugreifen zu können. HealthKit arbeitet auch direkt mit Gesundheits- und Fitnessgeräten, wie kompatible Herzfrequenzmessgeräte, die Bluetooth LE verwenden, und den M7- bzw. M8-Coprozessor, der in viele iOS-Geräte integriert ist.

Gesundheitsdaten

HealthKit verwendet eine Datenbank zum Speichern der Gesundheitsdaten des Benutzers wie Größe, Gewicht, zurückgelegte Entfernung, Blutdruck usw. Diese Datenbank wird mit der Datensicherheitsklasse „Vollständiger Schutz" gespeichert, sodass nur auf sie zugegriffen werden kann, wenn der Benutzer seinen Code eingibt oder das Gerät mit Touch ID entsperrt.

In einer anderen Datenbank werden die Betriebsdaten gespeichert, wie Zugriffstabellen für Apps, Namen der mit HealthKit verbundenen Geräte und Planungsinformationen, die verwendet werden, um Apps zu starten, wenn neue Daten verfügbar werden. Diese Datenbank wird mit der Datensicherheitsklasse „Geschützt bis zur ersten Benutzerauthentifizierung" gespeichert.

In temporären Journal-Dateien werden die Gesundheitsdatensätze gespeichert, die erzeugt werden, wenn das Gerät gesperrt ist, z. B. wenn der Benutzer trainiert. Diese Daten werden mit der Datensicherheitsklasse „Geschützt, außer wenn offen" gespeichert. Wenn das Gerät entsperrt wird, werden sie in die Hauptdatenbank für Gesundheitsdaten importiert und anschließend gelöscht.

Gesundheitsdaten werden nicht über iCloud freigegeben oder geräteübergreifend synchronisiert. Gesundheitsdatenbanken sind in verschlüsselten Geräte-Backups in iCloud oder iTunes enthalten. In nicht verschlüsselten iTunes-Backups sind die Gesundheitsdaten nicht enthalten.

Datenintegrität

Die in der Datenbank gespeicherten Daten enthalten Metadaten, um die Herkunft der Datensätze zurückverfolgen zu können. Zu diesen Metadaten gehören eine App-Kennung, die anzeigt, welche App den Datensatz gespeichert hat. Außerdem kann ein optionales Metadatenobjekt eine digital signierte Kopie des Datensatzes enthalten. Diese Kopie dient der Datenintegrität für Datensätze, die von einem vertrauenswürdigen Gerät erzeugt wurden. Das für die digitale Signatur verwendete Format ist Cryptographic Message Syntax (CMS), wie in IETF RFC 5652 festgelegt.

Zugriff durch Apps anderer Anbieter

Der Zugriff auf die HealthKit-API wird über Berechtigungen gesteuert und Apps müssen sich an Einschränkungen für die Verwendung der Daten halten. Zum Beispiel dürfen Apps Gesundheitsdaten nicht zu Werbezwecken verwenden. Apps müssen dem Benutzer auch eine Datenschutzrichtlinie bereitstellen, die beschreibt, wie die Gesundheitsdaten verwendet werden.

Der Zugriff von Apps auf die Gesundheitsdaten wird in den Datenschutzeinstellungen des Benutzers festgelegt. Benutzer werden gefragt, ob Zugriff gewährt werden soll, wenn Apps Gesundheitsdaten abfragen wollen, ähnlich wie bei Kontakten, Fotos und anderen iOS-Datenquellen. Aber bei Gesundheitsdaten werden Lese- und Schreibzugriff und Zugriff auf die einzelnen Arten von Gesundheitsdaten getrennt voneinander gewährt. Die Benutzer können die erteilten Zugriffsrechte auf Gesundheitsdaten unter „Quellen" in der Health-App einsehen und zurückziehen.

Apps mit Schreibzugriff können auch die von ihnen geschriebenen Daten lesen. Apps mit Lesezugriff können von anderen Apps geschriebene Daten lesen. Apps können aber nicht den Zugriff anderer Apps auf die Daten bestimmen. Außerdem können Apps nicht überprüfen, ob sie Lesezugriff auf Gesundheitsdaten haben. Apps ohne Lesezugriff erhalten keine Antworten auf ihre Anfragen, genau wie bei einer leeren Datenbank. So können die Apps keine Informationen über die Gesundheit des Benutzers davon ableiten, welche Datenarten geschrieben werden.

Notfallpass

Die in iOS 8 enthaltene App „Health" bietet dem Benutzer die Möglichkeit, ein Formular für den Notfallpass mit Informationen auszufüllen, die bei einem Notfall wichtig sein können. Diese Informationen werden manuell eingegeben bzw. aktualisiert und werden nicht mit den Daten in den Gesundheitsdatenbanken synchronisiert.

Die Informationen zum Notfallpass werden angezeigt, wenn Sie auf dem Sperrbildschirm auf die Taste „Notfall" tippen. Die Informationen werden mit der Datensicherheitsklasse „Kein Schutz" auf dem Gerät gespeichert, sie ist also auch ohne Eingabe des Gerätecodes zugänglich. Das Einrichten eines Notfallpasses ist optional und ermöglicht es dem Benutzer, selbst zwischen Sicherheit und Datenschutz abzuwägen.

Apple Watch

Apple Watch verwendet für iOS entwickelte Sicherheitsfunktionen und -technologien zum Schutz der Daten auf dem Gerät und der Kommunikation mit einem verbundenen iPhone und dem Internet. Zu diesen Technologien gehören der Datenschutz und die Schlüsselbundzugriffssteuerung. Der Gerätecode des Benutzers ist auch mit der UID des Geräts verknüpft, um einen Schlüssel zur Verschlüsselung zu erstellen.

Die Kopplung der Apple Watch mit dem iPhone ist durch einen Out-of-Band-Prozess (OOB) gesichert, bei dem öffentliche Schlüssel ausgetauscht werden. Anschließend werden die BTLE-Link Shared Secrets ausgetauscht. Auf der Apple Watch wird ein animiertes Muster angezeigt, das von der Kamera des iPhone aufgenommen wird. Das Muster enthält ein codiertes Secret, das für die BTLE 4.1-Out-of-Band-Kopplung verwendet wird. Als Ersatzfunktion zur Kopplung (falls erforderlich) wird die Eingabe eines BTLE-Schlüssels verwendet.

Nach dem Herstellen der BTLE-Sitzung verwenden Apple Watch und iPhone zum Austausch der Schlüssel einen auf IDS basierenden Prozess, wie im Abschnitt „iMessage" in diesem Dokument beschrieben. Beachten Sie, dass die IDS-Server von Apple nicht verwendet werden, da die Dienste von dem verbundenen iPhone bereitgestellt werden. Nach dem Austauschen der Schlüssel wird der Schlüssel der Bluetooth-Sitzung verworfen und die Kommunikation zwischen Apple Watch und iPhone wird mit IDS verschlüsselt, wobei die BTLE- und WLAN-Links eine zusätzliche Verschlüsselungsebene bereitstellen. Das Key Rolling erfolgt in 15-Minuten-Intervallen, damit das System nur kurzzeitig ohne Schutz ist, falls der Datenverkehr kompromittiert wird.

Zur Unterstützung von Apps, die Daten-Streaming erfordern, erfolgt die Verschlüsselung mithilfe der Methoden, die im Abschnitt „FaceTime" dieses Dokuments beschrieben werden. Hierzu wird der vom verbundenen iPhone bereitgestellte IDS-Dienst verwendet.

Befindet sich die Apple Watch nicht in Reichweite der vom verbundenen iPhone erzeugten Bluetooth-Verbindung, wird iCloud zur Datenweiterleitung zwischen den Geräten verwendet. Apps können festlegen, dass nur eine lokale Kommunikation verwendet werden darf.

Die Apple Watch implementiert eine hardwarebasierte Speicherverschlüsselung und einen klassenbasierten Schutz von Dateien und Schlüsselbundobjekten. Näheres hierzu finden Sie im Abschnitt über die Sicherheit von Daten in diesem Dokument. Ferner werden zugriffsgesteuerte Keybags für Schlüsselbundobjekte verwendet. Die für die Kommunikation zwischen Apple Watch und iPhone verwendeten Schlüssel sind durch einen klassenbasierten Schutz gesichert.

Die Apple Watch stellt erst dann eine Verbindung zu WLANs her, wenn die entsprechenden Anmeldedaten auf dem verbundenen iPhone vorhanden sind, das der Uhr automatisch eine Liste der bekannten Netzwerke bereitstellt.

Die Apple Watch kann durch Drücken und Halten der Seitentaste manuell gesperrt werden. Darüber hinaus wird Bewegungsheuristik verwendet, um das Gerät kurz nach dem Entfernen vom Handgelenk automatisch zu sperren. Die Handgelenkserkennung wird mit der App „Apple Watch“ auf dem iPhone ausgeschaltet. Diese Einstellung kann mithilfe von MDM (Mobile Device Management) erzwungen werden.

Vorausgesetzt, dass die Uhr getragen wird, kann sie auch mit dem verbundenen iPhone entsperrt werden. Dies geschieht, indem eine Verbindung hergestellt wird, die durch die während der Kopplung eingerichteten Schlüssel authentifiziert wird. Das iPhone sendet dann den Schlüssel, den die Uhr zum Entsperren der Schlüssel für den Datenschutz verwendet. Der Gerätecode der Uhr ist dem iPhone nicht bekannt und wird auch nicht übermittelt. Diese Funktion kann mit der App „Apple Watch“ auf dem iPhone deaktiviert werden. Wenn sich der Benutzer auf der Apple Watch für einen komplexen Code entschieden hat, ist diese Funktion nicht verfügbar und der Gerätecode für die Uhr muss eingegeben werden.

Die Apple Watch kann immer nur mit einem iPhone gekoppelt werden. Durch das Koppeln mit einem neuen iPhone werden alle Inhalte und Daten von der Apple Watch entfernt.

Netzwerksicherheit

Zusätzlich zu den integrierten Sicherheitsmaßnahmen, die Apple zum Schutz der auf dem iOS-Gerät gespeicherten Daten verwendet, gibt es viele Netzwerksicherheitsmaßnahmen, die Organisationen ergreifen können, damit Informationen bei der Übertragung an und von einem iOS-Gerät sicher bleiben.

Mobile Benutzer müssen von überall auf der Welt auf Unternehmensnetzwerke zugreifen können. Daher muss sichergestellt werden, dass sie über die entsprechenden Zugriffsrechte verfügen und ihre Daten während der Übertragung zuverlässig geschützt sind. Entwickler können die von iOS verwendeten Standardnetzwerkprotokolle für authentifizierte, autorisierte und verschlüsselte Kommunikation einsetzen. iOS setzt bewährte Technologien und die aktuellen Standards sowohl für WLAN als auch für Mobilfunk-Datenverbindungen ein, um diese Sicherheitsziele zu erreichen.

Auf anderen Plattformen wird Firewall-Software benötigt, um die Kommunikations-Ports vor Angreifern zu schützen. iOS-Geräte benötigen keine zusätzliche Firewall-Software, da die Angriffsfläche durch Beschränkung der Listening-Ports und Entfernen unnötiger Netzwerkdienstprogramme, wie Telnet, Shells oder Webserver reduziert wird.

SSL, TLS

iOS unterstützt Secure Socket Layer (SSL Version 3) sowie Transport Layer Security (TLS Version 1.0, TLS Version 1, TLS Version 1.2) und DTLS. Safari, Kalender, Mail und weitere Internetprogramme verwenden diese Technologien automatisch, um die verschlüsselte Datenübertragung zwischen dem Gerät und Netzwerkdiensten sicherzustellen. High-Level-APIs (wie CFNetwork) erleichtern es Entwicklern, TLS für ihre Apps zu verwenden, im Gegensatz dazu bieten Low-Level-APIs (wie SecureTransport) präzise Einstellungsmöglichkeiten.

VPN

Die Einbindung sicherer Netzwerkdienste, wie „Virtual Private Network“, in iOS erfordert nur minimalen Einrichtungs- und Konfigurationsaufwand. iOS-Geräte arbeiten mit VPN-Servern, die die folgenden Protokolle und Authentifizierungsmethoden bieten:

- Juniper Networks, Cisco, Aruba Networks, SonicWALL, Check Point, Palo Alto Networks, Open VPN, AirWatch, MobileIron, NetMotion Wireless, und F5 Networks SSL-VPN, die die entsprechende Client-App aus dem App Store verwenden.
- Cisco IPSec mit Benutzerauthentifizierung per Passwort, RSA SecurID oder CRYPTOCard, sowie die Systemauthentifizierung über gemeinsame geheime Schlüssel (Shared Secret) und Zertifikate. Cisco IPSec unterstützt VPN On Demand für Domains, die bei der Konfiguration des Geräts angegeben werden.
- L2TP/IPSec mit Benutzerauthentifizierung per MS-CHAPV2-Passwort, RSA SecurID oder CRYPTOCard, sowie die Systemauthentifizierung über gemeinsame geheime Schlüssel (Shared Secret).
- PPTP mit Benutzerauthentifizierung über ein MS-CHAPV2-Passwort, RSA SecurID oder CRYPTOCard.

iOS unterstützt VPN On Demand für Netzwerke, die eine zertifikatsbasierte Authentifizierung verwenden. IT-Richtlinien legen über das verwendete Konfigurationsprofil fest, für welche Domains eine VPN-Verbindung benötigt wird.

Außerdem bietet iOS Unterstützung für VPN pro App, mit der sich VPN-Verbindungen noch detaillierter und präziser einstellen lassen. Mit Mobile Device Management (MDM) kann eine Verbindung für jede verwaltete App und/oder spezielle Domains in Safari bestimmt werden. So lässt sich sicherstellen, dass nur sichere Daten und keine private Daten des Benutzers in das Unternehmensnetzwerk hinein- und daraus hinaus gelangen.

iOS 8 bietet erstmals „VPN immer eingeschaltet". Diese Funktion kann für Geräte konfiguriert werden, die über MDM verwaltet und mit Apple Configurator oder dem Programm zur Geräteregistrierung (DEP) betreut werden. So müssen die Benutzer die Sicherung über VPN nicht manuell aktivieren, wenn sie sich mit WLAN-Netzwerken verbinden. „VPN immer eingeschaltet" verleiht einer Organisation die uneingeschränkte Kontrolle über den Datenverkehr von Geräten, da der gesamte IP-Datenverkehr zurück zur Organisation getunnelt wird. Das Standard-Tunnelingprotokoll, IKEv2, schützt den Datenverkehr mit einer Datenverschlüsselung. Die Organisation kann jetzt den Datenverkehr von und an ihre Geräte überwachen und filtern, Daten im Netzwerk sichern und den Internetzugriff der Geräte beschränken.

WLAN

iOS unterstützt die Branchenstandards für WLAN-Protokolle, darunter WPA2 Enterprise, um authentifizierten Zugriff auf drahtlose Unternehmensnetzwerke bereitzustellen. WPA2 Enterprise nutzt die 128-Bit AES-Verschlüsselung und bietet damit den Benutzern die größte Sicherheit, dass ihre Daten geschützt bleiben, wenn sie Informationen über eine WLAN Netzwerkverbindung senden und empfangen. Da iOS-Geräte den Standard 802.1X unterstützen, lassen sie sich in eine Vielzahl von RADIUS-Authentifizierungsumgebungen integrieren. iPhone und iPad unterstützen unter anderem folgende 802.1X-Identifizierungsverfahren für Funknetzwerke: EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0, PEAPv1 und LEAP.

iOS 8 verwendet eine zufällige MAC-Adresse, wenn PNO-Scans (Preferred Network Offload) durchgeführt werden, das Gerät nicht mit einem WLAN-Netzwerk verbunden ist *und* der Prozessor sich im Ruhezustand befindet. Der Prozessor wird kurz nachdem der Bildschirm abgeschaltet wird, in den Ruhezustand geschaltet. PNO-Scans werden durchgeführt, um zu überprüfen, ob sich der Benutzer mit einem bevorzugten WLAN verbinden kann, um Daten z. B. drahtlos mit iTunes zu synchronisieren.

iOS 8 verwendet außerdem zufällige MAC-Adressen, wenn ePNO-Scans (enhanced Preferred Network Offload) durchgeführt werden, wenn ein Gerät nicht mit einem WLAN verbunden ist *oder* sich der Prozessor im Ruhezustand befindet. ePNO-Scans werden durchgeführt, wenn ein Gerät die Ortungsdienste für Apps, die Geofences nutzen, verwendet, z. B. ortsbasierte Erinnerungen, die feststellen, ob das Gerät an einem bestimmten Ort sein muss.

Da sich die MAC-Adresse des Geräts ändert, wenn es nicht mit einem WLAN verbunden ist, kann sie nicht dafür verwendet werden, um über die passive Beobachtung des WLAN-Datenverkehrs ein Bewegungsprofil für ein Gerät zu erstellen, selbst wenn es mit dem Mobilfunknetz verbunden ist.

Wir kooperieren mit WLAN-Herstellern, damit diese wissen, welche Hintergrundscans eine zufällige MAC-Adresse verwenden, aber die zufällige MAC-Adresse kann weder von Apple noch vom Hersteller vorhergesagt werden.

Zufällige MAC-Adressen für WLAN werden auf iPhone 5c, 5s, 6 und 6 Plus, iPad Air und iPad mini mit Retina-Display unterstützt.

Bluetooth

Die Bluetooth-Unterstützung in iOS bietet nützliche Funktionen ohne unnötig erhöhten Zugriff auf private Daten. iOS-Geräte unterstützen Verbindungen über Verschlüsselungsmodus 3, Sicherheitsmodus 4 und Servicelevel 1. iOS unterstützt die folgenden Bluetooth-Profile:

- Hands-Free Profile (HFP 1.6)
- Phone Book Access Profile (PBAP)
- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP 1.4)
- Personal Area Network Profile (PAN)
- Human Interface Device Profile (HID)
- Message Access Profile (MAP)

Die Unterstützung für diese Profile hängt vom jeweiligen Gerät ab. Weitere Information finden Sie unter: https://support.apple.com/kb/ht3647?viewlocale=de_DE.

Single-Sign-On

iOS unterstützt die Gesamtauthentifizierung (Single Sign-On, SSO) für Unternehmensnetzwerke. SSO kann bei auf Kerberos basierenden Netzwerken verwendet werden, um Benutzer für Dienste, auf die sie zugreifen dürfen, zu authentifizieren. SSO kann für eine Reihe verschiedener Netzwerkaktivitäten verwendet werden, z. B. in sicheren Safari-Sitzungen oder in Apps anderer Anbieter.

Die Gesamtauthentifizierung in iOS nutzt SPNEGO-Token und das Protokoll HTTP Negotiate, um auf Kerberos basierende Authentifizierungsgateways und in Windows integrierte Authentifizierungssysteme, die Kerberos-Tickets unterstützen, verwenden zu können. Die zertifikatsbasierte Authentifizierung wird ebenfalls unterstützt. Die SSO-Unterstützung basiert auf dem Open-Source-Projekt Heimdal.

Es werden die folgenden Verschlüsselungsarten unterstützt:

- AES128-CTS-HMAC-SHA1-96
- AES256-CTS-HMAC-SHA1-96
- DES3-CBC-SHA1
- ARCFOUR-HMAC-MD5

Safari unterstützt die Gesamtauthentifizierung und Apps anderer Anbieter, die die Standard-Netzwerk-APIs von iOS verwenden, können ebenfalls für sie konfiguriert werden. Für die Konfiguration der SSO unterstützt iOS ein neues Konfigurationsprofil, das es MDM-Servern erlaubt, die notwendigen Einstellungen für die Gesamtauthentifizierung auf ein Gerät zu pushen. Dazu gehören auch Einstellungen zum Benutzerprinzipalnamen (d. h. der Benutzeraccount von Active Directory) und zum Kerberos Realm sowie Konfigurationen, die festlegen, welche Apps bzw. Safari Web-URLs Single Sign On verwenden dürfen.

Sicherheit bei AirDrop

iOS-Geräte mit AirDrop-Unterstützung verwenden Bluetooth Low Energy (BLE) und von Apple entwickelte Peer-To-Peer-WLAN-Technologien, um Dateien und Information auf Geräte in der Nähe zu senden, zu denen auch Mac-Computer mit AirDrop-Unterstützung und OS X Yosemite gehören. Die Geräte kommunizieren direkt per WLAN-Funk miteinander, ohne dass eine Internetverbindung oder ein WLAN-Zugangspunkt benötigt wird.

Wenn der Benutzer AirDrop aktiviert, wird auf dem Gerät eine 2048-Bit RSA-Identität gespeichert. Außerdem wird eine AirDrop-Hash-Identität erstellt, die auf den E-Mail-Adressen und Telefonnummern basiert, die mit der Apple-ID des Benutzers verknüpft sind.

Wenn ein Benutzer AirDrop für die Freigabe eines Objekts verwendet, sendet das Gerät per Bluetooth Low Energy ein AirDrop-Signal aus. Andere Geräte, die sich in der unmittelbaren Umgebung und nicht im Ruhezustand befinden und bei denen AirDrop aktiviert ist, erkennen das Signal und antworten mit einer Kurzversion der Hash-Identität ihres Eigentümers.

AirDrop verwendet standardmäßig die Freigabeeinstellung „Nur Kontakte“. Benutzer können auch die Freigabe für „Alle“ auswählen oder die Funktion komplett deaktivieren. Im Modus „Nur Kontakte“ wird die empfangene Hash-Identität mit den Personen in den Kontakten des Initiators abgeglichen. Bei einem Treffer erstellt der Sender ein P2P-WLAN-Netzwerk und gibt die AirDrop-Verbindung über Bonjour bekannt. Über diese Verbindung senden die Empfänger ihre vollständige Hash-Identität an den Initiator. Stimmt die vollständige Hash-Identität ebenfalls mit den Kontakten überein, werden Vorname und Foto (soweit in den Kontakten vorhanden) auf der AirDrop-Freigabeseite angezeigt.

In AirDrop bestimmt der Benutzer, für wen Inhalte freigegeben werden sollen. Der Sender startet eine verschlüsselte (TLS) Verbindung mit dem Empfänger, über die die iCloud-Identitätszertifikate ausgetauscht werden. Die Identität in den Zertifikaten wird mit den Kontakten der Benutzer abgeglichen. Anschließend wird der Empfänger gebeten, die eingehende Dateiübertragung von der identifizierten Person/dem identifizierten Gerät anzunehmen. Wenn mehrere Empfänger ausgewählt wurden, wird dieses Verfahren für jeden einzelnen wiederholt.

Im Modus „Alle“ wird dasselbe Verfahren verwendet, aber wenn in den Kontakten kein Treffer gefunden wird, werden die Empfänger auf der AirDrop-Freigabeseite als Silhouette mit dem Namen des Geräts angezeigt, der unter „Systemeinstellungen“ > „Allgemein“ > „Über“ > „Name“ festgelegt wird.

Internetdienste

Erstellen eines sicheren Passworts für die Apple-ID

Apple-IDs werden für die Anmeldung bei verschiedenen Diensten, darunter iCloud, FaceTime und iMessage, verwendet. Alle Passwörter für neue Accounts müssen die folgenden Attribute besitzen, damit das Passwort sicher ist:

- Mindestens acht Zeichen
- Mindestens ein Buchstabe
- Mindestens ein Großbuchstabe
- Mindestens eine Ziffer
- Maximal drei identische Zeichen hintereinander
- Darf nicht mit dem Accountnamen identisch sein

Apple hat eine Reihe zuverlässiger Dienste ins Leben gerufen, mit denen Benutzer ihre Geräte besser und produktiver nutzen können. Dazu zählen iMessage, FaceTime, Siri, Spotlight-Vorschläge, iCloud, iCloud-Backup und iCloud-Schlüsselbund.

Diese Internetdienste wurden mit denselben Sicherheitszielen entwickelt, die auf der gesamten iOS-Plattformen gelten. Dazu zählen die sichere Verarbeitung von Daten, unabhängig davon, ob sie auf dem Gerät gespeichert sind oder über Funknetzwerke übertragen werden; der Schutz der privaten Daten des Benutzers sowie der Schutz vor unbefugten Zugriffen auf Daten und Dienste. Jeder Dienst verwendet eine eigene leistungsstarke Sicherheitsarchitektur, ohne dadurch insgesamt die Benutzerfreundlichkeit von iOS zu beeinträchtigen.

Apple-ID

Eine Apple-ID besteht aus einem Benutzernamen und einem Passwort, die verwendet werden, um sich bei verschiedenen Apple-Diensten wie iCloud, iMessage, FaceTime, iTunes Store, App Store, iBooks Store und mehr anzumelden. Es ist von entscheidender Bedeutung, dass Benutzer ihre Apple-ID sicher aufbewahren, um unbefugten Zugriff auf den Account zu verhindern. Um den Benutzern dabei zu helfen, sichere Passwörter zu erstellen, schreibt Apple vor, dass sichere Passwörter mindestens achtstellig sein müssen, sowohl Buchstaben als auch Zahlen enthalten müssen, nicht mehr als dreimal in Folge das gleiche Zeichen verwenden dürfen und kein häufig genutztes Passwort sein dürfen. Benutzer können gerne über diese Richtlinien hinaus (Sonder)zeichen hinzufügen, um ein noch sichereres Passwort zu erhalten. Apple sendet auch E-Mails und Push-Benachrichtigungen an Benutzer, wenn wichtige Änderungen am Account durchgeführt werden, z. B. wenn Passwort oder Rechnungsdaten geändert wurden oder die Apple-ID auf einem neuen Gerät für die Anmeldung verwendet wurde. Wenn Benutzern etwas seltsam erscheint, sollten sie sofort das Passwort für ihre Apple-ID ändern.

Apple bietet auch zweistufige Verifizierung der Apple-ID an, die für zusätzlichen Schutz des Accounts des Benutzers sorgt. Bei aktivierter zweistufiger Verifizierung muss die Identität des Benutzers mithilfe eines temporären Codes, der an eines der vertrauenswürdigen Geräte gesendet wird, bestätigt werden, bevor von einem neuen Gerät aus die Accountinformationen geändert werden können, der Benutzer sich bei iCloud anmelden oder Einkäufe im iTunes Store, iBooks Store oder dem App Store tätigen kann. Dadurch wird der Zugriff auf den Benutzeraccount für Unbefugte selbst dann verhindert, wenn sie das Passwort besitzen. Zusätzlich erhalten Benutzer einen vierzehnstelligen Wiederherstellungsschlüssel zur sicheren Aufbewahrung für den Fall, dass sie ihr Passwort vergessen oder nicht mehr auf die vertrauenswürdigen Geräte zugreifen können.

Weitere Informationen zur zweistufigen Verifizierung für Apple-ID finden Sie unter: https://support.apple.com/kb/ht5570?viewlocale=de_DE.

iMessage

iMessage ist ein Messaging-Dienst für iOS Geräte und Mac-Computer, der Text und Anhänge wie Fotos, Kontakte und Standorte unterstützt. Nachrichten werden auf allen registrierten Geräten des Benutzers angezeigt, sodass ein Chat auf jedem beliebigen Gerät des Benutzers fortgesetzt werden kann. iMessage nutzt den Apple-Dienst für Push-Benachrichtigungen (Apple Push Notification Service – APNS) vollumfänglich aus. Apple zeichnet keine Nachrichten oder Anhänge auf und der Inhalt wird mit einer End-to-End-Verschlüsselung geschützt, sodass nur Sender und Empfänger darauf zugreifen können. Apple kann die Daten nicht entschlüsseln.

Aktiviert ein Benutzer auf einem Gerät iMessage, erzeugt es zwei Schlüsselpaare für den Dienst: einen RSA 1280-Bit-Schlüssel für die Verschlüsselung und einen ECDSA 256-Bit-Schlüssel auf der NIST P-256-Kurve für die Signatur. Die privaten Schlüssel für alle Schlüsselpaare werden im Schlüsselbund des Geräts gespeichert und die öffentlichen Schlüssel werden an den Verzeichnisdienst von Apple (IDS) gesendet, wo sie mit der Telefonnummer oder E-Mail-Adresse des Benutzers und der Adresse des Geräts für den APNS Dienst verknüpft werden.

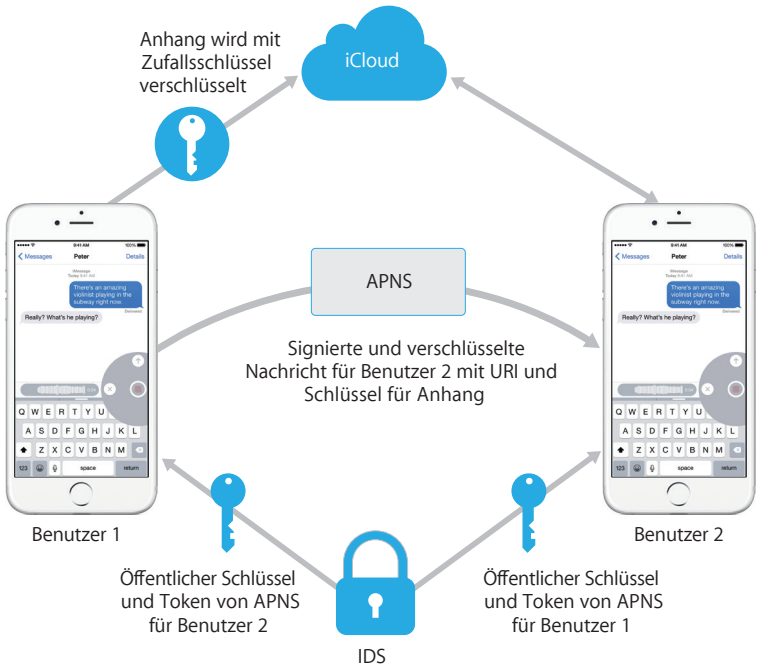
Wenn Benutzer zusätzliche Geräte für iMessage aktivieren, werden ihre Verschlüsselung und öffentlichen Schlüssel für die Signatur, APNS Adressen und verknüpften Telefonnummern ebenfalls dem Verzeichnisdienst hinzugefügt. Benutzer können auch mehrere E-Mail-Adressen hinzufügen, die über den Link in einer Bestätigungsmail verifiziert werden. Telefonnummern werden über das Mobilfunknetz und die SIM-Karte verifiziert. Außerdem zeigen alle registrierten Geräte eines Benutzers eine Warnung an, wenn ein neues Gerät, eine neue Telefonnummer oder eine neue E-Mail-Adresse hinzugefügt werden.

So sendet und empfängt iMessage Nachrichten

Benutzer können in iMessage eine neue Konversation starten, indem sie eine Adresse oder einen Namen eingeben. Wenn sie eine Telefonnummer oder eine E-Mail-Adresse eingeben, kontaktiert das Gerät den Verzeichnisdienst, um die öffentlichen Schlüssel und APNS Adressen für alle mit dem Benutzer verknüpften Geräte abzurufen. Wenn der Benutzer einen Namen eingibt, sucht das Gerät zuerst in den Kontakten des Benutzers nach Telefonnummern und E-Mail-Adressen, die mit diesem Namen verknüpft sind und ruft dann die öffentlichen Schlüssel und APNS Adressen vom IDS ab.

Die vom Benutzer gesendeten Nachrichten werden für alle Geräte des Empfängers einzeln verschlüsselt. Die öffentlichen RSA-Schlüssel der Empfangsgeräte werden vom IDS abgerufen. Das Sendegerät erzeugt für jedes Empfangsgerät einen zufälligen 128-Bit-Schlüssel und verschlüsselt die Nachricht damit mit AES im CTR-Modus. Dieser pro Nachricht erzeugte AES-Schlüssel wird mit RSA-OAEP für den öffentlichen Schlüssel des Empfangsgeräts verschlüsselt. Die Kombination aus dem verschlüsselten Nachrichtentext und dem verschlüsselten Nachrichtenschlüssel wird mit SHA-1 hash-codiert und der Hash wird mittels ECDSA mit dem privaten Signaturschlüssel des Sendegerät signiert. Die dadurch entstehenden Nachrichten, je eine pro Empfangsgerät, bestehen aus dem verschlüsselten Nachrichtentext, dem verschlüsselten Nachrichtenschlüssel und der digitalen Signatur des Senders. Sie werden dann an APNS zur Weitersendung übertragen. Metadaten wie der Zeitstempel und die APNS Routing-Informationen werden nicht verschlüsselt. Die Kommunikation mit APNS wird über einen TLS-Kanal mit Forward Secrecy verschlüsselt.

APNS können nur Nachrichten weiterleiten, die bis zu 4 KB bzw. 16 KB (abhängig von der iOS Version) groß sind. Wenn der Nachrichtentext zu lang ist oder einen Anhang, z. B. ein Foto, enthält wird der Anhang mit einem per AES im CTR-Modus zufällig erzeugten 256-Bit-Schlüssel verschlüsselt und in iCloud hochgeladen. Der AES-Schlüssel für den Anhang, seine URI (Uniform Resource Identifier) und ein SHA-1-Hash der verschlüsselten Form werden an den Empfänger als Inhalt der iMessage gesendet, wobei Vertraulichkeit und Integrität mit der normalen iMessage-Verschlüsselung geschützt werden, die im Folgenden beschrieben wird.



Bei Gruppenkonversationen wird dieser Prozess für jeden Empfänger und jedes Gerät wiederholt.

Auf der Empfängerseite erhält jedes Gerät eine Kopie der Nachricht vom APNS und ruft gegebenenfalls den Anhang in iCloud ab. Die Telefonnummer oder E-Mail-Adresse des Absenders wird mit den Kontakten des Empfängers abgeglichen, sodass nach Möglichkeit ein Name angezeigt werden kann.

Wie bei allen Push-Benachrichtigungen, wird die Nachricht nach der Zustellung beim APNS gelöscht. Im Gegensatz zu anderen APNS Mitteilungen werden iMessage Nachrichten in eine Warteliste eingefügt, wenn sich das Empfänger-Gerät im Offline-Modus befindet. Nachrichten werden derzeit für bis zu 30 Tage gespeichert.

FaceTime

FaceTime ist der Dienst für Video- und Audioanrufe von Apple. Für FaceTime-Anrufe wird, genau wie für iMessage, der Apple-Dienst für Push-Benachrichtigungen zum Herstellen der Verbindung mit den registrierten Geräten des Benutzers verwendet. Die Audio/Video-Inhalte eines FaceTime-Anrufs werden mit einer End-to-End-Verschlüsselung geschützt, sodass nur Sender und Empfänger auf sie zugreifen können. Apple kann die Daten nicht entschlüsseln.

FaceTime verwendet Internet Connectivity Establishment (ICE), um eine Peer-To-Peer-Verbindung zwischen Geräten herzustellen. Die Geräten verifizieren ihre Identitätszertifikate und erstellen ein Shared Secret für jede Sitzung mit SIP-Nachrichten (Session Initiation Protocol). Die kryptografischen Nonces aller Geräte werden für jeden Medienkanal zu Salt-Schlüsseln kombiniert, die über ein Secure Real Time Protocol (SRTP) mit einer AES-256 Verschlüsselung übertragen werden.

iCloud

iCloud speichert Kontakte, Kalender, Fotos, Dokumente und mehr für den Benutzer und synchronisiert die Daten geräteübergreifend. iCloud kann auch von Apps anderer Anbieter verwendet werden, um Dokumente und Schlüssel/Werte für App-Daten, wie vom Entwickler festgelegt, zu speichern und zu synchronisieren. Der Benutzer kann iCloud einrichten, indem er sich mit einer Apple-ID anmeldet und auswählt, welche

Dienste er verwenden will. Funktionen von iCloud, z. B. „Mein Fotostream", iCloud Drive und iCloud-Backup können von IT-Administratoren über ein Konfigurationsprofil deaktiviert werden. Der Dienst behandelt alle Dateiinhalte identisch; als eine Ansammlung von Bytes.

Jede Datei wird in einzelne Teile zerlegt, die von iCloud mit AES-128 und einem SHA-256 Schlüssel, der von den Inhalten der einzelnen Teile abgeleitet wird, verschlüsselt werden. Die Schlüssel und die Metadaten der Datei werden von Apple im iCloud-Account des Benutzers gespeichert. Die verschlüsselten Teile der Datei werden ohne Informationen, über die der Benutzer identifiziert werden kann, auf Speicherdiensten anderer Anbieter, z. B. Amazon S3 und Windows Azure, gespeichert.

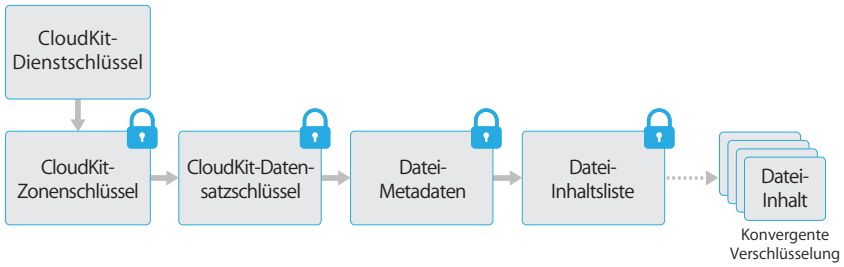
iCloud Drive

iCloud Drive führt accountbasierte Schlüssel ein, um in iCloud gespeicherte Dokumente zu schützen. Wie bei bestehenden iCloud-Diensten, werden die Inhalte der Datei aufgeteilt, verschlüsselt und in Diensten anderer Anbieter gespeichert. Die Schlüssel für die Dateiinhalte werden hingegen mit Datensatzschlüsseln verpackt, die zusammen mit den iCloud Drive-Metadaten gespeichert werden. Diese Datensatzschlüssel werden wiederum mit dem Serviceschlüssel des Benutzers für iCloud Drive geschützt, der zusammen mit dem iCloud-Account des Benutzers gespeichert wird. Benutzer können auf die Metadaten ihrer iCloud-Dokumente zugreifen, wenn sie sich in iCloud authentifiziert haben. Sie müssen jedoch auch den Serviceschlüssel für iCloud Drive besitzen, um die geschützten Teile des iCloud Drive-Speichers zu öffnen.

CloudKit

CloudKit ermöglicht es Entwicklern von Apps, Schlüssel/Wert-Daten, strukturierte Daten und Medien in iCloud zu sichern. Der Zugriff auf CloudKit wird mit App-Berechtigungen gesteuert. CloudKit unterstützt sowohl öffentliche als auch private Datenbanken. Öffentliche Datenbanken werden von allen Kopien der App verwendet, meist für allgemeine Materialien, und nicht verschlüsselt. Private Datenbanken speichern die Daten des Benutzers.

Wie bei iCloud Drive verwendet CloudKit accountbasierte Schlüssel, um die Informationen zu sichern, die in der privaten Datenbank des Benutzers gespeichert werden. Die Daten werden, wie bei anderen iCloud-Diensten, aufgeteilt, verschlüsselt und mit Diensten anderer Anbieter gespeichert. CloudKit verwendet eine Schlüsselhierarchie, ähnlich wie die Sicherheit von Dateidaten. Die pro Datei erzeugten Schlüssel werden mit Schlüsseln der CloudKit-Datensatzeinträge verpackt. Diese Datensatzschlüssel werden wiederum von einem Schlüssel für den Bereich geschützt, der mit dem Schlüssel des Benutzers für den Dienst „CloudKit" geschützt ist. Der Schlüssel für den Dienst „CloudKit" wird im iCloud-Account des Benutzers gespeichert und ist erst dann verfügbar, wenn sich der Benutzer mit iCloud authentifiziert hat.



iCloud-Backup

iCloud sichert auch Informationen – einschließlich Geräteeinstellungen, App-Daten, Fotos und Videos in „Aufnahmen“ sowie Konversationen in der App „Nachrichten“ – täglich per WLAN. iCloud sichert Inhalte, indem sie verschlüsselt über das Internet gesendet und im verschlüsselten Format gespeichert werden. Zur Authentifizierung werden sichere Token verwendet. iCloud-Backup wird nur ausgeführt, wenn das Gerät gesperrt und mit einer Stromquelle verbunden ist und WLAN Zugriff auf das Internet besteht. Aufgrund der in iOS verwendeten Verschlüsselung ist das System auf den Schutz der Daten ausgelegt. Es ermöglicht inkrementelle, unbeaufsichtigte Sicherungen und Wiederherstellungen.

iCloud sichert die folgenden Inhalte:

- Informationen zu gekaufter Musik, Filmen, TV-Sendungen, Apps und Büchern, jedoch nicht die gekauften Inhalte selbst
- Fotos und Videos in „Aufnahmen“
- Kontakte, Kalendereereignisse, Erinnerungen und Notizen
- Geräteeinstellungen
- App-Daten
- PDF-Dateien und Bücher, die iBooks hinzugefügt, aber nicht gekauft wurden
- Anruflisten
- Home-Bildschirm und Anordnung der Apps
- iMessages, SMS und MMS
- Klingeltöne
- HomeKit-Daten
- HealthKit-Daten
- Visual Voicemail

Wenn Dateien in Datensicherheitsklassen erstellt werden, auf die bei gesperrtem Gerät nicht zugegriffen werden kann, werden ihre pro Datei erzeugten Schlüssel mit den Klassenschlüssel des iCloud-Backup-Keybags verschlüsselt. Die Datei werden im ursprünglichen, verschlüsselten Zustand in iCloud gesichert. Dateien der Datensicherheitsklasse „Kein Schutz“ werden bei der Übertragung verschlüsselt.

Der iCloud-Backup-Keybag enthält asymmetrische Schlüssel (Curve25519) für alle Datensicherheitsklassen, die zur Verschlüsselung der pro Datei erzeugten Schlüssel verwendet werden. Weitere Informationen zum Inhalt des Backup-Keybags und des iCloud-Backup-Keybags finden Sie unter „Schutz von Schlüsselbunddaten“ im Abschnitt „Verschlüsselung und Datensicherheit“.

Die Sicherungen werden im iCloud-Account des Benutzers gespeichert und bestehen aus einer Kopie der Dateien des Benutzers und dem iCloud Backup-Keybag. Der iCloud Backup-Keybag wird mit einem zufälligen Schlüssel geschützt, der mit den Sicherungen gespeichert wird. (Das iCloud-Passwort des Benutzers wird nicht für die Verschlüsselung eingesetzt, sodass bestehende Sicherungen bei einer Passwortänderung nicht ungültig werden.)

Die Schlüsselbund-Datenbank des Benutzers wird in iCloud gesichert, wird aber von einem mit der UID verknüpften Schlüssel geschützt. Dadurch kann der Schlüsselbund auf dem Gerät, von dem er ursprünglich stammt, wiederhergestellt werden und niemand außer dem Benutzer, noch nicht einmal Apple, kann die Objekte im Schlüsselbund des Benutzers lesen.

Bei der Wiederherstellung, werden die gesicherten Dateien, der iCloud-Backup-Keybag und der Schlüssel für den Keybag vom iCloud-Account des Benutzers abgerufen. Der iCloud-Backup-Keybag wird mit seinem Schlüssel entschlüsselt, anschließend werden die pro Datei erzeugten Schlüssel im Keybag verwendet, um die Dateien in den Sicherungen zu entschlüsseln, die als neue Dateien auf das Dateisystem geschrieben und so gemäß ihrer Datensicherheitsklasse neu verschlüsselt werden.

Safari-Integration mit dem iCloud-Schlüsselbund
Safari kann automatisch kryptografisch sichere zufällige Zeichenfolgen als Passwörter für Websites erzeugen, die im Schlüsselbund gesichert und mit Ihren anderen Geräten synchronisiert werden. Schlüsselbundobjekte werden von einem Gerät über die Apple Server auf ein anderes Gerät übertragen, werden dabei aber so verschlüsselt, dass weder Apple noch andere Geräte den Inhalt lesen können.

iCloud-Schlüsselbund

Der iCloud-Schlüsselbund ermöglicht es Benutzern, ihre Passwörter sicher zwischen iOS-Geräten und Mac-Computern zu synchronisieren, ohne dass Apple diese Informationen einsehen kann. Zusätzlich zu Datenschutz und Sicherheit waren Benutzerfreundlichkeit und Wiederherstellbarkeit eines Schlüsselbundes weitere Ziele, die sich nachhaltig auf Konzeption und Architektur des iCloud-Schlüsselbundes ausgewirkt haben. Der iCloud-Schlüsselbund umfasst zwei Dienste: Schlüsselbundsynchronisierung und Schlüsselbundwiederherstellung.

Apple hat den iCloud-Schlüsselbund und die Schlüsselbundwiederherstellung so konzipiert, dass die Passwörter selbst unter den folgenden Umständen sicher sind:

- Der iCloud-Account eines Benutzers wurde kompromittiert.
- iCloud wird von einem Angreifer von außen oder einem Mitarbeiter kompromittiert.
- Dritte erlangen Zugriff auf Benutzeraccounts.

Schlüsselbundsynchronisierung

Aktiviert ein Benutzer den iCloud-Schlüsselbund das erste Mal, richtet das Gerät einen „Circle of Trust“ ein und erstellt für sich eine Synchronisationsidentität. Diese Synchronisationsidentität besteht aus einem privaten und einem öffentlichen Schlüssel. Der öffentliche Schlüssel dieser Synchronisationsidentität wird Teil des „Circle of Trust“ und dieser wird zweimal signiert: zuerst mit dem privaten Schlüssel der Synchronisationsidentität und anschließend mittels asymmetrischer Elliptische-Kurven-Kryptografie (P256) mit einem weiteren Schlüssel, der vom Passwort für den iCloud-Account des Benutzers abgeleitet wird. Außerdem werden im Circle die Parameter gespeichert, mit denen der Schlüssel aus dem Passwort für den iCloud-Account erstellt wurde (Salt und Iterationen).

Der signierte Circle wird in den iCloud-Schlüssel/Wert-Speicher gegeben. Er kann ohne das iCloud-Passwort nicht ausgelesen werden und ohne den privaten Schlüssel der Synchronisationsidentität der Mitglieder nicht gültig verändert werden.

Aktiviert der Benutzer den iCloud-Schlüsselbund auf einem anderen Gerät, sieht es in iCloud, dass der Benutzer einen Circle of Trust eingerichtet hat, zu dem es nicht gehört. Das Gerät erstellt ein Schlüsselpaar für die Synchronisationsidentität und beantragt anschließend die Aufnahme in den Circle. Das Ticket dafür besteht aus dem öffentlichen Schlüssel der Synchronisationsidentität und der Benutzer wird zur Bestätigung mit dem iCloud-Passwort aufgefordert. Die Parameter für die Elliptische-Kurven-Kryptografie werden von iCloud abgerufen. Anschließend wird aus ihnen ein Schlüssel erzeugt, mit dem das Ticket signiert wird. Schließlich wird das Ticket in iCloud gespeichert.

Sobald das erste Gerät das neue Ticket erkennt, zeigt es dem Benutzer eine Mitteilung, dass ein neues Gerät dem Circle of Trust beitreten möchte. Der Benutzer gibt sein iCloud-Passwort ein und das Ticket wird mit dem entsprechenden privaten Schlüssel verifiziert. Dadurch wird nachgewiesen, dass die Anfrage zur Aufnahme in den Circle mit dem iCloud-Passwort des Benutzers bestätigt wurde.

Wenn der Benutzer die Aufnahme des neuen Geräts bestätigt hat, fügt das erste Gerät den öffentlichen Schlüssel des neuen Mitglieds dem Circle hinzu und signiert erneut mit der Synchronisationsidentität und dem Schlüssel, der aus dem iCloud-Passwort des Benutzers abgeleitet wurde. Der neue Circle of Trust wird dann in iCloud gespeichert und dort analog von dem neuen Mitglied signiert.

Der Circle besteht nun aus zwei Mitglieder, von denen jedes den öffentlichen Schlüssel des anderen besitzt. Sie tauschen nun untereinander einzelne Schlüsselbundobjekte über den iCloud-Schlüssel/Wert-Speicher aus. Ist ein Objekt bei beiden vorhanden, wird das zuletzt geänderte Objekt synchronisiert. Wenn das andere Mitglied das gleiche Objekt besitzt und zur gleichen Zeit zuletzt geändert wurde, wird das Objekt übersprungen. Jedes synchronisierte Objekt wird speziell für das Gerät, an das es gesendet wird, verschlüsselt. Weder andere Geräte noch Apple können es entschlüsseln. Zudem ist das verschlüsselte Objekt in iCloud nur temporär und wird mit jedem neuen Objekt, das synchronisiert wird, überschrieben.

Dieser Prozess wird für jedes neue Gerät im Circle wiederholt. Wenn z. B. ein drittes Gerät beitritt, erscheint die Bestätigungsmitteilung auf beiden anderen Geräten des Benutzers. Der Benutzer kann das Mitglied von einem der beiden anderen Geräte bestätigen. Werden neue Mitglieder hinzugefügt, synchronisieren sich alle Geräte mit diesem, damit alle Mitglieder dieselben Schlüsselbundobjekte verwenden.

Es wird jedoch nicht der gesamte Schlüsselbund synchronisiert. Einige Objekte sind gerätespezifisch, z. B. VPN-Identitäten, und sollten das Gerät nicht verlassen. Nur Objekte mit dem Attribut `kSecAttrSynchronizable` werden synchronisiert. Apple hat dieses Attribut für Safari Benutzerdaten gesetzt (dazu gehören Benutzernamen, Passwörter und Kreditkartennummern), sowie für WLAN-Passwörter und HomeKit-Schlüssel.

Außerdem werden Schlüsselbundobjekte, die von Apps anderer Anbieter hinzugefügt wurden, standardmäßig nicht synchronisiert. Entwickler müssen das Attribut `kSecAttrSynchronizable` verwenden, wenn Objekte dem Schlüsselbund hinzugefügt werden sollen.

Schlüsselbundwiederherstellung

Die Schlüsselbundwiederherstellung ermöglicht es den Benutzern, ihren Schlüsselbund bei Apple treuhänderisch zu hinterlegen (escrow), ohne dass Apple die Passwörter und andere darin enthaltene Daten lesen kann. Sogar dann, wenn der Benutzer nur ein einzelnes Gerät hat, bietet die Schlüsselbundwiederherstellung Schutz vor möglichem Datenverlust. Das ist besonders wichtig, wenn Safari verwendet wird, um zufällige, sichere Passwörter für Webaccounts zu generieren, weil diese ausschließlich im Schlüsselbund aufgezeichnet werden.

Ein Eckpfeiler der Schlüsselbundwiederherstellung ist die sekundäre Authentifizierung und ein sicherer Treuhanddienst (escrow), der von Apple speziell für diese Funktion erstellt wurde. Der Schlüsselbund des Benutzers wird mithilfe eines sicheren Codes verschlüsselt, und der Treuhanddienst stellt nur dann eine Kopie des Schlüsselbundes bereit, wenn eine Reihe strikter Bedingungen erfüllt ist.

Bei der Aktivierung des iCloud-Schlüsselbunds wird der Benutzer gebeten, einen iCloud-Sicherheitscode zu erstellen. Dieser Code wird benötigt, um einen Escrow-Schlüsselbund wiederherzustellen. Standardmäßig wird der Benutzer gebeten, einen einfachen Sicherheitscode aus vier Zahlen zu erstellen. Benutzer können aber auch eigene, längere Codes erstellen oder von ihren Geräten einen zufälligen kryptografischen Code erzeugen lassen, den sie selbst speichern.

Anschließend exportiert das iOS-Gerät eine Kopie des Schlüsselbundes des Benutzers wobei die Schlüssel in einem asymmetrischen Keybag verpackt werden und in den iCloud Schlüsselwertspeicher des Benutzers gelegt werden. Der Keybag wird mit dem iCloud-Sicherheitscode des Benutzers und dem öffentlichen Schlüssel des HSM-Clusters (Hardwaresicherheitsmodul), der den Escrow-Eintrag speichert, verpackt. Daraus setzt sich der iCloud Escrow-Eintrag des Benutzers zusammen.

Wenn der Benutzer sich dafür entscheidet, einen zufälligen kryptografischen Sicherheitscode anstelle eines eigenen oder eines vierstelligen Werts zu verwenden, wird kein Escrow-Eintrag benötigt. Stattdessen wird der zufällige Schlüssel direkt mit dem iCloud-Sicherheitscode verpackt.

Zusätzlich zum Erstellen des Sicherheitscodes muss der Benutzer eine Telefonnummer registrieren. Dies dient als zusätzliche Authentifizierungsebene bei der Schlüsselbundwiederherstellung. Der Benutzer erhält einen Code in einer SMS, der eingegeben werden muss, damit der Wiederherstellungsprozess fortgesetzt werden kann.

Escrow-Sicherheit

Mit der sicheren iCloud Infrastruktur wird beim treuhänderischen Hinterlegen des Schlüsselbunds sichergestellt, dass nur autorisierte Benutzer und Geräte eine Wiederherstellung durchführen können. Hinter iCloud stehen HSM-Cluster (Hardwaresicherheitsmodule). Mit diesen Clustern werden die Escrow-Einträge geschützt. Jedes besitzt einen Schlüssel, mit dem die geschützten Escrow-Einträge verschlüsselt werden, wie bereits beschrieben.

Um einen Schlüsselbund wiederherzustellen, muss sich der Benutzer mit seinem iCloud-Account, seinem Passwort und dem Code aus der an die registrierte Telefonnummer gesendeten Code authentifizieren. Anschließend muss der Benutzer seinen iCloud-Sicherheitscode eingeben. Der HSM-Cluster überprüft mit SRP-Protokoll (Secure Remote Password), ob der Benutzer den iCloud-Sicherheitscode kennt, der Code selbst wird nicht an Apple gesendet. Alle Bestandteile des Clusters verifizieren unabhängig voneinander, dass der Benutzer die maximale Anzahl zulässiger Versuche, die zum Abrufen des Eintrags zulässig sind, nicht überschritten hat, wie im Folgenden beschrieben. Wenn dies mehrheitlich bestätigt wird, entpackt das Cluster den Escrow-Eintrag und sendet ihn an das Gerät des Benutzers.

Anschließend verwendet das Gerät den iCloud-Sicherheitscode, um den zufälligen Schlüssel zu entpacken, mit dem der Schlüsselbund des Benutzers verschlüsselt wurde. Mit diesem Schlüssel wird der Schlüsselbund aus dem iCloud-Schlüssel/Wert-Speicher entschlüsselt und auf dem Gerät wiederhergestellt. Es sind maximal 10 Versuche zulässig, um einen Escrow-Eintrag zu authentifizieren und abzurufen. Nach mehreren fehlgeschlagenen Versuchen wird der Eintrag gesperrt und der Benutzer muss sich für zusätzliche Versuche an den Apple Support werden. Nach dem 10. fehlgeschlagenen Versucht löscht der HSM-Cluster den Escrow-Eintrag unwiderruflich. Dies bietet Schutz vor Brute-Force-Angriffen auf den Eintrag, wobei die Daten für den Schlüsselbund geopfert werden.

Diese Richtlinien sind in der HSM-Firmware codiert. Die Zugangskarten, mit denen die Firmware geändert werden kann, wurden zerstört. Alle Versuche, die Firmware zu ändern oder auf den privaten Schlüssel zuzugreifen, führen dazu, dass der HSM-Cluster den privaten Schlüssel löscht. In diesem Fall erhalten die Eigentümer aller mit dem Cluster geschützten Schlüsselbunde eine Mitteilung, dass der Escrow-Eintrag gelöscht wurde. Sie können sich dann erneut registrieren.

Siri

Benutzer können Siri einfach ansprechen, um E-Mails zu senden, Termine zu planen, Anrufe zu starten und mehr. Siri verwendet Spracherkennung, Sprachausgabe sowie ein Client-Server-Modell zur Beantwortung verschiedenster Anfragen. Bei den von Siri unterstützten Aufgaben wurde Wert darauf gelegt, dass möglichst wenig persönliche Daten genutzt und diese vollständig geschützt werden.

Wird Siri aktiviert, erstellt das Gerät zufällige Kennungen für die Verwendung mit der Spracherkennung und den Siri Servern. Diese Kennungen werden nur in Siri zur Verbesserung des Dienstes verwendet. Wird Siri deaktiviert, erzeugt das Gerät eine neue zufällige Kennung, die bei der Reaktivierung von Siri verwendet wird.

Um die Funktionen von Siri nutzen zu können, sendet das Gerät bestimmte Benutzerinformation an den Server. Dazu gehören Informationen zu der Mediathek (Songtitel, Interpreten und Wiedergabelisten), die Namen der Erinnerungslisten und in den Kontakten definierte Namen und Beziehungen. Für die gesamte Kommunikation mit dem Server wird HTTPS verwendet.

Wenn Siri eine Sitzung startet, werden Vor- und Nachname des Benutzers (aus den Kontakten) zusammen mit dem ungefähren Standort an den Server gesendet. Dadurch kann Siri mit dem Namen und auf Fragen antworten, für die nur der ungefähre Standort, z. B. für das Wetter, benötigt wird.

Wird ein genauerer Standort benötigt, z. B. um Kinos in der Nähe zu finden, bittet der Server das Gerät um genauere Ortsdaten. Das ist ein Beispiel dafür, wie Informationen standardmäßig nur dann an den Server gesendet werden, wenn es zur Bearbeitung der Anfrage des Benutzers notwendig ist. In jedem Fall werden die Sitzungsdaten nach 10 Minuten Inaktivität gelöscht.

Wenn Siri auf der Apple Watch verwendet wird, erstellt die Uhr eine eigene, zufällige Kennung, wie oben beschrieben. Statt die Benutzerinformationen erneut zu senden, senden die Anfragen auch die Siri-Kennung an das gekoppelte iPhone, um eine Referenz auf diese Informationen bereitzustellen.

Eine Aufzeichnung der gesprochenen Wörter wird an den Spracherkennungsserver von Apple gesendet. Wenn nur etwas diktiert werden soll, wird der erkannte Text an das Gerät zurückgesendet. Ansonsten analysiert Siri den Text und kombiniert ihn gegebenenfalls mit Informationen über das mit dem Gerät verknüpfte Profil. Wenn die Anfrage z. B. „Sende meiner Mutter eine Nachricht“ lautet, werden die aus den Kontakten geladenen Beziehungen und Namen verwendet. Der Befehl für die identifizierte Aktion wird anschließend an das Gerät zurückgesendet und dort ausgeführt.

Viele Funktionen von Siri werden unter der Steuerung des Servers auf dem Gerät ausgeführt. Wenn der Benutzer z. B. Siri bittet eine erhaltene Nachricht vorzulesen, teilt der Server dem Gerät lediglich mit, dass es den Inhalt der ungelesenen Nachricht vorlesen soll. Inhalt und Absender der Nachricht werden nicht an den Server gesendet.

Die Sprachaufzeichnungen werden bis zu sechs Monate gespeichert, damit das Spracherkennungssystem sie verwenden kann, um den Benutzer besser zu verstehen. Nach sechs Monaten wird eine Kopie (ohne Kennung) gespeichert, die Apple bis zu 2 Jahre für die Verbesserung und Entwicklung nutzen kann. Außerdem können manche Aufnahmen, die sich auf Musik, Mannschaften oder Spieler, Unternehmen oder Sehenswürdigkeiten beziehen, ebenfalls zur Verbesserung von Siri gespeichert werden.

Siri kann außerdem mit der Sprachaktivierung im Freisprechmodus verwendet werden. Die Sprachkommandoerkennung wird lokal auf dem Gerät durchgeführt. In diesem Modus wird Siri nur aktiviert, wenn das eingehende Sprachmuster dem eingestellten Kommando genügend ähnlich ist. Wird das Kommando erkannt, wird das entsprechende Audiomaterial mit dem folgenden Siri-Befehl nach denselben Richtlinien wie für andere Sprachdaten aus Siri an den Apple Spracherkennungsserver zur weiteren Bearbeitung gesendet.

Continuity

Continuity (auch „Integration"), eine neue Funktion in iOS 8 und OS X Yosemite, nutzt Technologien wie iCloud, Bluetooth und WLAN, um es dem Benutzer zu ermöglichen, eine Aktivität von einem Gerät auf ein anderes zu übertragen, Telefonanrufe zu tätigen und zu empfangen, SMS zu senden und zu empfangen und die Internetverbindung über Mobilfunk gemeinsam zu nutzen.

Handoff

Mit Handoff kann der Benutzer automatisch alles, woran er auf einem Gerät arbeitet, an seine Macs oder iOS-Geräte in der Nähe übertragen. Mit Handoff kann der Benutzer das Gerät wechseln und sofort weiterarbeiten.

Meldet sich der Benutzer auf einem zweiten Handoff-fähigen Gerät bei iCloud an, wird zwischen den Geräten mittels Bluetooth Low Energy 4.0 eine Out-of-Band-Kopplung über den Apple-Dienst für Push-Benachrichtigungen (APNS) hergestellt. Die einzelnen Nachrichten werden ähnlich wie für iMessage verschlüsselt. Wenn die Geräte gekoppelt wurden, erstellt jedes einen symmetrischen 256-Bit AES-Schlüssel, der im Schlüsselbund des Geräts gespeichert wird. Dieser Schlüssel wird verwendet, um die Bluetooth Low Energy-Ankündigungen, die den anderen in iCloud gekoppelten Geräten die aktuelle Aktivität des Geräts mitteilen, zu verschlüsseln und authentifizieren, wobei zum Schutz vor Replay-Attacken AES-256 im GCM-Modus verwendet wird. Wenn ein Gerät zum ersten Mal eine Ankündigung von einem neuen Schlüssel erhält, stellt es über Bluetooth Low Energy eine Verbindung zum Absender her und führt einen Schlüsselaustausch für die Verschlüsselung von Ankündigungen durch. Diese Verbindung wird mit der Standardverschlüsselung für Bluetooth Low Energy 4.0 gesichert und die einzelnen Nachrichten werden zusätzlich verschlüsselt. Die Verschlüsselung ähnelt damit der für iMessage. Unter bestimmten Umständen, werden diese Nachrichten über den Apple-Dienst für Push-Benachrichtigungen anstelle von Bluetooth Low Energy versendet. Der Payload der Aktivität wird genau wie eine iMessage geschützt und übertragen.

Handoff zwischen nativen Apps und Websites

Handoff ermöglicht es nativen iOS-Apps, Webseiten auf Domains, deren Eigentümer der Entwickler der App ist, zu übernehmen. Die Benutzeraktivität in der nativen App kann auch in einem Internetbrowser übernommen werden.

Damit native Apps keine Websites übernehmen können, deren Eigentümer nicht der Entwickler ist, muss die App belegen, dass sie die Web-Domain, die sie übernehmen will, tatsächlich kontrolliert. Die Kontrolle über die Domain einer Website wird über die Methode für die gemeinsam genutzten Internetanmeldedaten überprüft. Weitere Informationen finden Sie unter „Zugriff auf gesicherte Passwörter in Safari" im Abschnitt „Verschlüsselung und Datensicherheit". Das System muss die Kontrolle der App über den Domain-Namen validieren, bevor sie Handoff für die Benutzeraktivität verwenden kann.

Ursprung für den Handoff einer Webseite kann jeder Browser sein, der die Handoff-APIs unterstützt. Wenn der Benutzer eine Webseite öffnet, kündigt das System den Domain-Namen der Webseite in den verschlüsselten Ankündigungsbytes für Handoff an. Nur die anderen Geräte des Benutzers können die Ankündigungsbytes entschlüsseln (wie oben beschrieben).

Auf dem Empfängergerät erkennt das System, dass eine installierte native App Handoff von dem angekündigten Domain-Namen annimmt, und zeigt das Symbol für die native App als Handoff-Option an. Wird sie gestartet, empfängt die App die vollständige URL und den Titel der Webseite. Es werden keine anderen Informationen vom Browser an die native App übertragen.

Demgegenüber kann eine native App eine Fallback-URL angeben, wenn auf dem Empfängergerät die native App nicht installiert ist. In diesem Fall zeigt das System den Standardbrowser des Benutzers als Handoff-Option an (wenn dieser Browser Handoff-APIs übernommen hat). Wird Handoff angefordert, öffnet der Browser die Fallback-URL, die die App gesendet hat. Die Fallback-URL ist nicht auf Domain-Namen, deren Eigentümer der Entwickler der nativen App ist, beschränkt.

Handoff für größere Datenmengen

Zusätzlich zu der Grundfunktion von Handoff können manche Apps auch APIs zum Senden größerer Datenmengen über die von Apple entwickelte Peer-To-Peer-WLAN-Technologie (ähnlich wie AirDrop) senden. Beispielsweise verwendet Mail diese APIs, um das Handoff eines E-Mail-Entwurfs, einschließlich großer Anhänge, zu ermöglichen.

Nutzt eine App diese Funktion, wird der Austausch zwischen den Geräten wie bei Handoff gestartet (siehe oben). Nach dem Empfang der ersten Nutzerdaten über Bluetooth Low Energy startet das Empfängergerät eine neue Verbindung über WLAN. Über diese verschlüsselte Verbindung (TLS) werden die iCloud-Identitätszertifikate ausgetauscht. Die Identität in den Zertifikaten wird mit der Identität des Benutzers abgeglichen. Die nachfolgenden Nutzerdaten werden über diese verschlüsselte Verbindung gesendet, bis die Übertragung abgeschlossen ist.

iPhone-Mobilanrufumleitung

Wenn sich Ihr Mac, iPad oder iPod im selben WLAN-Netzwerk wie Ihr iPhone befinden, können mit ihnen Telefonanrufe über die Mobilfunkverbindung des iPhone gestartet und empfangen werden. Für die Konfiguration müssen Ihre Geräte mit derselben Apple-ID bei iCloud und FaceTime angemeldet sein.

Bei einem eingehenden Anruf werden alle konfigurierten Geräte über den Apple-Dienst für Push-Benachrichtigungen (APNS) benachrichtigt, wobei für jede Benachrichtigung dieselbe End-To-End-Verschlüsselung wie für iMessage verwendet wird. Geräte im selben Netzwerk zeigen die Mitteilung für eingehende Anrufe an. Wird der Anruf entgegengenommen, werden die Audiodaten nahtlos über eine sichere Peer-To-Peer-Verbindung zwischen den beiden Geräten übertragen.

Ausgehende Anrufe werden ebenfalls über den Apple-Dienst für Push-Benachrichtigungen umgeleitet und die Audiodaten werden genauso über eine sichere Peer-To-Peer-Verbindung zwischen den Geräten übertragen.

Der Benutzer kann die Mobilanrufumleitung auf einem Gerät deaktivieren, indem er in den FaceTime-Einstellungen „iPhone-Mobilanrufe“ deaktiviert.

iPhone-Nachrichtenweiterleitung

Mit der Nachrichtenweiterleitung werden SMS vom iPhone automatisch auf die registrierten iPads, iPod touches und Macs des Benutzers übertragen. Alle Geräte müssen bei dem Dienst „iMessage“ mit derselben Apple-ID angemeldet sein. Ist die Nachrichtenweiterleitung aktiviert, wird die Registrierung für jedes Gerät bestätigt, indem ein vom iPhone erzeugter zufälliger sechsstelliger Code auf ihm eingegeben wird.

Sind die Geräte verknüpft, verschlüsselt das iPhone eingehende SMS und leitet sie an die anderen Geräte weiter, wobei die im Abschnitt „iMessage“ dieses Dokuments beschriebenen Methoden verwendet werden. Die Antworten werden mit derselben Methode an das iPhone zurückgesendet, das die Antwort dann über den Mobilfunkanbieter als SMS verschickt. Die Nachrichtenweiterleitung kann in den Nachrichteneinstellungen aktiviert oder deaktiviert werden.

Instant-Hotspot

iOS-Geräte, die Instant-Hotspot unterstützen, verwenden Bluetooth Low Energy, um Geräte zu erkennen, die beim selben iCloud-Account angemeldet sind, und um mit ihnen zu kommunizieren. Kompatible Mac-Computer mit OS X Yosemite verwenden dieselbe Technologie, um iOS-Geräte mit Instant-Hotspot zu erkennen und um mit ihnen zu kommunizieren.

Wenn der Benutzer auf dem iOS-Gerät die WLAN-Einstellungen eingibt, sendet das Gerät ein Bluetooth-Low-Energy-Signal mit einer Kennung, die alle an demselben iCloud-Account angemeldeten Geräte verwenden. Diese Kennung wird aus einer mit dem iCloud-Account verknüpften DSID (Destination Signaling Identifier) erzeugt und regelmäßig geändert. Wenn sich andere an demselben iCloud-Account angemeldeten Geräte in unmittelbarer Nähe befinden und den persönlichen Hotspot unterstützen, erkennen sie das Signal und signalisieren ihre Verfügbarkeit.

Wenn der Benutzer ein verfügbares Gerät als persönlichen Hotspot auswählt, wird eine Anfrage für die Aktivierung des persönlichen Hotspots an das Gerät gesendet. Die Anfrage wird über eine Verbindung gesendet, die die Bluetooth Low Energy-Standardverschlüsselung verwendet und die Anfrage selbst wird ähnlich wie eine iMessage verschlüsselt. Das Gerät sendet dann über dieselbe Bluetooth Low Energy-Verbindung mit derselben nachrichtenspezifischen Verschlüsselung die Verbindungsinformationen für den persönlichen Hotspot.

Spotlight-Vorschläge

Die Suche in Safari und Spotlight enthält jetzt Spotlight-Vorschläge, die Ergebnisse aus iTunes, dem App Store, dem Kinoprogramm, Orten in der Nähe und mehr vorschlägt.

Damit die Vorschläge interessanter für den Benutzer werden, enthalten Spotlight-Vorschläge Benutzerkontext und Feedback bei jeder an Apple gesendeten Suchanfrage.

Über den mit der Suchanfrage gesendeten Kontext erfährt Apple: a) den ungefähren Standort des Geräts, b) die Art des Geräts (z. B. Mac, iPhone, iPad oder iPod), c) die Client-App, also entweder Spotlight oder Safari, d) die Einstellungen für „Sprache & Region“, e) die drei zuletzt auf dem Gerät verwendeten Apps und f) eine anonyme Sitzungs-ID. Die gesamte Kommunikation mit dem Server wird mit HTTPS verschlüsselt.

Spotlight-Vorschläge senden zum Schutz der Privatsphäre nie den exakten Standort, sondern verschleiern den Standort zuvor auf dem Client. Der Umfang der Verschleierung ist abhängig von der ungefähren Bevölkerungsdichte am Standort des Geräts, d. h. auf dem Land wird sie stärker verschleiert als in der Innenstadt, wo sich mehr Benutzer befinden werden. Außerdem kann der Benutzer in den Systemeinstellungen festlegen, dass keine Ortsdaten an Apple gesendet werden sollen, indem er die Ortungsdienste für Spotlight-Vorschläge deaktiviert. Wenn die Ortungsdienste deaktiviert sind, kann Apple den ungefähren Standort des Clients aus der IP-Adresse des Clients ableiten.

Mit der anonymen Sitzungs-ID kann Apple Muster bei innerhalb von 15 Minuten gestellten Anfragen analysieren. Wenn ein Benutzer zum Beispiel häufig nach „Telefonnummer Café“ sucht, nachdem er nach „Café“ gesucht hat, kann Apple die Telefonnummer in Zukunft gleich in den Ergebnissen anzeigen. Anders als die meisten Suchmaschinen verwendet der Suchdienst von Apple keine permanente persönliche Kennung mit dem Suchanfragen einem Benutzer oder Gerät zugeordnet werden können, stattdessen nutzen Apple-Geräte eine temporäre anonyme Sitzungs-ID für maximal 15 Minuten, bevor diese ID gelöscht wird.

Informationen zu den drei zuletzt auf dem Gerät genutzten Apps werden für zusätzlichen Kontext in die Suchanfrage aufgenommen. Um die Privatsphäre der Benutzer zu schützen, werden nur Apps aufgenommen die in einer von Apple gepflegten Whitelist beliebter Apps aufgeführt sind und in den letzten 3 Stunden verwendet wurden.

Das an Apple gesendete Feedback enthält: a) die vergangene Zeit zwischen Aktionen des Benutzers, wie gedrückte Tasten und Auswahl eines Ergebnisses, b) ggf. der ausgewählte Spotlight-Vorschlag und c) Art des ausgewählten lokalen Ergebnisses (z. B. „Lesezeichen" oder „Kontakt"). Genau wie beim Suchkontext wird das Feedback nicht mit einem bestimmten Benutzer oder Gerät verknüpft.

Apple speichert die Protokolle für Spotlight-Vorschläge mit Anfragen, Kontext und Feedback bis zu 18 Monate lang. Die gekürzten Protokolle, die nur Anfrage, Land, Sprache, Datum (auf die Stunde genau) und Gerätetyp enthalten, werden bis zu 2 Jahre gespeichert. Mit den Anfrageprotokollen werden keine IP-Adressen gespeichert.

Unter bestimmten Umständen können Spotlight-Vorschläge Anfragen für häufige Wörter und Ausdrücke an einen qualifizierten Partner weiterleiten, um die Suchergebnisse des Partners erhalten und anzeigen zu können. Diese Anfragen werden vom qualifizierten Partner nicht gespeichert und die Partner erhalten kein Feedback für die Suche. Ebenso erhalten die Partner keine IP-Adressen. Die Kommunikation mit dem Partner wird mit HTTPS verschlüsselt. Apple gibt Stadt, Gerätetyp und Sprache des Clients als Suchkontext an den Partner weiter, abhängig davon welche Standorte, Gerätetypen und Sprachen wiederholt Suchanfragen an Apple senden.

Die Spotlight-Vorschläge können in den Einstellungen für Spotlight und/oder Safari deaktiviert werden. Wenn sie für Spotlight deaktiviert werden, sucht Spotlight nur noch lokal auf dem Gerät und sendet keine Informationen an Apple. Wenn sie für Safari deaktiviert werden, werden die Suchanfragen, Suchkontext und Feedback nicht an Apple übertragen.

Gerätesteuerungen

iOS unterstützt flexible Sicherheitsrichtlinien und Konfigurationen, die einfach umgesetzt und verwaltet werden können. Dadurch können Organisationen interne Informationen schützen und sicherstellen, dass Mitarbeiter die Vorgaben der Organisation einhalten, selbst wenn sie ihre eigenen Geräte verwenden, z. B. im Rahmen eines BYOD-Programms („Bring Your Own Device“).

Organisationen können Methoden wie Codesicherheit, Konfigurationsprofile, Fernlöschung und MDM-Lösungen anderer Anbieter verwenden, um den Gerätebestand zu verwalten und Firmendaten zu schützen, selbst wenn Mitarbeiter über ihre persönlichen iOS-Geräte auf diese Daten zugreifen.

Codesicherheit

Neben dem zuvor beschriebenen kryptografischen Schutz verhindern Gerätecodes den unbefugten Zugriff auf die Benutzeroberfläche des Geräts. Die iOS-Benutzeroberfläche setzt steigende zeitliche Verzögerungen durch, wenn ein ungültiger Code auf dem Sperrbildschirm eingegeben wurde. Benutzer können festlegen, dass das Gerät nach zehn fehlgeschlagenen Code-Eingaben automatisch gelöscht werden soll. Diese Einstellung ist auch als Verwaltungsrichtlinie über MDM und Exchange ActiveSync verfügbar und die maximal zulässige Anzahl von Fehleingaben kann zudem verringert werden.

Der Gerätecode des Benutzers ist standardmäßig eine vierstellige PIN. Der Benutzer kann unter „Einstellungen“ > „Allgemein“ > „Code“ > „Längerer Code“ einen längeren, alphanumerischen Code festlegen. Längere und komplexere Codes sind schwerer zu erraten oder anzugreifen und werden für Unternehmen empfohlen.

Administratoren können komplexe Codes und andere Richtlinien mit MDM oder Exchange ActiveSync durchsetzen oder von Benutzern verlangen, Konfigurationsprofile manuell zu installieren. Es gibt die folgenden Richtlinien für Codes:

- Einfachen Wert erlauben
- Alphanumerische Werte erforderlich
- Mindestlänge für Codes
- Mindestanzahl von komplexen Zeichen
- Maximale Gültigkeitsdauer für Codes
- Codeverlauf
- Timeout für die automatische Sperre
- Maximale Zeitgrenze für die Gerätesperre
- Maximale Anzahl Fehlversuche
- Touch ID erlauben

Nähere Informationen zu den einzelnen Richtlinien finden Sie in der Dokumentation „Configuration Profile Key Reference“ unter https://developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/.

iOS-Kopplungsmodell

iOS verwendet ein Kopplungsmodell, um den Zugriff auf ein Gerät von einem Host-Computer zu steuern. Die Kopplung stellt über den Austausch der öffentlichen Schlüssel eine vertrauenswürdige Verbindung zwischen dem Gerät und dem verbundenen Host her. iOS verwendet diesen Vertrauensbeweis, um zusätzliche Funktionen mit dem verbundenen Host, wie Synchronisation der Daten, zu aktivieren.

Für die Kopplung muss der Benutzer das Gerät entsperren und die Anfrage des Hosts annehmen. Daraufhin tauschen Host und Gerät öffentliche 1024-Bit RSA-Schlüssel aus und sichern sie. Der Host erhält einen 256-Bit Schlüssel mit dem er den auf dem Gerät gespeicherten Escrow-Keybag entsperren kann (siehe „Escrow-Keybags" im Abschnitt „Keybags"). Die ausgetauschten Schlüssel werden zum Starten einer verschlüsselten SSL-Sitzung verwendet, die das Gerät benötigt, bevor es geschützte Daten an den Host senden oder einen Dienst starten kann (iTunes-Synchronisierung, Dateiübertragung, Xcode-Entwicklung, usw.). Das Gerät benötigt eine Verbindung von dem Host über WLAN, um diese verschlüsselte Sitzung für die gesamte Kommunikation zu verwenden, es muss also zuvor über USB gekoppelt werden. Die Kopplung ermöglicht auch mehrere Diagnosefähigkeiten. Weitere Informationen finden Sie unter https://support.apple.com/kb/HT6331?viewlocale=de_DE.

In iOS 8 sind bestimmte Dienste, z. B. com.apple.pcapd, nur über USB möglich. Außerdem benötigt der Dienst com.apple.file_relay in iOS 8 ein von Apple signiertes Konfigurationsprofil, um installiert werden zu können.

Der Benutzer kann die Liste vertrauenswürdiger Hosts löschen, indem er die Option „Netzwerkeinstellungen zurücksetzen" oder „Standort & Datenschutz zurücksetzen" verwendet. Weitere Information finden Sie unter https://support.apple.com/kb/HT5868?viewlocale=de_DE.

Erzwingen von Konfigurationen

Konfigurationsprofile sind XML-Dateien, mit deren Hilfe ein Administrator Konfigurationsdaten auf iOS-Geräte übertragen kann. Einstellungen, die von einem installierten Konfigurationsprofil festgelegt wurden, können vom Benutzer nicht geändert werden. Wird ein Konfigurationsprofil gelöscht, werden auch alle damit festgelegten Einstellungen zurückgesetzt. Administratoren können so Einstellungen durchsetzen, indem sie Richtlinien mit dem Zugriff verknüpfen. Beispielsweise kann ein Konfigurationsprofil für die E-Mail-Konfiguration auch verwendet werden, um eine Richtlinie für den Gerätecode festzulegen. Der Benutzer kann nur auf den E-Mail-Account zugreifen, wenn sein Code den Anforderungen des Administrators entspricht.

iOS-Konfigurationsprofile können verschiedene Einstellungen festlegen, darunter:

- Coderichtlinien
- Einschränkung der Funktionen des Geräts (z. B. Deaktivieren der Kamera)
- WLAN-Einstellungen
- VPN-Einstellungen
- Mail-Server-Einstellungen
- Exchange-Einstellungen
- LDAP-Verzeichnisdiensteinstellungen
- CalDAV-Kalenderdiensteinstellungen
- Webclips
- Anmeldedaten und Schlüssel
- Erweiterte Mobilfunknetzeinstellungen

Konfigurationsprofile können signiert und verschlüsselt werden, um ihren Ursprung und damit ihre Integrität zu verifizieren und den Inhalt zu schützen. Konfigurationsprofile werden mittels CMS (RFC 3852) verschlüsselt und unterstützen 3DES und AES-128.

Konfigurationsprofile können auch fest an ein Gerät gebunden werden, sodass sie nicht mehr oder nur mit einem Code entfernt werden können. Da bei Unternehmenslösungen viele Benutzer ihre iOS-Geräte selbst besitzen, können Konfigurationsprofile, die ein Gerät fest mit einem MDM-Server verbinden, entfernt werden, dabei werden aber alle verwalteten Konfigurationsinformationen, Daten und Apps entfernt.

Benutzer können Konfigurationsprofile mit Apple Configurator direkt auf ihren Geräten installieren oder sie in Safari oder aus einer Mail laden oder von einem MDM-Server zugesendet bekommen

Mobile Device Management (MDM)

Die MDM-Unterstützung in iOS ermöglicht Unternehmen die sichere Konfiguration und Verwaltung skalierter Implementierungen von iPhone und iPad im Unternehmen. MDM-Funktionen basieren auf vorhandenen iOS-Technologien wie Konfigurationsprofile, kabellose Registrierung und dem Apple-Dienst für Push-Benachrichtigungen. Zum Beispiel wird APNS verwendet, um den Ruhezustand zu beenden, damit das Gerät direkt mit seinem MDM-Server über eine sichere Verbindung kommunizieren kann. Über APNS werden keine vertraulichen oder betriebsinternen Informationen übertragen.

MDM eröffnet IT-Abteilungen die Möglichkeit, iOS-Geräte sicher in einer Unternehmensumgebung zu registrieren, drahtlos Einstellungen zu konfigurieren und zu aktualisieren, die Einhaltung von Unternehmensrichtlinien zu überwachen und die Geräte sogar per Fernzugriff zu löschen oder zu sperren. Weitere Informationen zu MDM finden Sie unter https://www.apple.com/iphone/business/it/management.html.

Programm zur Geräteregistrierung

Das Programm zur Geräteregistrierung (DEP) bietet eine schnelle, effiziente Methode, um iOS-Geräte bereitzustellen, die eine Organisation direkt bei Apple oder über teilnehmende autorisierte Apple-Händler und Anbieter gekauft hat. Geräte können automatisch in MDM registriert werden, ohne sie physisch berühren oder vorbereiten zu müssen, bevor die Benutzer sie erhalten. Darüber hinaus kann der Konfigurationsprozess für Benutzer weiter vereinfacht werden, indem bestimmte Schritte im Systemassistenten entfernt werden, sodass die Benutzer schnell mit ihrer Arbeit beginnen können. Es lässt sich auch steuern, ob der Benutzer das MDM Profil vom Gerät löschen kann. Sie können beispielsweise die Geräte direkt bei Apple bestellen, alle Verwaltungseinstellungen konfigurieren und die Geräte direkt an die Postanschrift des Benutzers senden lassen. Nach dem Entpacken und Aktivieren des Gerät wird es im MDM der Organisation registriert und alle verwalteten Einstellungen, Apps und Bücher stehen für den Benutzer bereit.

Der Vorgang ist einfach: Nach der Registrierung im Programm melden Administratoren sich an der Programm-Website an, verknüpfen das Programm mit ihrem MDM-Server und weisen sich die über Apple erworbenen iOS-Geräte zu. Die Geräte können dann den Benutzern über MDM zugewiesen werden. Sobald die Zuweisung zu einem Benutzer erfolgt ist, werden in MDM festgelegte Konfigurationen, Einschränkungen und Steuerungen automatisch installiert. Weitere Informationen finden Sie unter http://www.apple.com/de/business/programs/#dep/.

Hinweis: Das Programm zur Geräteregistrierung ist nicht in allen Ländern und Regionen verfügbar.

Apple Configurator

Zusammen mit MDM erleichtert Apple Configurator für OS X die Bereitstellung von iOS-Geräten. Apple Configurator kann verwendet werden, um schnell große Mengen an Geräten mit Einstellungen, Apps und Daten zu konfigurieren. Geräte, die zuerst mit Apple Configurator konfiguriert wurden, können „betreut" werden, was die Installation zusätzlicher Einstellungen und Einschränkungen ermöglicht. Sobald ein Gerät durch Apple Configurator betreut wird, können alle verfügbaren Einstellungen und Einschränkungen auch drahtlos über MDM installiert werden.

Weitere Informationen zum Konfigurieren und Verwalten von Geräten mit Apple Configurator und MDM finden Sie in der „iOS-Implementierung: Referenz" unter https://help.apple.com/deployment/ios.

Geräteeinschränkungen

Administratoren können Gerätefunktionen einschränken, indem sie ein Konfigurationsprofil installieren. Es gibt die folgenden Einschränkungen:

- Installation von Apps erlauben
- Verwendung der Kamera erlauben
- FaceTime erlauben
- Bildschirmfotos erlauben
- Sprachwahl erlauben
- Automatische Synchronisierung beim Roaming erlauben
- In-App-Einkäufe erlauben
- Synchronisierung aktueller E-Mails erlauben
- Store-Passwort für alle Einkäufe verlangen
- Mehrspielermodus erlauben
- Hinzufügen von Game Center-Freunden erlauben
- Siri erlauben
- Siri bei gesperrtem Gerät erlauben
- Verwendung von YouTube erlauben
- Passbook-Mitteilungen bei gesperrtem Gerät erlauben
- Verwendung von iTunes Store erlauben
- Anstößige Medien erlauben
- Erotische Inhalte aus iBooks Store erlauben
- Dokumente aus verwalteten Quellen an nicht verwaltete Ziele erlauben
- Dokumente aus nicht verwalteten Quellen an verwaltete Ziele erlauben
- iCloud Schlüsselbund erlauben
- Drahtlose Aktualisierung der Datenbank vertrauenswürdiger Zertifikate erlauben
- Anzeige von Mitteilungen auf dem Sperrbildschirm erlauben
- Verwendung von Passwörtern bei der Kopplung über eine AirPlay-Verbindung erzwingen
- Anzeigen nutzergenerierter Inhalte aus dem Internet in Spotlight erlauben
- Spotlight-Vorschläge in Safari erlauben
- Spotlight-Vorschläge in Spotlight erlauben
- Handoff erlauben
- Sicherung firmenweiter Bücher erlauben
- Geräteübergreifendes Synchronisieren von Notizen und Lesezeichen in firmenweiten Büchern erlauben

- Altersfreigaben für Filme aktivieren
- Altersfreigaben für TV-Sendungen aktivieren
- Altersfreigaben für Apps aktivieren
- Verwendung von Safari erlauben
- „Automatisch ausfüllen“ in Safari erlauben
- Warnung für betrügerische Websites erzwingen
- JavaScript erlauben
- Ad-Tracking in Safari einschränken
- Pop-Ups unterdrücken
- Cookies akzeptieren
- iCloud-Backup erlauben
- Synchronisieren von Dokumenten und Schlüsseln über iCloud erlauben
- iCloud-Fotomediathek erlauben
- iCloud-Fotofreigabe erlauben
- Fotostreams erlauben
- Freigegebene Fotostreams erlauben
- Senden von Diagnosedaten an Apple erlauben
- Annehmen nicht vertrauenswürdiger TLS-Zertifikate erlauben
- Verschlüsselte Backups erzwingen
- Medien nach Altersfreigabe einschränken
- Touch ID erlauben
- Zugriff auf das Kontrollzentrum aus dem Sperrbildschirm erlauben
- Anzeige der Tagesansicht auf dem Sperrbildschirm erlauben
- Apple Watch-Handgelenkerkennung anfordern

Einschränkungen nur für betreute Geräte

- iMessage erlauben
- Game Center erlauben
- iBooks Store erlauben
- Entfernen von Apps erlauben
- Siri Filter für anstößige Sprache aktivieren
- Manuelle Installation von Konfigurationsprofilen erlauben
- Globaler Netzwerk-Proxy für HTTP
- Kopplung mit Computern zur Synchronisierung von Inhalten erlauben
- AirPlay Verbindungen mit Whitelist und optionalen Verbindungscodes einschränken
- AirDrop erlauben
- Podcasts erlauben
- Änderung von „Meine Freunde suchen“ erlauben
- Autonomen Einzel-App-Modus für bestimmte verwaltete Apps erlauben
- Accountänderungen erlauben
- Änderung der Mobilfunkdaten erlauben
- Host-Kopplung erlauben (iTunes)
- Aktivierungssperre erlauben
- Löschen aller Inhalte und Einstellungen verhindern
- Aktivieren von Einschränkungen verhindern

- Filter für Inhalte anderer Anbieter
- Einzel-App-Modus
- Dauerhaft aktivierte VPN-Verbindung

Fernlöschung

iOS-Geräte können von einem Administrator oder Benutzer per Fernzugriff gelöscht werden. Die sofortige Fernlöschung wird dadurch aktiviert, dass der Speicherblockschlüssel aus dem Effaceable Storage gelöscht wird, sodass die Daten nicht mehr gelesen werden können. Die Fernlöschung kann über MDM, Exchange oder iCloud aktiviert werden.

Wenn MDM oder iCloud die Fernlöschung auslösen, sendet das Gerät eine Bestätigung und führt den Löschvorgang durch. Bei der Fernlöschung über Exchange meldet sich das Gerät am Exchange Server an, bevor es den Löschvorgang startet.

Benutzer können das Gerät, wenn sie direkten Zugriff darauf haben, auch in den Einstellungen löschen. Außerdem kann das Gerät, wie bereits erwähnt, so eingestellt werden, dass es sich nach einer bestimmten Anzahl fehlgeschlagener Code-Eingaben automatisch löscht.

Mein iPhone suchen und Aktivierungssperre

Es ist wichtig, Geräte bei Verlust oder Diebstahl zu deaktivieren und die darauf befindlichen Daten zu löschen. Ab iOS 7 kann ein Gerät nicht ohne Eingabe der Apple-ID Anmeldedaten des Eigentümers erneut aktiviert werden, solange die Option „Mein iPhone suchen“ aktiviert ist. Es ist empfehlenswert, Geräte zu betreuen, die sich im Eigentum der Organisation befinden, oder festzulegen, dass Benutzer diese Option deaktivieren müssen. Andernfalls würde die Organisation daran gehindert, das Gerät einem anderen Benutzer zuzuweisen.

In iOS 7.1 oder neuer kann eine kompatible MDM-Lösung die Aktivierungssperre auf betreuten Geräten einschalten, wenn „Mein iPhone suchen“ von einem Benutzer aktiviert wird. MDM-Administratoren können die Aktivierungssperre für „Mein iPhone suchen“ verwalten, indem sie die Geräte mit Apple Configurator oder über das Programm zur Geräteregistrierung betreuen. Die MDM-Lösung kann bei eingeschalteter „Aktivierungssperre“ einen Umgehungscode speichern und diesen später verwenden, um die Aktivierungssperre automatisch aufzuheben, wenn das Gerät gelöscht und einem neuen Benutzer bereitgestellt werden soll. Weitere Informationen finden Sie in der Dokumentation Ihrer MDM-Lösung.

Achtung: Standardmäßig ist bei betreuten Geräten die Aktivierungssperre immer ausgeschaltet, selbst wenn der Benutzer „Mein iPhone suchen“ auswählt. Ein MDM-Server kann jedoch den Umgehungscode abrufen und die Aktivierungssperre auf dem Gerät erlauben. Ist „Mein iPhone suchen“ ausgewählt, wenn der MDM-Server die Aktivierungssperre aktiviert, bleibt die Sperre eingeschaltet. Ist „Mein iPhone suchen“ ausgeschaltet, wenn der MDM-Server die Aktivierungssperre aktiviert, wird die Sperre eingeschaltet, wenn der Benutzer das nächste Mal „Mein iPhone suchen“ auswählt.

Datenschutzeinstellungen

Apple respektiert die Privatsphäre seiner Kunden und hat diverse Einstellungen und Optionen integriert, mit dem Benutzer von iOS-Geräten selbst bestimmen können, wie und wann Apps welche Informationen verwenden können.

Ortungsdienste

Ortungsdienste verwenden GPS, Bluetooth, öffentliche WLAN-Hotspots und Mobilfunkmasten, um den ungefähren Standort des Benutzers zu bestimmen. Die Ortungsdienste können mit einem einzigen Schalter in den Einstellungen oder für jede App einzeln deaktiviert werden. Apps können anfragen, Ortsdaten bei geöffneter App oder immer verwenden zu dürfen. Der Benutzer kann diese Anfrage ablehnen und kann seine Wahl später in den Einstellungen ändern. In den Einstellungen kann der Zugriff generell deaktiviert, nur bei geöffneter App oder immer aktiviert werden, abhängig von der angefragten Verwendung. Außerdem wird der Benutzer daran erinnert, wenn eine App, die immer Zugriff hat, Ortungsdienste im Hintergrundmodus verwenden möchte, und dass er dieses ändern kann.

Darüber hinaus hat der Benutzer präzise Einstellungsmöglichkeiten für die Verwendung der Ortsdaten durch die Systemdienste. Unter anderem kann der Benutzer einstellen, dass die Ortsdaten nicht in die Diagnose- und Nutzungsdaten aufgenommen werden, die Apple verwendet, um iOS zu verbessern, und ortsbasierte Siri-Informationen, ortsbasierter Kontext für Spotlight-Vorschläge, lokale Verkehrsbedingungen zum Berechnen der Wegzeit und häufig besuchte Orte deaktiviert werden.

Zugriff auf persönliche Daten

iOS kann verhindern, dass Apps ohne Zustimmung auf die persönlichen Daten des Benutzers zugreifen können. Zusätzlich sieht der Benutzer in den Einstellungen, welche Apps Zugriff auf welche Informationen haben und kann diese Einstellung ändern. Dazu gehören Zugriff auf:

- Kontakte
- Kalender
- Erinnerungen
- Fotos
- Aktivitätsdaten (iPhone 5s oder neuer)
- Accounts in sozialen Netzwerken wie Twitter oder Facebook
- Mikrofon
- Kamera
- HomeKit
- HealthKit
- Bluetooth-Freigabe

Wenn sich der Benutzer bei iCloud anmeldet, haben Apps standardmäßig Zugriff auf iCloud Drive. Der Benutzer kann den iCloud-Zugriff für einzelne Apps in den Einstellungen ändern. Darüber hinaus bietet iOS Einschränkungen, welche den Datenaustausch zwischen vom MDM und dem Benutzer installierten Apps und Accounts verhindern.

Datenschutzrichtlinie

Die Datenschutzrichtlinie von Apple finden Sie online unter https://www.apple.com/de/legal/privacy.

Fazit

Der Sicherheit verpflichtet

Mit führenden Technologien für Datenschutz und Sicherheit, die dafür entwickelt wurden, persönliche Daten zu schützen, und umfassenden Methoden zum Schutz von Daten in Unternehmensumgebungen fördert Apple den Schutz seiner Kunden besonders stark.

Die Sicherheit ist integraler Bestandteil von iOS. Von der Plattform über das Netzwerk bis zu den Apps findet jedes Unternehmen alles auf der iOS-Plattform, was es braucht. Dank dieser Kombination bietet iOS branchenführende Sicherheit, ohne dass die Benutzerfreundlichkeit dadurch beeinträchtigt wird.

Apple verwendet eine konsistente, fest integrierte Sicherheitsinfrastruktur für iOS und das iOS-Apps-Ökosystem Mit der hardwarebasierten Speicherverschlüsselung kann ein verlorenes Gerät per Fernzugriff gelöscht werden und Benutzer können alle persönlichen und Unternehmensdaten vollständig von einem Gerät, das verkauft oder übertragen werden soll, löschen. Die Diagnosedaten werden ebenfalls anonym gesammelt.

Apple legt bei der Entwicklung seiner Apps für iOS besonderes Augenmerk auf die Sicherheit. Safari bietet privates Surfen und unterstützt das Online Certificate Status Protocol (OCSP), EV-Zertifikate und Verifizierungswarnungen für Zertifikate. Mail nutzt Zertifikate für authentifizierte und verschlüsselte E-Mails mit S/MIME-Unterstützung, die ab iOS 8 nachrichtenspezifische S/MIME-Verschlüsselung bietet, sodass der Benutzer standardmäßig E-Mails immer signieren und verschlüsseln kann oder gezielt auswählen kann, wie einzelne Nachrichten geschützt werden sollen. iMessage und FaceTime bieten außerdem Client-To-Client-Verschlüsselung.

Bei Apps von Drittanbietern wird über eine Kombination aus obligatorischer Codesignierung, Sandbox und Berechtigungen dafür gesorgt, dass der Benutzer, anders als bei anderen Plattformen, vollständig vor Viren, Malware und anderen Gefahren geschützt ist. Das Einreichverfahren für den App Store schützt Benutzer zusätzlich vor diesen Gefahren, da jede iOS App vor der Aufnahme sorgfältig geprüft wird.

Um die umfassenden Sicherheitsfunktionen optimal nutzen können, sollten Unternehmen ihre Richtlinien für IT und Sicherheit überprüfen, damit alle Ebenen der Sicherheitstechnologien dieser Plattform zum Einsatz kommen.

Apple verfügt über ein eigenes Sicherheitsteam für alle Apple-Produkte. Dieses Team führt Sicherheitsprüfungen für Produkte in der Entwicklung sowie für auf dem Markt befindliche Produkte durch. Das Apple-Team stellt auch Sicherheitstools und Schulungen bereit und überwacht aktiv Berichte zu neuen Sicherheitslücken und Bedrohungen. Apple ist Mitglied bei FIRST (Forum of Incident Response und Security Teams). Weitere Informationen, wie Problemberichte an Apple gesendet und Sicherheitsmitteilungen abonniert werden können, finden Sie unter: apple.com/de/support/security.

Glossar

Apple Push-Benachrichtigungsdienst (APNS	Ein globaler Dienst von Apple, der Push-Benachrichtigungen an iOS-Geräte sendet.
Bereitstellungsprofil	Von Apple signierte plist, die Entitäten und Berechtigungen enthält, mit denen Apps auf einem iOS-Gerät installiert und geprüft werden können. Ein Entwicklerbereitstellungsprofil führt alle Geräte auf, die der Entwickler für die Ad-Hoc-Verteilung ausgewählt hat und ein Verteilungsbereitstellungsprofil enthält die App-ID für firmeninterne Apps.
Boot-ROM	Der erste Code, den der Prozessor des Geräts am Beginn des Startvorgangs ausführt. Als integraler Bestandteil des Prozessors kann er weder von Apple noch von einem Angreifer modifiziert werden.
Datensicherheit	Schutzmechanismen für Dateien und Schlüsselbundobjekten in iOS. Kann sich auch auf APIs beziehen, die Apps zum Schutz von Dateien und Schlüsselbundobjekten verwenden.
Dateisystemschlüssel	Der Schlüssel mit dem die Metadaten jeder Datei, einschließlich des Klassenschlüssels, verschlüsselt werden. Wird im Effaceable Storage gespeichert und ermöglicht eine schnelle Löschung und dient weniger der Vertraulichkeit.
DFU-Modus (Device Firmware Upgrade)	Modus, bei dem der Boot-ROM des Geräts darauf wartet, über USB wiederhergestellt zu werden. Der Bildschirm bleibt im DFU-Modus schwarz, bis eine Verbindung zu einem Computer mit iTunes hergestellt wird. Daraufhin wird die folgende Eingabeaufforderung angezeigt: „iTunes hat ein iPad im Wartungsmodus erkannt. Sie müssen dieses iPad wiederherstellen, bevor es mit iTunes verwendet werden kann."
ECID	Eine eindeutige 64-Bit-Kennung für jeden Prozessor eines iOS-Geräts. Wird für die Personalisierung verwendet und ist nicht geheim.
Effaceable Storage (Auslöschbarer Speicher)	Ein bestimmter Bereich des NAND-Speichers, in dem kryptografische Schlüssel gespeichert werden und der direkt abgerufen und sicher gelöscht werden kann. Zwar bietet er keinen Schutz, wenn ein Angreifer direkt auf das Gerät zugreifen kann, die Schlüssel im Effaceable Storage können aber als Teil einer Schlüsselhierarchie verwendet werden und so eine schnelle Löschung und Folgenlosigkeit (Forward Secrecy) ermöglichen.
Gerätegruppen-ID (GID)	Ähnlich der UID, aber für alle Prozessoren einer Klasse identisch.
Hardwaresicherheitsmodule (HSM)	Spezieller manipulationssicherer Computer, der digitale Schlüssel schützt und verwaltet.
iBoot	Code der im Rahmen des sicheren Startvorgangs von LLB geladen wird und selbst XNU lädt.
Identity Service (IDS)	Apple-Verzeichnis der öffentlichen iMessage-Schlüssel, APNS-Adressen, Telefonnummern und E-Mail-Adressen, die zur Überprüfung von Schlüsseln und Geräteadressen verwendet werden.
Integrierter Schaltkreis (IC)	Auch als Mikrochip bezeichnet.
Joint Test Action Group (JTAG)	Standardwerkzeug für Hardwaredebugging, das von Programmierern und Schaltkreisentwicklern verwendet wird.

Keybag	Eine Datenstruktur, die zum Speichern von Klassenschlüsselsammlungen verwendet wird. Alle Keybag-Arten (System, Backup, Escrow oder iCloud-Backup) besitzen dasselbe Format: • Ein Header mit: – Version (in iOS 5 auf 3 eingestellt) – Typ (System, Backup, Escrow oder iCloud-Backup) – Keybag UUID – HMAC bei signierten Keybags – Die für die Klassenschlüssel verwendete Verpackungsmethode: Verknüpfung mit der UID oder PBKDF2, zusammen mit Salt und Iterationen. • Eine Liste der Klassenschlüssel: – Schlüssel UUID – Klasse (Datensicherheitsklasse der Datei bzw. des Schlüsselbundes) – Verpackungsart (nur von UID abgeleiteter Schlüssel, von UID abgeleiteter Schlüssel und von Code abgeleiteter Schlüssel) – Verpackter Klassenschlüssel – Öffentlicher Schlüssel für asymmetrische Klassen
Key-Wrapping	Verschlüsseln eines Schlüssels mit einem anderen Schlüssel. iOS verwendet die Verpackungsmethode NIST AES Key Wrapping in Übereinstimmung mit RFC 3394.
Low-Level Bootloader (LLB)	Code der im Rahmen des sicheren Startvorgangs vom Boot-ROM abgerufen und selbst iBoot lädt.
Pro Datei erzeugter Schlüssel	Der AES 256-Bit-Schlüssel, mit dem eine Datei im Dateisystem verschlüsselt wird. Der pro Datei erzeugte Schlüssel wird mit einem Klassenschlüssel verpackt und in den Metadaten der Datei gespeichert.
Schlüsselbund	Infrastruktur und APIs, die iOS und Apps anderer Anbieter nutzen, um Passwörter, Schlüssel und andere vertrauliche Anmeldedaten zu sichern und abzurufen.
Speicherverwürfelung (Address Space Layout Randomization, ASLR)	Von iOS verwendete Technik, die das erfolgreiche Ausnutzen von Softwarebugs erschwert. Da Speicheradressen und Segmentierung nicht vorhergesagt werden können, kann Exploit-Code diese Werte nicht hart codieren. Ab iOS 5 ist die Position aller Apps und Bibliotheken des Systems zufällig, ebenso die von Apps anderer Anbieter, die als positionsunabhängige ausführbare Dateien kompiliert wurden.
System on a Chip (SoC)	Integrierter Schaltkreis (IC) der mehrere Komponenten in einem einzigen Chip zusammenfasst. Die Secure Enclave ist ein SoC im CPU A7 (oder neuer) von Apple.
Uniform Resource Identifier (URI)	Eine Zeichenkette, mit der webbasierte Ressourcen identifiziert werden können.
Unique ID (UID)	Ein AES 256-Bit-Schlüssel der bei der Fertigung in den Prozessor eingebrannt wird. Er kann weder von der Firmware noch von Software gelesen werden und wird nur von der AES Engine der Prozessorhardware verwendet. Um den eigentlichen Schlüssel zu erhalten müsste ein potentieller Angreifer einen extrem komplexen und kostenintensiven physischen Angriff auf das Prozessorsilizium ausführen. Die UID ist mit keiner anderen Kennung des Geräts, einschließlich der UDID, verbunden.
Verknüpfung	Verfahren, mit dem der Code eines Benutzers in einen kryptografischen Schlüssel konvertiert wird, der mit der UID des Geräts zusätzlich gesichert wird. Dadurch müssen Brute-Force-Angriffe auf dem jeweiligen Gerät durchgeführt werden, sodass sie nicht von mehreren Geräten gleichzeitig ausgeführt werden können. Für die Verknüpfung wird der Algorithmus PBKDF2 benutzt, der AES mit der UID des Geräts als pseudozufällige Funktion (PRF) für jede Iteration verwendet.
XNU	Der Kernel im Zentrum der Betriebssysteme iOS und OS X. Er wird als vertrauenswürdig eingestuft und setzt Sicherheitsmaßnahmen wie Codesignierung, Sandbox, Überprüfen von Berechtigungen und ASLR durch.