



iOS-Sicherheit

iOS 11

Januar 2018

Inhalt

Seite 4 Einleitung

Seite 5 Systemsicherheit

- Sicherer Startvorgang
- Autorisierung der Systemsoftware
- Secure Enclave
- Touch ID
- Face ID

Seite 12 Verschlüsselung und Datensicherheit

- Funktionen für die Hardwaresicherheit
- Sicherheit von Dateidaten
- Gerätecodes
- Datensicherheitsklassen
- Sicherheit von Schlüsselbunddaten
- Zugriff auf gesicherte Passwörter in Safari
- Keybags
- Sicherheitszertifizierungen und -programme

Seite 23 Sicherheit in Apps

- App-Codesignierung
- Sicherheit von Laufzeitprozessen
- Erweiterungen
- App-Gruppen
- Sicherheit von Daten in Apps
- Zubehör
- HomeKit
- SiriKit
- HealthKit
- ReplayKit
- Geschützte Notizen
- Geteilte Notizen
- Apple Watch

Seite 36 Netzwerksicherheit

- TLS
- VPN
- WLAN
- Bluetooth
- Single-Sign-On
- Sicherheit bei AirDrop
- WLAN-Passwortfreigabe

Seite 41 Apple Pay

- Apple Pay-Komponenten
- So nutzt Apple Pay das Secure Element
- So nutzt Apple Pay den NFC-Controller
- Bereitstellung von Kredit-, Debit- und Prepaid-Karten
- Autorisierung von Zahlungen

Transaktionsspezifischer dynamischer Sicherheitscode
Kontaktlose Zahlungen mit Apple Pay
Mit Apple Pay in Apps zahlen
Mit Apple Pay im Web oder mit Handoff zahlen
Kundenkarten
Apple Pay Cash
Suica-Karten
Karten sperren, entfernen und löschen

Seite 52 Internetdienste

Apple-ID
iMessage
FaceTime
iCloud
iCloud-Schlüsselbund
Siri
Integration
Safari-Vorschläge, Siri-Vorschläge in „Suchen“, „Nachschlagen“,
#images, App „News“ und News-Widget in Ländern ohne die
App „News“

Seite 68 Gerätesteuerungen

Codesicherheit
iOS-Kopplungsmodell
Erzwingen von Konfigurationen
Mobile Device Management (MDM)
Geteiltes iPad
Apple School Manager
Geräteregistrierung
Apple Configurator 2
Betreuung
Einschränkungen
Fernlöschen
Modus „Verloren“
Aktivierungssperre

Seite 75 Datenschutzeinstellungen

Ortungsdienste
Zugriff auf persönliche Daten
Datenschutzrichtlinie

Page 77 Apple Security Bounty

Seite 78 Fazit

Der Sicherheit verpflichtet

Seite 79 Glossar

Seite 81 Dokumentrevisionsen

Einleitung

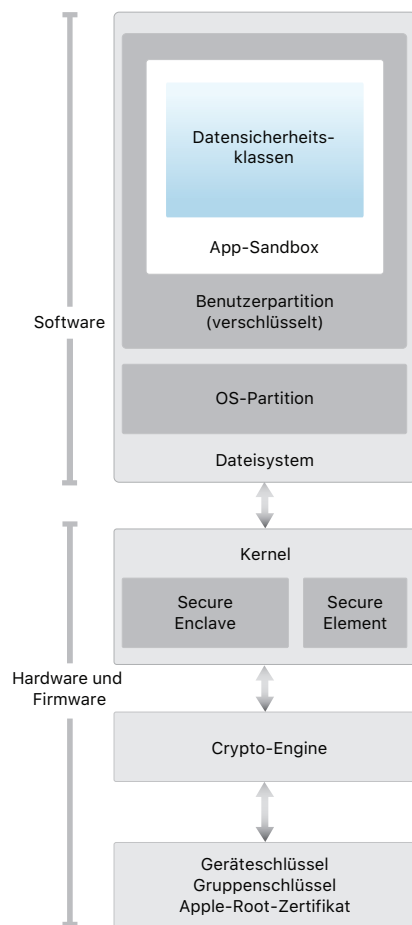


Diagramm der Sicherheitsarchitektur von iOS mit einem grafischen Überblick über die verschiedenen Technologien, auf die in diesem Dokument näher eingegangen wird

Apple hat bei der Entwicklung der iOS-Plattform die Sicherheit in den Mittelpunkt gestellt. Dank jahrzehntelanger Erfahrung konnten wir eine völlig neue Architektur entwickeln, um die beste mobile Plattform aller Zeiten 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 wie Face ID erhöhen die Benutzerfreundlichkeit, da mit ihnen das Gerät einfacher und intuitiver geschützt werden kann.

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

Das 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, falls das Gerät verloren oder gestohlen wird oder unbefugte Personen versuchen, es zu verwenden oder zu modifizieren
- **Sicherheit in Apps:** Systeme, die dafür sorgen, dass Apps sicher und ohne Gefährdung der Plattformintegrität ausgeführt werden können
- **Netzwerksicherheit:** Netzwerkprotokolle nach Industriestandards, die eine sichere Authentifizierung und die Verschlüsselung von Daten bei der Übertragung ermöglichen
- **Apple Pay:** Implementierung sicherer Zahlungen durch Apple
- **Internetdienste:** Netzwerkbasierte Infrastruktur von Apple für Nachrichten, Synchronisation und Backup
- **Gerätesteuerungen:** Methoden, die die Verwaltung von iOS-Geräten erlauben, eine unbefugte Verwendung der Geräte verhindern und das Fernlöschen ermöglichen, falls das Gerät verloren oder gestohlen wurde
- **Datenschutzeinstellungen:** Möglichkeiten von iOS zum Steuern des Zugriffs auf Ortungsdienste und Benutzerdaten

Systemicherheit

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.

Verbinde das Gerät zuerst über ein USB-Kabel mit einem Computer.

Anschließend:

Auf iPhone X, iPhone 8 oder iPhone 8 Plus – Drücke kurz die Taste „Lauter“. Drücke kurz die Taste „Leiser“. Halte dann die Seitentaste gedrückt, bis der Bildschirm des Wartungsmodus angezeigt wird.

Auf iPhone 7 oder iPhone 7 Plus – Halte gleichzeitig die Seitentaste und die Taste „Leiser“ gedrückt. Halte sie so lange gedrückt, bis der Bildschirm des Wartungsmodus angezeigt wird.

Auf iPhone 6s und älter, iPad oder iPod touch – Halte gleichzeitig die Home-Taste und die Taste oben (bzw. an der Seite) gedrückt. Halte sie so lange gedrückt, bis der Bildschirm des Wartungsmodus angezeigt wird.

Hinweis: Wenn auf dem Gerät der DFU-Modus aktiv ist, bleibt der Bildschirm schwarz. Wenn das Apple-Logo angezeigt wird, hast du die Seitentaste oder den Ein-/Ausschalter zu lange gehalten.

Die Systemicherheit 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 Hardware, Software und Diensten 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. Dieser sichere Startvorgang sorgt dafür, dass die unteren Software-Ebenen nicht unbefugt manipuliert werden können.

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 iBoot-Bootloader 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. Bei Beendigung seiner Aufgaben prüft und führt iBoot den iOS-Kernel aus. Bei Geräten mit einem Prozessor der Reihe S1, A9 oder einer früheren A-Reihe lädt und prüft das Boot-Rom einen zusätzlichen Low-Level-Bootloader (LLB), der wiederum iBoot lädt und überprüft.

Wenn das Boot-Rom den LLB (auf älteren Geräten) oder iBoot (auf neueren Geräten) nicht laden kann, wird der DFU-Modus auf dem Gerät aufgerufen. Kann der LLB oder iBoot nicht geladen oder der nächste Schritt nicht überprüft werden, dann wird der Startvorgang abgebrochen, und auf dem Display wird „Mit iTunes verbinden“ angezeigt. Dies wird als Wartungsmodus bezeichnet. In beiden Fällen muss das Gerät per USB mit iTunes verbunden und auf die Werksvoreinstellungen zurückgesetzt werden.

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 einer Secure Enclave nutzt der „Secure Enclave“-Coprozessor ebenfalls einen sicheren Startvorgang, um zu überprüfen, ob seine Software von Apple überprüft und signiert wurde. Informationen hierzu findest du im Abschnitt „Secure Enclave“ in diesem Dokument.

Näheres zum manuellen Aufrufen des Wartungsmodus findest du unter <https://support.apple.com/de-de/HT201263>

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 zuvor 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*. Wenn 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 Secure Enclave 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. Informationen hierzu findest du im Abschnitt „Secure Enclave“ in diesem Dokument.

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 Mac mit macOS High Sierra und aktiviertem Inhaltscaching zwischengespeichert werden, damit iOS-Geräte die erforderliche Aktualisierung nicht erneut über das Internet laden müssen. Sie müssen allerdings immer noch eine Verbindung zu Apple-Servern herstellen, um den Aktualisierungsprozess abzuschließen.

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. 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 sichert und sie dafür verwenden kann, ein Gerät zu manipulieren oder die Systemsoftware anderweitig zu verändern.

Secure Enclave

Die Secure Enclave ist ein Coprozessor, der in Apple-Prozessoren der Reihe Apple T1, Apple S2, Apple S3, Apple A7 oder neueren Prozessoren der A-Reihe integriert ist. Sie nutzt einen verschlüsselten Speicher und verfügt über einen Hardwarezufallsgenerator. Die Secure Enclave stellt sämtliche Verfahren für die Schlüsselverwaltung zur Datensicherheit bereit und garantiert auch dann den Schutz der Daten, wenn der Kernel beeinträchtigt wurde. Die Kommunikation zwischen der Secure Enclave und dem Anwendungsprozessor findet isoliert in einem Interrupt-gesteuerten Postfach und Shared-Memory-Datenpuffern statt.

Die Secure Enclave verwendet eine Apple-spezifische Version des L4-Microkernels. Der Microkernel wird von Apple signiert, als Teil des sicheren iOS-Startvorgangs verifiziert und über einen personalisierten Softwareaktualisierungsprozess aktualisiert.

Wenn das Gerät startet, wird ein temporärer Schlüssel erstellt, der mit der UID des Geräts verknüpft ist und verwendet wird, um den Teil des Speicherplatzes des Geräts für die Secure Enclave zu verschlüsseln. Der Speicher der Secure Enclave wird auch mit dem temporären Schlüssel authentifiziert; ausgenommen hiervon ist der Apple A7-Prozessor. Beim Apple A11-Prozessor wird eine Integritätsstruktur verwendet, um ein Replay des sicherheitskritischen Secure Enclave-Speichers zu verhindern. Diese wird durch den temporären Schlüssel und die Nonces authentifiziert, die im SRAM auf dem Chip gespeichert sind.

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. Anti-Replay-Dienste in Secure Enclave werden verwendet, um Daten über Ereignisse zu widerrufen, die Anti-Replay-Grenzen markieren. Dazu gehören unter anderem:

- Code ändern
- Touch ID oder Face ID aktivieren/deaktivieren
- Fingerabdruck hinzufügen/löschen
- Face ID zurücksetzen
- Apple Pay-Karte hinzufügen/entfernen
- Inhalte & Einstellungen löschen

Die Secure Enclave ist auch zuständig für die Verarbeitung des Fingerabdrucks und der Gesichtsdaten der Touch ID- und Face ID-Sensoren. Sie bestimmt, ob es eine Übereinstimmung gibt, und ermöglicht anschließend den Zugriff oder Kauf im Namen des Benutzers.

Touch ID

Touch ID ist das Fingerabdrucksensorsystem, mit dem ein sicherer Zugriff auf das iPhone und iPad 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.

Face ID

Mit Face ID kannst du dein iPhone X mit einem einfachen Blick sicher entsperren. Das TrueDepth-Kamerasystem bietet eine intuitive und sichere Authentifizierung, die modernste Technologien verwendet, um die Geometrie deines Gesichts präzise zu erfassen. Face ID erkennt anhand der Blickrichtung, dass dein Gesicht auf das Gerät gerichtet ist und du mit dem Gerät interagieren möchtest, und verwendet dann neurale Netzwerke für den Abgleich und zum Verhindern von Betrug, damit du dein Telefon mit einem Blick entsperren kannst. Face ID passt sich automatisch an Änderungen in deinem Aussehen an und schützt deine Privatsphäre und die Sicherheit deiner biometrischen Daten.

Touch ID, Face ID und Codes

Für Touch ID oder Face ID musst du dein Gerät so einrichten, dass zum Entsperren ein Code benötigt wird. Erkennen Touch ID oder Face ID eine erfolgreiche Übereinstimmung, wird dein Gerät entsperrt, ohne dass nach dem Code für das Gerät gefragt wird. Das vereinfacht die Verwendung längerer, komplexerer Codes, da du diese weniger häufig eingeben musst. Touch ID und Face ID ersetzen deinen Code nicht, sondern ermöglichen den erleichterten Zugriff auf dein Gerät innerhalb ausgeklügelter Grenzen und Zeitbeschränkungen. Dies ist wichtig, da ein sicherer Code die Grundlage für den kryptografischen Schutz deines iOS-Geräts ist.

Du kannst den Code immer anstelle von Touch ID oder Face ID verwenden und er 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.
- Der Code wurde innerhalb der letzten 156 Stunden (6,5 Tage) nicht zum Entsperren des Geräts verwendet und das Gerät innerhalb der letzten vier Stunden nicht über Face ID entsperrt.
- Das Gerät wurde per Fernzugriff gesperrt.
- Es wurde fünf Mal kein erfolgreicher Abgleich ausgeführt.
- Nach dem Starten der Abschaltung oder eines Notrufs.

Wenn Touch ID oder Face ID aktiviert ist, wird das Gerät sofort gesperrt, wenn du auf die Seitentaste drückst. Außerdem wird das Gerät immer gesperrt, wenn der Ruhezustand aktiviert wird. Touch ID und Face ID benötigen einen erfolgreichen Abgleich (oder optional den Code), um den Ruhezustand zu beenden.

Die Wahrscheinlichkeit, dass eine beliebige andere Person auf dein iPhone X schaut und es mit Face ID entsperrt, liegt ungefähr bei 1:1.000.000 (im Vergleich zu 1:50.000 bei Touch ID). Als zusätzlichen Schutz lassen Touch ID und Face ID nur fünf fehlgeschlagene Abgleichversuche zu, bevor ein Code erforderlich ist, um Zugriff auf dein Gerät zu erhalten. Die Wahrscheinlichkeit einer falschen Übereinstimmung bei Face ID weicht bei Zwillingen und Geschwistern, die dir ähnlich sehen, sowie bei Kindern unter 13 Jahren ab, weil ihre individuellen Gesichtszüge möglicherweise noch nicht voll ausgeprägt sind. Wenn du aus diesem Grund Bedenken hast, empfiehlt Apple die Verwendung eines Codes für die Authentifizierung.

Sicherheit mit Touch ID

Der Fingerabdrucksensor ist nur aktiv, wenn der kapazitive Berührungssensor in dem Edelstahlring, der die Home-Taste umgibt, eine Fingerberührung erkennt, wodurch wiederum der Fingerabdruckscanner ausgelöst und das Ergebnis dann an Secure Enclave gesendet wird. 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 mit einem Sitzungsschlüssel verschlüsselt und authentifiziert, der mit einem gemeinsamen Schlüssel ausgehandelt wird und für jeden Touch ID-Sensor und die zugehörige Secure Enclave bei der Fertigung festgelegt wird. Der gemeinsame Schlüssel ist sicher, zufällig und für jeden Touch ID-Sensor unterschiedlich. 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.

Das Rasterbild 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 Fingerabdrucks 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.

Sicherheit mit Face ID

Face ID wurde entwickelt, um anhand der Blickrichtung zu erkennen, dass du mit dem Gerät interagieren möchtest, eine zuverlässige Authentifizierung mit einer niedrigen Fehlerrate beim Abgleich zu bieten und digitaler oder physischer Manipulation entgegenzuwirken.

Die TrueDepth-Kamera sucht automatisch nach deinem Gesicht, wenn du durch Anheben oder durch Tippen auf den Bildschirm den Ruhezustand deines iPhone X beendest, wenn es versucht, dich zum Anzeigen einer eingehenden Benachrichtigung zu authentifizieren, oder wenn eine unterstützte App eine Face ID-Authentifizierung anfordert. Wenn ein Gesicht erkannt wird, bestätigt Face ID deine Absicht zum Entsperren des Geräts, indem erkannt wird, ob deine Augen geöffnet und auf das Gerät gerichtet sind. Für den behindertengerechten Zugriff ist die Funktion deaktiviert, wenn VoiceOver aktiviert ist; und sie kann bei Bedarf auch separat deaktiviert werden.

Sobald ein auf das Gerät gerichtetes Gesicht bestätigt wurde, projiziert und liest die TrueDepth-Kamera mehr als 30.000 Infrarotpunkte, um eine Tiefendarstellung und ein 2D-Infrarotbild des Gesichts zu erstellen. Diese Daten werden verwendet, um eine Reihe von 2D-Bildern und Tiefendarstellungen zu erstellen, die digital signiert und an Secure Enclave gesendet werden. Als Gegenmaßnahme gegen digitale und physische Manipulation erstellt die TrueDepth-Kamera eine zufällige Reihenfolge der 2D-Bilder und Tiefendarstellungen und projiziert ein gerätespezifisches Zufallsmuster. Ein Teil der neuronalen Engine des A11-Bionic-Chips, der mit Secure Enclave geschützt ist, wandelt die Daten in eine mathematische Darstellung um und vergleicht diese Daten mit den registrierten Gesichtsdaten. Die registrierten Gesichtsdaten wiederum sind selbst eine mathematische Darstellung deines Gesichts in einer Vielzahl von Posen.

Der Gesichtsabgleich in Secure Enclave wird mit neuronalen Netzwerken durchgeführt, die speziell für diesen Zweck trainiert wurden. Wir haben die neuronalen Netzwerke für den Gesichtsabgleich mit über einer Milliarde Bildern entwickelt, einschließlich IR- und Tiefenbilder, die mit dem Einverständnis der Teilnehmer an Studien erfasst wurden. Apple hat mit Teilnehmern auf der ganzen Welt zusammengearbeitet, um eine repräsentative Personengruppe abzudecken, die sich in Geschlecht, Alter, Herkunft und anderen Faktoren unterscheiden. Die Studien wurden bei Bedarf verbessert, um einen hohen Grad an Genauigkeit für unterschiedliche Benutzer sicherzustellen. Face ID funktioniert mit Hüten, Halstüchern, Brillen, Kontaktlinsen und vielen Sonnenbrillen. Außerdem funktioniert es drinnen, draußen und sogar bei völliger Dunkelheit. Ein zusätzliches neuronales Netzwerk, das trainiert wurde, um Manipulationen zu erkennen, schützt dich vor Versuchen, dein iPhone X mit Fotos oder Masken zu entsperren.

Face ID-Daten (einschließlich mathematischer Darstellungen deines Gesichts) werden verschlüsselt und sind nur für Secure Enclave verfügbar. Diese Daten werden ausschließlich auf deinem Gerät gesichert. Sie werden nicht an Apple gesendet und sind nicht Teil einer Gerätesicherung. Die folgenden Face ID-Daten werden im Normalbetrieb gesichert und nur zur Verwendung durch Secure Enclave verschlüsselt:

- Die mathematische Darstellung deines Gesichts, die bei der Registrierung berechnet wurde.
- Die mathematische Darstellung deines Gesichts, die bei einem Versuch berechnet wurde, das Gerät zu entsperren, wenn Face ID diese als hilfreich für die Verbesserung eines zukünftigen Abgleichs einstuft.

Bilder des Gesichts, die im Normalbetrieb erfasst werden, werden nicht gesichert, sondern sofort nach der Berechnung der mathematischen Darstellung für die Registrierung oder einen Vergleich mit den registrierten Face ID-Daten verworfen.

Entsperren eines iOS-Geräts mit Touch ID oder Face ID

Wenn Touch ID oder Face 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 Schlüsselbunds dieser Klasse sind erst wieder verfügbar, wenn du das Gerät durch Eingabe des Codes wieder entsperrst.

Bei aktivierter Touch ID oder Face ID werden die Schlüssel nicht verworfen, wenn das Gerät gesperrt wird, sondern stattdessen mit einem Schlüssel verpackt, der an das Touch ID- oder Face ID-Teilsystem in der Secure Enclave übergeben wird. Wenn du versuchst, das Gerät zu entsperren, und der Abgleich erfolgreich ist, stellt das Gerät den Schlüssel zum Entpacken der Datensicherheitsschlüssel bereit und das Gerät wird entsperrt. Dieses Verfahren bietet zusätzlichen Schutz, da die Teilsysteme „Datensicherheit“ und „Touch ID“ oder „Face ID“ zusammenarbeiten müssen, damit das Gerät entsperrt werden kann.

Wenn das Gerät neu gestartet wird, sind die erforderlichen Schlüssel zum Entsperren des Geräts mit Touch ID oder Face ID verloren; sie werden von Secure Enclave verworfen, nachdem eine beliebige Bedingung erfüllt ist, die eine Eingabe des Codes erfordert (zum Beispiel, nachdem das Gerät 48 Stunden nicht entsperrt wurde oder nach fünf fehlgeschlagenen Abgleichsversuchen).

Zur Verbesserung der Leistung beim Entsperren und zur Kompensation natürlicher Veränderungen in deinem Gesicht und Aussehen verbessert Face ID die gespeicherte mathematische Darstellung im Laufe der Zeit. Bei einem erfolgreichen Entsperren verwendet Face ID die neu berechnete mathematische Darstellung (sofern sie eine ausreichende Qualität hat) möglicherweise für eine begrenzte Anzahl weiterer Entsperrvorgänge, bevor die Daten verworfen werden. Wenn Face ID dich nicht erkennt, die Abgleichqualität aber über einem bestimmten Schwellenwert liegt, und du direkt nach dem fehlgeschlagenen Versuch deinen Code eingibst, erstellt Face ID im Gegenzug eine weitere Aufnahme und verbessert die registrierten Face ID-Daten mit der neu berechneten mathematischen Darstellung. Die neuen Face ID-Daten werden verworfen, wenn du damit nicht mehr abgeglichen werden kannst, oder nach einer begrenzten Anzahl von Entsperrvorgängen. Mit diesen Verbesserungsprozessen kann Face ID sich an deutliche Änderungen deiner Gesichtsbehaarung oder deines Makeups anpassen und gleichzeitig falsche Zugriffe minimieren.

Touch ID, Face ID und Apple Pay

Du kannst Touch ID und Face ID auch mit Apple Pay verwenden, um einfach und sicher in Stores, Apps und im Web einzukaufen. Weitere Informationen zu Touch ID und Apple Pay findest du im Abschnitt „Apple Pay“ in diesem Dokument.

Um eine Zahlung im Store mit Face ID zu autorisieren, musst du zuerst deine Zahlungsabsicht mit einem Doppelklick auf die Seitentaste bestätigen. Danach musst du eine Authentifizierung mit Face ID durchführen, bevor du dein iPhone X in die Nähe des kontaktlosen Zahlungsterminals hältst. Wenn du nach der Face ID-Authentifizierung eine andere Apple Pay-Zahlungsmethode auswählen willst, musst du dich erneut authentifizieren; du musst aber nicht noch einmal auf die Seitentaste doppelklicken.

Für eine Zahlung in Apps und im Web bestätigst du die Zahlungsabsicht mit einem Doppelklick auf die Seitentaste. Dann authentifizierst du dich mit Face ID, um die Zahlung zu autorisieren. Wenn die Apple Pay-Transaktion nicht innerhalb von 30 Sekunden nach dem Doppelklick auf die Seitentaste abgeschlossen ist, musst du die Zahlungsabsicht noch einmal mit einem Doppelklick bestätigen.

Face ID-Diagnose

Face ID-Daten verbleiben immer auf deinem Gerät und werden nie in iCloud oder an einem anderen Ort gesichert. Nur wenn du AppleCare Face ID-Diagnosedaten zur Unterstützung bereitstellen willst, werden diese Informationen von deinem Gerät übertragen. Das Aktivieren der Face ID-Diagnose erfordert eine digital signierte Autorisierung von Apple, ähnlich wie beim Personalisierungsprozess bei einer Softwareaktualisierung. Nach der Autorisierung kannst du die Face ID-Diagnose aktivieren und den Konfigurationsprozess in den Einstellungen auf deinem iPhone X starten.

Als Teil der Konfiguration der Face ID-Diagnose wird deine vorhandene Face ID-Registrierung gelöscht und du wirst gebeten, dich erneut bei Face ID zu registrieren. Dein iPhone X beginnt dann mit dem Speichern von Face ID-Bildern, die bei Authentifizierungsversuchen der darauf folgenden 10 Tagen erfasst werden; Anschließend stoppt dein iPhone X automatisch das Speichern der Bilder. Die Face ID-Diagnose sendet nicht automatisch Daten an Apple. Du kannst Face ID-Diagnosedaten (einschließlich Bilder für fehlgeschlagene und erfolgreiche Registrierungen und Entsperrungen) prüfen, die im Diagnosemodus gesammelt werden, bevor sie an Apple gesendet werden. Die Face ID-Diagnose lädt nur die Face ID-Diagnosebilder hoch, die du bestätigt hast; die Daten werden vor dem Hochladen verschlüsselt und auf deinem iPhone X gelöscht, unmittelbar nachdem das Hochladen abgeschlossen ist. Bilder, die du ablehnst, werden sofort gelöscht.

Wenn du die Face ID-Diagnosesitzung nicht abschließt, indem du die Bilder prüfst und bestätigte Bilder hochlädst, wird die Face ID-Diagnose automatisch nach 40 Tagen beendet und alle Diagnosebilder werden von deinem iPhone X gelöscht. Du kannst die Face ID-Diagnose auch jederzeit deaktivieren. In diesem Fall werden alle lokalen Bilder sofort gelöscht und keine Face ID-Daten mit Apple geteilt.

Andere Verwendungen für Touch ID und Face ID

Apps anderer Anbieter können vom System bereitgestellte APIs verwenden, um den Benutzer aufzufordern, sich mit Touch ID, Face ID oder einem Code zu authentifizieren. Apps, die Touch ID unterstützen, unterstützen ohne jegliche Änderungen automatisch auch Face ID. Wenn Touch ID oder Face ID verwendet wird, wird die App nur benachrichtigt, ob die Authentifizierung erfolgreich war, sie kann aber nicht auf Touch ID, Face ID oder die mit dem registrierten Benutzer verbundenen Daten zugreifen. Objekte im Schlüsselbund können ebenfalls mit Touch ID oder Face ID so geschützt werden, dass sie über die Secure Enclave nur mit einem erfolgreichen Abgleich oder dem Code für das Gerät freigegeben werden können. App-Entwicklern stehen APIs zur Verfügung, um zu prüfen, ob der Benutzer einen Code festgelegt hat, bevor Objekte im Schlüsselbund mit Touch ID, Face ID oder einem Code entsperrt werden müssen. App-Entwickler haben folgende Möglichkeiten:

- Sie können vorgeben, dass API-Operationen für die Authentifizierung weder auf das Passwort einer App noch auf den Code für das Gerät zurückgreifen. Sie können abfragen, ob ein Benutzer registriert ist, und somit Touch ID oder Face ID als zweiten Faktor in sicherheitskritischen Apps verwenden.
- Sie können ECC-Schlüssel in Secure Enclave generieren und verwenden, die durch Touch ID oder Face ID geschützt werden können. Operationen mit diesen Schlüsseln erfolgen stets in der Secure Enclave, nachdem diese die Nutzung autorisiert hat.

Du kannst Touch ID oder Face ID auch so konfigurieren, dass damit Käufe im iTunes Store, App Store oder iBooks Store bestätigt werden können, damit du das Passwort für deine Apple-ID nicht eingeben musst. Bei iOS 11 oder neuer werden durch Touch ID und Face ID geschützte ECC-Schlüssel der Secure Enclave verwendet, um einen Kauf durch Signieren der Store-Anfrage zu autorisieren.

Verschlüsselung und Datensicherheit

Inhalte & Einstellungen löschen

Mit der Option „Inhalte & Einstellungen löschen“ in den Einstellungen werden alle Schlüssel im Effaceable Storage gelöscht. Dadurch kann nicht mehr auf die kryptografischen Benutzerdaten auf dem Gerät zugegriffen werden. Dies ist daher die beste Methode, um sicherzustellen, dass alle persönlichen Informationen auf einem Gerät gelöscht werden, bevor du es weitergibst oder zur Wartung bringst.

Achtung: Verwende die Option „Inhalte & Einstellungen löschen“ nur, wenn das Gerät gesichert wurde, da es nicht möglich ist, die gelöschten Daten wiederherzustellen.

Der sichere Startvorgang, die Code-Signierung und die 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 entscheidende 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 Implementierung diese Prioritäten nicht berücksichtigt werden.

In jedem iOS-Gerät ist eine dedizierte AES-256 Crypto Engine im DMA-Pfad zwischen dem Flash-Speicher und dem Hauptspeicher vorhanden, was eine höchst effiziente Dateiverschlüsselung ermöglicht. Bei A9-Prozessoren oder Prozessoren einer neueren A-Reihe befindet sich das Subsystem des Flash-Speichers auf einem isolierten Bus, der nur über die DMA Crypto Engine auf den Speicher zugreifen kann, in dem die Benutzerdaten enthalten sind.

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. Auf die UID und die GID kann auch nicht über JTAG oder andere Debugging-Schnittstellen zugegriffen werden.

Bei Prozessoren der Reihe T1, S2, S3 und A9 oder neueren Prozessoren der A-Reihe generiert Secure Enclave jeweils eine eigene UID (Unique ID, eindeutige Kennung). Weil die UID für jedes Gerät eindeutig ist und ausschließlich innerhalb von Secure Enclave generiert wird, anstatt in einem Fertigungssystem außerhalb des Geräts, können Apple oder seine Lieferanten nicht auf die UID zugreifen und sie auch nicht speichern. Software, die in Secure Enclave ausgeführt wird, nutzt die Vorteile der UID, um gerätespezifische Geheimnisse zu schützen.

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.

Die GID wird auf allen Prozessoren einer Geräteklasse (zum Beispiel alle Geräte mit dem Apple A8-Prozessor) gemeinsam genutzt.

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 Systementropie wird beim Starten aus Zeitabweichungen und nach abgeschlossenem Startvorgang

zusätzlich aus dem Interrupt-Timing erzeugt. In der Secure Enclave erzeugte Schlüssel verwenden den eigenen echten Hardwarezufallszahlengenerator, der auf mehreren Ringoszillatoren basiert und mit CTR_DRBG nachbearbeitet wird.

Das sichere Löschen gespeicherter Schlüssel ist genauso wichtig wie deren Erstellung. Dies ist insbesondere bei Flash-Speichern eine 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.

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. Der Datenschutz 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. Durch die Einführung des Apple-Dateisystems (Apple File System, APFS) kann das Dateisystem die Schlüssel anhand der jeweiligen Länge weiter unterteilen (Teile einer Datei können verschiedene Schlüssel haben).

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. (Auf Geräten mit einem A8-Prozessor oder neuer wird AES-XTS verwendet.) Der Initialisierungsvektor (IV) wird anhand des Block-Offsets in der Datei berechnet und mit dem SHA-1 Hash des pro Datei erzeugten Schlüssels verschlüsselt.

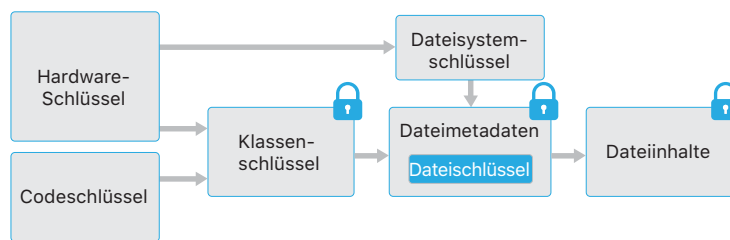
Der pro Datei (oder pro Länge) erzeugte Schlüssel wird mit einem von mehreren Klassenschlüsseln sicher verpackt. Der ausgewählte Schlüssel richtet sich nach den Umständen, unter denen die Datei zugänglich sein soll. Das sichere Verpacken wird immer mit NIST AES Key Wrapping, gemäß RFC 3394, durchgeführt.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

Geräte mit dem Apple-Dateisystemformat können das Klonen von Dateien (Kopien ohne Aufwand durch Copy-on-Write-Technologie (COW)) unterstützen. Wenn eine Datei geklont wird, wird jeder Hälfte des Klons ein neuer Schlüssel zum Akzeptieren von eingehenden Schreib Anforderungen zugeordnet, damit neue Daten mit einem neuen Schlüssel auf die Datenträger geschrieben werden. Mit der Zeit setzt sich die Datei aus verschiedenen Längen (oder

Fragmenten) zusammen, die jeweils auf eigene Schlüssel verweisen. Alle Teile, aus denen sich eine Datei zusammensetzt, werden jedoch durch denselben Klassenschlüssel geschützt.

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 (oder pro Länge) 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 gesamte Verarbeitung des verpackten Dateischlüssels erfolgt innerhalb der Secure Enclave; der Dateischlüssel wird zu keinem Zeitpunkt gegenüber dem Anwendungsprozessor offen gelegt. Beim Booten handelt die Secure Enclave einen temporären Schlüssel mit der AES-Engine aus. Schlüssel einer Datei, die innerhalb der Secure Enclave entschlüsselt werden, werden mit diesem temporären Schlüssel neu verschlüsselt und erst in dieser Form an den Anwendungsprozessor zurück gesendet.

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. Auf Geräten, die das Apple-Dateisystem unterstützen, wird der Metaschlüssel des Dateisystems mit dem UID-Schlüssel von Secure Enclave für den Langzeitspeicher verschlüsselt. Genau wie die Schlüssel pro Datei (oder pro Länge) wird der Metaschlüssel dem Anwendungsprozessor nie direkt offen gelegt; Secure Enclave stellt stattdessen eine temporäre Version pro Bootvorgang bereit. Wenn er gespeichert wird, wird der verschlüsselte Dateisystemschlüssel zusätzlich mit einem „auslöschbaren Schlüssel“ verschlüsselt, der im Effaceable Storage gespeichert ist. Dieser Schlüssel sorgt nicht für zusätzliche Vertraulichkeit der Daten. Stattdessen ist er so konzipiert, dass er auf Wunsch schnell gelöscht werden kann (durch den Benutzer mit der Option „Inhalte & Einstellungen löschen“ oder durch einen Benutzer oder Administrator, der einen Befehl zum Fernlöschen in einer MDM-Lösung, in Exchange ActiveSync oder iCloud absendet). Wird der Schlüssel auf diese Weise gelöscht, werden alle Dateien kryptografisch unzugänglich gemacht.



Der Inhalt einer Datei kann mit einem oder mehreren pro Datei (oder pro Länge) erzeugten Schlüsseln verschlüsselt werden, die mit einem Klassenschlüssel verpackt und in den Metadaten der Datei gespeichert werden, 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 erzeugte Schlüssel neu verpackt werden. Bei Änderung des Codes wird nur der Klassenschlüssel neu verpackt.

Erwägungen zum Gerätecode

Wenn ein langes Passwort eingegeben wird, das nur aus Ziffern besteht, wird auf dem Sperrbildschirm ein numerisches Tastenfeld anstelle der vollständigen Tastatur angezeigt. Ein längerer Gerätecode, der nur aus Ziffern besteht, kann einfacher einzugeben sein als ein kürzerer alphanumerischer Gerätecode und bietet ähnlich hohe Sicherheit.

Verzögerungen zwischen Codeeingabeversuchen

Versuche	Erzwungene Verzögerung
1–4	keine
5	1 Minute
6	5 Minuten
7–8	15 Minuten
9	1 Stunde

Gerätecodes

Durch das Einrichten eines Gerätecodes aktiviert der Benutzer automatisch die Datensicherheit. iOS unterstützt Gerätecodes, die aus sechs oder vier Ziffern bestehen, und alphanumerische Gerätecodes beliebiger Länge. Zusätzlich zum Entsperren des Geräts stellt der Gerätecode die Entropie für bestimmte Verschlüsselungscodes 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 fünfzehn Jahre dauern würde, alle sechsstelligen alphanumerischen Gerätecodes, die aus Kleinbuchstaben und Ziffern bestehen, auszuprobieren.

Je sicherer ein Benutzercode ist, desto sicherer ist auch der Verschlüsselungscode. Touch ID und Face ID können 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 den Datenschutz verwendeten Verschlüsselungsschlüssel geschützt werden, ohne dass die Benutzerfreundlichkeit leidet, wenn iOS-Geräte mehrmals täglich entsperrt werden müssen.

Um Brute-Force-Codeangriffe noch besser abzuwehren, werden zunehmende zeitliche Verzögerungen eingesetzt, wenn ein ungültiger Gerätecode auf dem Sperrbildschirm eingegeben wurde. Wenn du „Daten löschen“ im Bereich „Einstellungen“ > „Touch ID & Code“ aktivierst, wird nach zehn aufeinander folgenden Fehlversuchen, den Code einzugeben, das Gerät automatisch gelöscht. Diese Einstellung ist auch als Verwaltungsrichtlinie über MDM und Exchange ActiveSync verfügbar, und die maximal zulässige Anzahl von Fehleingaben kann zudem verringert werden.

Bei Geräten mit einer Secure Enclave werden die Verzögerungen durch den „Secure Enclave“-Coprozessor erzwungen. Wird das Gerät während einer zeitlich fixierten Verzögerung neu gestartet, wird die Verzögerung dennoch durchgesetzt und der Timer auf den Anfang des aktuellen Verzögerungsintervalls zurückgesetzt.

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 ein Benutzer ein Gerät sperrt (10 Sekunden, wenn die Einstellung „Passwort erforderlich“ auf „Sofort“ festgelegt ist), wird der entschlüsselte Klassenschlüssel verworfen, damit auf alle Daten in dieser Klasse erst wieder zugegriffen werden kann, wenn der Benutzer erneut den Code eingibt oder das Gerät mit Touch ID oder Face ID entsperrt.

Geschützt, außer wenn offen

(`NSFileProtectionCompleteUnlessOpen`): Möglicherweise müssen einige Dateien geschrieben werden, während das Gerät gesperrt ist. Ein gutes Beispiel sind E-Mail-Anhänge, die im Hintergrund geladen werden. Dieses Verhalten wird durch die Verschlüsselung mit asymmetrischen elliptischen Kurven (ECDH über Curve25519) erreicht. Der übliche Schlüssel pro Datei wird durch einen Schlüssel geschützt, der mit Einweg-Diffie-Hellman-Schlüsselaustausch abgeleitet wird, das in der NIST SP 800-56A beschrieben ist.

Der temporäre öffentliche Schlüssel für den Austausch wird zusammen mit dem verpackten, pro Datei erzeugten Schlüssel gespeichert. Hierfür wird die KDF (Concatenation Key Derivation Function, anerkannte Alternative 1) verwendet, wie unter 5.8.1 der NIST SP 800-56A beschrieben. Die Algorithmus-ID wird weggelassen. Als temporäre bzw. statisch öffentliche 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. Diese werden 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, allerdings 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).

Schlüssel der Datenschutzzklasse

Klasse A Vollständiger Schutz	(<code>NSFileProtectionComplete</code>)
Klasse B Geschützt, außer wenn offen	(<code>NSFileProtectionCompleteUnlessOpen</code>)
Klasse C Geschützt bis zur ersten Benutzerauthentifizierung	(<code>NSFileProtectionCompleteUntilFirstUserAuthentication</code>)
Klasse D Kein Schutz	(<code>NSFileProtectionNone</code>)

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. Und schließlich enthalten sie auch Verschlüsselungsdaten, u. a.:

- 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“ weitergeben), als binäre plist codiert und mit dem pro Datei erzeugten Schlüssel verschlüsselt werden

Als Verschlüsselung wird AES 128 im Galois/Counter Mode (GCM) 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 (z. B. 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 Berechtigungen „keychain-access-groups“, „application-identifier“ und „application-group“ abfragt. Anstatt den Zugriff auf einen einzelnen Prozess einzuschränken, ermöglichen es Zugriffsgruppen, Schlüsselbundeinträge zwischen Apps zu teilen.

Schlüsselbundeinträge können nur zwischen Apps desselben Entwicklers gemeinsam genutzt werden. Apps anderer Anbieter werden angewiesen, Zugriffsgruppen mit einem Präfix zu verwenden, das ihnen durch das Apple-Entwicklerprogramm über Anwendungsgruppen zugewiesen wurde. Das Präfix-Erfordernis und die Einzigartigkeit der Anwendungsgruppe werden über die Codesignierung, die Bereitstellungsprofile und das Apple-Entwicklerprogramm sichergestellt.

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

Verfügbar	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	Nicht verfügbar	kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly

Apps, die Hintergrundaktualisierungsdienste nutzen, können

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

Die Klasse `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly` verhält sich ähnlich wie die Klasse `kSecAttrAccessibleWhenUnlocked`, ist aber nur verfügbar, wenn das Gerät mit einem Gerätecode konfiguriert wurde. Diese Klasse existiert nur im System-Keybag; ihre Objekte werden nicht mit dem iCloud-Schlüsselbund synchronisiert, sie 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 Geräts 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 Informationen 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
Safari-Lesezeichen	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
Von einem 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 Face ID oder durch Eingabe des Gerätecodes authentifizieren muss, um auf sie zugreifen zu können. Der Zugriff auf Objekte kann auch eingeschränkt werden, indem festgelegt wird, dass die Touch ID- oder Face ID-Registrierung nicht geändert wurde, seit das Objekt hinzugefügt wurde. Mit dieser Einschränkung lässt sich verhindern, dass ein Angreifer seinen eigenen Fingerabdruck hinzufügt, um so auf ein Objekt im Schlüsselbund zuzugreifen. ACLs 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 Datei mit der eindeutigen Kennung der Apps befinden, die zugelassen sind. Wird eine App mit der Berechtigung „com.apple.developer.associated-domains“ installiert, stellt iOS eine TLS-Anfrage für die Datei „/apple-app-site-association“ an alle aufgeführten Websites. Wenn in der Datei die Kennung der zu installierenden App aufgelistet ist, 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.

Unter iOS können Benutzer gespeicherte Benutzernamen und Passwörter in Felder mit Anmeldeinformationen in Apps eingeben, indem sie in der QuickType-Leiste der iOS-Tastatur auf eine Einverständnistaste tippen. Dabei wird derselbe Mechanismus „apple-app-site-association“ verwendet, um Apps und Websites eng zu verbinden. Die Schnittstelle legt einer App keine Anmeldeinformationen offen, bis der Benutzer zustimmt, sie für die App freizugeben. Wenn iOS eine Website und eine App mit einer vertrauenswürdigen Verbindung markiert hat, schlägt die QuickType-Leiste automatisch Anmeldeinformationen vor, mit denen die App ausgefüllt wird. So können Benutzer in Safari gespeicherte Anmeldeinformationen für Apps mit demselben Sicherheitshintergrund offenlegen, ohne dass Apps eine API übernehmen müssen.

Keybags

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

Im **User-Keybag** werden die verpackten Klassenschlüssel gespeichert, die im normalen Betrieb des Geräts verwendet werden. Wenn beispielsweise ein Gerätecode eingegeben wird, wird der Schlüssel `NSFileProtectionComplete` aus dem User-Keybag geladen und entpackt. Es handelt sich um eine binäre plist, die in der Klasse „Kein Schutz“ gespeichert ist und deren Inhalte 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 User-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 User-Keybag zugegriffen werden kann und sie erfolgreich entpackt wurden.

Im **Device-Keybag** werden die verpackten Klassenschlüssel gespeichert, die für Operationen mit gerätespezifischen Daten verwendet werden. iOS-Geräte, die für die gemeinsame Nutzung konfiguriert sind, benötigen manchmal Zugriff auf Anmeldeinformationen, bevor sich ein Benutzer angemeldet hat. Dies erfordert einen Keybag, der nicht mit dem Code eines Benutzers geschützt ist. Eine kryptografische Trennung von benutzerbasierten Dateisysteminhalten wird von iOS nicht unterstützt; das heißt, das System verwendet Klassenschlüssel

aus dem Device-Keybag, um pro Datei erzeugte Schlüssel zu verpacken. Der Schlüsselbund verwendet hingegen Klassenschlüssel des User-Keybag, um Objekte im Schlüsselbund des Benutzers zu schützen. Bei iOS-Geräten, die für die Nutzung durch einen einzigen Benutzer konfiguriert sind (Standardkonfiguration), sind Device-Keybag und User-Keybag identisch und mit dem Code des Benutzers geschützt.

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 zuvor 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 Millionen 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 Backup-Dateien unabhängig von ihrer Datenschutzklasse 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 Code eingeben muss; außerdem ermöglicht er es einer MDM-Lösung, den Code eines Benutzers fernzulöschen. Er wird auf dem Computer gespeichert, der zum Synchronisieren von iTunes verwendet wird, oder in der MDM-Lösung, die 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.

Bei OTA-Softwareaktualisierungen wird der Benutzer vor dem Starten der Aktualisierung nach seinem Code gefragt. Auf dieser Basis wird in einer sicheren Umgebung ein nur einmal nutzbares Token erstellt, das nach dem Aktualisieren zum Entsperren des User-Keybag verwendet wird. Dieses Token kann nicht generiert werden, ohne dass der Code des Benutzers eingegeben wird. Wird der Code des Benutzers geändert, wird ein zuvor erstelltes Token hinfällig.

Einmal nutzbare Tokens für das Entsperren können für die beaufsichtigte und auch die unbeaufsichtigte Installation einer Softwareaktualisierung verwendet werden. Diese Token werden mithilfe eines Codes verschlüsselt, der aus dem aktuellen Wert eines monotonen Zählers in der Secure Enclave, der UUID des Keybag und der UID der Secure Enclave gebildet wird.

Erhöht sich in der Secure Enclave der Wert des Zählers für die einmal nutzbaren Token zum Entsperren, verliert ein ggf. vorhandenes Token seine Gültigkeit. Der Wert des Zählers erhöht sich, wenn ein Token verwendet wird, wenn ein Gerät nach einem Neustart erstmals entsperrt wird, wenn eine Softwareaktualisierung (durch den Benutzer oder das System) abgebrochen wird oder wenn der richtlinienspezifische Timer für das Token abläuft.

Ein einmal nutzbares Token, das zum Entsperren nach einer beaufsichtigten Softwareaktualisierung verwendet wird, verliert nach 20 Minuten seine Gültigkeit. Ein solches Token wird aus der Secure Enclave exportiert und im Effaceable Storage gespeichert. Ein richtlinienspezifischer Timer veranlasst, dass der Wert des Zählers erhöht wird, wenn das Gerät nicht innerhalb von 20 Minuten neu gebootet wird.

Eine unbeaufsichtigte Softwareaktualisierung liegt vor, wenn sich der Benutzer bei der Benachrichtigung, dass eine Aktualisierung verfügbar ist, für die Option „Später installieren“ entscheidet. In diesem Fall kann der Anwendungsprozessor das nur einmal nutzbare Token zum Entsperren für maximal acht Stunden in der Secure Enclave vorhalten. Nach Ablauf dieser Zeitspanne veranlasst der richtlinienspezifische Timer, dass der Wert des Zählers erhöht wird.

Der **iCloud-Backup-Keybag** ist dem Backup-Keybag ähnlich. Alle Klassenschlüssel in diesem Keybag sind asymmetrisch (es wird Curve25519 verwendet, wie bei der Datenschutzklasse „Geschützt außer wenn offen“), sodass iCloud-Backups im Hintergrund durchgeführt werden können. Bei allen Datenschutzklassen 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-Schlüsselbunds verwendet.

Sicherheitszertifizierungen und -programme

Hinweis: Weitere Informationen über für die iOS-Sicherheit relevante Zertifizierungen, Validierungen und Empfehlungen findest du unter: <https://support.apple.com/de-de/HT202739>.

ISO 27001- und ISO 27018-Zertifizierungen

Apple besitzt die ISO 27001- und ISO 27018-Zertifizierungen für das Managementsystem zur Informationssicherheit für Infrastruktur, Entwicklung und Verfahren, die folgende Produkte und Dienste unterstützen: Apple School Manager, iCloud, iMessage, FaceTime, verwaltete Apple-IDs und iTunes U in Übereinstimmung mit dem Statement of Applicability 2.1 (Erklärung zur Anwendbarkeit) vom 11. Juli 2017. Die Einhaltung des ISO-Standards seitens Apple wurde von der British Standards Institution zertifiziert. Auf der BSI-Website findest du Zertifikate zur Einhaltung von ISO 27001 und ISO 27018. Diese Zertifikate kannst du auf folgenden Websites anzeigen:

<https://www.bsigroup.com/en-GB/our-services/certification/certificate-and-client-directory/search-results/?searchkey=company=apple&licencenumber=IS+649475>

<https://www.bsigroup.com/en-GB/our-services/certification/certificate-and-client-directory/search-results/?searchkey=company=Apple&licencenumber=PII%20673269>

Cryptographic Validation (FIPS 140-2)

Für die Verschlüsselungsmodule in iOS wurde – nach jedem Release seit iOS 6 – die Konformität mit dem U.S. Federal Information Processing Standard (FIPS) 140-2 Level 1 mehrmals geprüft und validiert. Wie bei jedem Hauptrelease übergibt Apple die Module zur erneuten Validierung an CMVP, wenn das iOS-Betriebssystem veröffentlicht wird. Mit dem Programm wird die Integrität bei Verschlüsselungsvorgängen in Apps von Apple und von anderen Anbietern validiert, die die Verschlüsselungsdienste und anerkannten Algorithmen von iOS ordnungsgemäß nutzen.

Common Criteria Certification (ISO 15408)

Seit dem Release von iOS 9 hat Apple iOS-Zertifizierungen unter dem Common Criteria Certification-Programm für jedes iOS-Hauptrelease für folgende Komponenten erhalten:

- Profil für grundlegenden Schutz von mobilen Geräten
- Profil zum Schutz von VPN-IPSec-Clients
- Erweitertes Paket für MDM-Agenten
- Erweitertes Paket für drahtlose LAN-Clients

iOS 11 enthält zusätzliche Zertifizierungen für folgende Komponenten:

- Profil zum Schutz von Anwendungssoftware
- Erweitertes Paket für E-Mail-Clients
- Erweitertes Paket für Internetbrowser

Apple plant, dies auch mit jedem folgenden Hauptrelease von iOS beizubehalten. Apple forciert innerhalb der ITC (International Technical Community) die Evaluierung grundlegender mobiler Sicherheitstechnologien mit dem Ziel, derzeit noch nicht verfügbare Schutzprofile (Collaborative Protection Profiles, cPPs) zu entwickeln. Apple wird auch in Zukunft Zertifizierungen auf der Basis neuer und aktualisierter Versionen der heute verfügbaren cPPs evaluieren und weiter vorantreiben.

Commercial Solutions for Classified (CSfC)

Sofern möglich hat Apple auch die iOS-Plattform und verschiedene Dienste zur Aufnahme in die Komponentenliste des CSfC-Programms (Commercial Solutions for Classified) eingereicht. Plattformen und Dienste von Apple, die im Rahmen des CCC-Programms geprüft werden, werden immer auch in die Liste der Komponenten für das CSfC-Programm aufgenommen.

Du findest die kürzlich aufgelisteten Komponenten unter:

<https://www.nsa.gov/resources/everyone/csfc/components-list/>

Handbücher zur Sicherheitskonfiguration

Apple hat in Zusammenarbeit mit Regierungsvertretern und staatlichen Behörden weltweit Handbücher erarbeitet, die unter dem Stichwort „Device Hardening for High-Risk Environments“ Anleitungen und Empfehlungen dafür enthalten, wie die Sicherheit einer Umgebung verbessert werden kann. Diese Handbücher enthalten sorgsam definierte und geprüfte Informationen darüber, wie integrierte Funktionen in iOS im Interesse eines verbesserten Schutzes und einer höheren Sicherheit konfiguriert und genutzt werden sollten und können.

Sicherheit in Apps

Apps sind die kritischen Elemente einer modernen 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 Apple-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 können Entwickler Frameworks in ihre Apps integrieren, die von der jeweiligen 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 Apple Developer Enterprise Program (ADEP) bewerben. Apple genehmigt Bewerbungsanträge nach Prüfung der Identität und Eignung der Bewerber. Tritt eine Organisation dem ADEP 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. Über MDM installierte Apps sind implizit vertrauenswürdig, da die Beziehung zwischen der Organisation und dem Gerät bereits etabliert ist. In allen anderen Fällen müssen die Benutzer das Bereitstellungsprofil der App in den Einstellungen bestätigen. Organisationen können es Benutzern verwehren, Apps zu bestätigen, die von unbekanntem Entwicklern stammen. Beim erstmaligen Starten einer Unternehmens-App muss das Gerät daher die positive Bestätigung durch Apple einholen, dass die App ausgeführt werden darf.

Anders als andere mobile Plattformen erlaubt iOS es seinen Benutzern nicht, potenziell schädliche, unsigned Apps von Websites zu installieren oder nicht vertrauenswürdigen Code auszuführen. Während der Laufzeit wird die Code-Signatur 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 Home-Verzeichnis für die dazugehörigen Dateien, das 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 System-App oder einen Daemon reduziert.

Außerdem können Apps nur über vom System bereitgestellte 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 System-Apps 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 einer 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 werden.

Eigene Tastaturen stellen eine Sonderform der Erweiterung dar, da sie vom Benutzer für das gesamte System aktiviert werden. Nach der Aktivierung wird eine Tastaturerweiterung für alle Textfelder verwendet, außer für die Eingabe des Codes und verschlüsselte Textfelder. Um die Übertragung von Benutzerdaten einzuschränken, werden eigene Tastaturen standardmäßig in einer besonders eingeschränkten Sandbox ausgeführt.

Diese Sandbox blockiert den Zugriff auf das Netzwerk, auf Dienste, die Netzwerkoperationen stellvertretend für den Prozess ausführen, und auf APIs, die es der Erweiterung erlauben würden, Eingabedaten aufzuzeichnen. 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 in einer MDM-Lösung registriert sind, folgen Dokument- und Tastaturerweiterungen der Regel „Verwaltetes Öffnen“ (Managed Open In). Die MDM-Lösung 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 eines bestimmten Entwickleraccounts können Inhalte gemeinsam verwenden, wenn sie als Teil einer App-Gruppe konfiguriert sind. Der Entwickler muss die entsprechenden Gruppen im Apple-Entwicklerportal erstellen und die gewünschte Gruppe mit Apps und Erweiterungen hinzufügen. Apps, die einer App-Gruppe zugewiesen wurden, haben Zugriff auf Folgendes:

- Einen geteilten Datenträgercontainer als Speicher, der auf dem Gerät bleibt, solange mindestens eine App aus der Gruppe installiert ist
- Geteilte Einstellungen
- Geteilte 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.

Die Mail-Datenbank (inklusive Anhänge), verwaltete Bücher, Safari-Lesezeichen, App-Start-Images und Ortsdaten werden ebenfalls in verschlüsselter Form gesichert; verwendet werden hierfür Schlüssel, 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, iPad and iPod touch“ (MFi) bietet geprüften Zubehörherstellern Zugriff auf das „iPod Accessories Protocol“ (iAP) und die notwendigen unterstützten Hardwarekomponenten.

Wenn ein MFi-Zubehör über einen Lightning Connector oder über 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 antwortet, das dann vom Gerät überprüft wird. Das Gerät sendet anschließend eine Anfrage, auf die das Zubehör eine signierte Antwort senden muss.

Dieses Verfahren wird vollständig von einem maßgeschneiderten integrierten Schaltkreis (Integrated Circuit, IC) 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 unterstützt, erhält es nur Zugriff auf analoge Audiosignale und in begrenztem Umfang auf serielle Audiowiedergabesteuerung (UART).

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 per 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 private 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 Werkeinstellungen zurückgesetzt wird, wird ein neues Schlüsselpaar generiert.

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 verschlüsselt, die auf für diese Sitzung erzeugten Curve25519-Schlüsseln basieren. Dies gilt sowohl für IP-basiertes Zubehör als auch für Bluetooth Low Energy-Zubehör.

Lokaler Datenspeicher

HomeKit speichert Daten über das Zuhause, 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, dort aber nicht verwendet. Da die Verschlüsselung mit Codes erfolgt, die nur auf den iOS-Geräten des Benutzers verfügbar sind, kann auf den Inhalt weder bei der Übertragung noch in iCloud zugegriffen werden.

Die HomeKit-Daten werden auch für mehrere Benutzer im selben Zuhause 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 Zuhause ein Benutzer hinzugefügt wird. Wurde dem Zuhause ein neuer Benutzer hinzugefügt, wird jede weitere Kommunikation mit dem Station-to-Station-Protokoll und für die Sitzung erzeugten Schlüsseln authentifiziert und verschlüsselt.

Der Benutzer, der die häusliche Umgebung in HomeKit angelegt hat, oder ein anderer Benutzer mit Bearbeitungsrechten kann neue Benutzer hinzufügen. Der Eigentümer des Geräts konfiguriert das Zubehör mit dem öffentlichen Schlüssel des neuen Benutzers, sodass das Zubehör den neuen Benutzer authentifizieren und Befehle von ihm empfangen kann. Wenn ein Benutzer mit Bearbeitungsrechten einen neuen Benutzer hinzufügt, wird der Vorgang an einen privaten Hub delegiert, der das Verfahren abschließt.

Die Bereitstellung des Apple TV für die Verwendung mit HomeKit erfolgt automatisch, wenn der Benutzer sich bei iCloud anmeldet. Für den iCloud-Account muss die Zwei-Faktor-Authentifizierung aktiviert sein. Das Apple TV und der Eigentümer des Geräts tauschen temporäre öffentliche Ed25519-Schlüssel via iCloud aus. Wenn sich der Eigentümer des Geräts und das Apple TV in demselben lokalen Netzwerk befinden, werden die temporären Schlüssel verwendet, um eine Verbindung über das lokale Netzwerk mittels Station-to-Station-Protokoll und für die Sitzung erzeugten Schlüsseln zu schützen. Dieses Verfahren verwendet dieselbe Authentifizierung und Verschlüsselung wie zwischen iOS-Geräten und HomeKit-Zubehör. Über diese sichere lokale Verbindung überträgt das Gerät des Eigentümers die öffentlich/privaten Ed25519-Schlüsselpaare des Benutzers an das Apple TV. Diese Schlüssel werden dann verwendet, um die Kommunikation zwischen dem Apple TV und dem HomeKit-Zubehör zu schützen und auch zwischen dem Apple TV und anderen iOS-Geräten, die Teil des HomeKit-Hauses sind.

Wenn ein Benutzer nicht mehrere Geräte verwendet und sich weigert, weiteren Benutzern Zugriff auf sein Zuhause zu gewähren, 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 unter folgender Adresse beschrieben werden: <https://developer.apple.com/homekit/>.

HomeKit und Siri

Siri kann verwendet werden, um Zubehör abzufragen und zu steuern und Szenen zu aktivieren. Minimale Informationen zur Konfiguration des Zuhauses werden anonym an Siri gesendet, um Namen von Zimmern, Zubehör oder Szenen bereitzustellen, die für die Erkennung von Befehlen erforderlich sind. Audiodaten, die an Siri gesendet werden, können bestimmtes Zubehör oder Befehle enthalten. Die Siri-Daten werden aber keiner anderen Apple-Funktion wie HomeKit zugeordnet. Weitere Informationen dazu findest du unter „Siri“ im Abschnitt „Internetdienste“ dieses Dokuments.

HomeKit-IP-Kameras

IP-Kameras in HomeKit senden Video- und Audiostreams direkt an das iOS-Gerät im lokalen Netzwerk, das auf den Stream zugreift. Die Streams werden auf dem iOS-Gerät und in der IP-Kamera mit zufällig generierten Schlüsseln verschlüsselt, die über die sichere HomeKit-Sitzung mit der Kamera ausgetauscht werden. Wenn sich das iOS-Gerät nicht im lokalen Netzwerk befindet, werden die verschlüsselten Streams über den privaten Hub an das iOS-Gerät weitergeleitet. Der private Hub entschlüsselt die Streams nicht, sondern funktioniert nur als Umleitung zwischen dem iOS-Gerät und der IP-Kamera. Wenn eine App dem Benutzer das Video der HomeKit-IP-Kamera anzeigt, gibt HomeKit die Videobilder sicher über einen separaten Systemprozess wieder, damit die App nicht auf den Videostream zugreifen oder diesen speichern kann. Außerdem dürfen Apps keine Bildschirmfotos aus diesem Stream speichern.

iCloud Remote-Zugriff für HomeKit-Zubehör

Das HomeKit-Zubehör kann die Verbindung zu iCloud direkt herstellen, sodass iOS-Geräte in der Lage sind, das Zubehör auch in Situationen zu kontrollieren, wenn keine Bluetooth- oder keine WLAN-Kommunikation möglich ist.

Beim Design des iCloud Remote-Zugriffs wurde sorgsam darauf geachtet, dass das Zubehör gesteuert werden kann und Benachrichtigungen gesendet werden können, ohne dass Apple gegenüber offen gelegt wird, um welche Art Zubehör es sich im Einzelfall handelt oder welche Befehle und Benachrichtigungen gesendet werden. HomeKit sendet beim iCloud Remote-Zugriff keine Informationen über die häusliche Umgebung.

Wenn ein Benutzer einen Befehl per iCloud Remote-Zugriff sendet, werden das Zubehör und das iOS-Gerät wechselseitig authentifiziert und die Daten auf die gleiche Weise verschlüsselt, wie dies bei lokalen Verbindungen geschieht. Der kommunizierte Inhalt wird verschlüsselt und ist für Apple nicht sichtbar. Die Adressierung über iCloud basiert auf den iCloud-Kennungen, die im Zuge des Konfigurationsprozesses registriert werden.

Zubehör, das den iCloud Remote-Zugriff unterstützt, wird beim jeweiligen Konfigurationsprozess bereitgestellt. Der Prozess der Bereitstellung beginnt in dem Moment, in dem sich der Benutzer bei iCloud anmeldet. Das iOS-Gerät sendet daraufhin eine Anfrage an das Zubehör, die das Zubehör signieren muss. Hierfür wird der Apple-Authentifizierungsprozessor (Authentication Coprocessor) verwendet, der in jedes „Built for HomeKit“-Zubehör eingebaut ist. Das Zubehör generiert außerdem ECC-Schlüssel (Elliptische-Kurven-Schlüssel) auf der Basis von prime256v1. Der öffentliche Schlüssel wird zusammen mit der signierten Anfrage und dem X.509-Zertifikat des Authentifizierungsprozessors an das iOS-Gerät gesendet. Diese Informationen werden verwendet, um vom iCloud-

Bereitstellungsserver ein Zertifikat für das Zubehör anzufordern. Das Zertifikat wird vom Zubehör gespeichert; es enthält aber keinerlei identifizierende Informationen über das Zubehör selbst, sondern nur die Information, dass das Zubehör die Zugriffsberechtigung für den iCloud Remote-Zugriff für HomeKit hat. Das iOS-Gerät, das die Bereitstellung übernimmt, sendet zusätzlich einen Behälter (Bag) an das Zubehör, der die URLs und die sonstigen Informationen enthält, die notwendig sind, um die Verbindung zum iCloud Remote Access Server herzustellen. Hierbei handelt es sich nicht um Informationen, die für einen Benutzer oder ein Zubehör spezifisch sind.

Für jedes Zubehör wird auf dem iCloud Remote Access Server die Liste der jeweils berechtigten Benutzer registriert. Dabei handelt es sich um die Benutzer, die von der Person, die das Zubehör der häuslichen Umgebung hinzugefügt hat, zum Steuern des Zubehörs berechtigt wurden. Diese Benutzer erhalten vom iCloud-Server eine Kennung und können einem iCloud-Account zugewiesen werden, damit Benachrichtigungen und Antworten vom Zubehör zugestellt werden können. Auf ähnliche Weise erhält auch das Zubehör von iCloud ausgestellte Kennungen; diese Kennungen sind aber „blickdicht“, das heißt, sie geben keinerlei Informationen über das Zubehör selbst preis.

Wenn ein Zubehör die Verbindung zum iCloud Remote Access Server für HomeKit herstellt, weist es sich mit einem Zertifikat und einer Karte (Pass) aus. Diese Karte stammt von einem anderen iCloud-Server und sie wird nicht für jedes Zubehör als eindeutiges, einzigartiges Attribut generiert. In die Anforderung auf die Zuteilung einer solchen Karte bindet das Zubehör den Namen des Herstellers, das Modell und die Firmware-Version ein. Diese Anforderung enthält aber keinerlei Informationen, die Rückschlüsse auf den Benutzer oder die häusliche Umgebung erlauben. Die Verbindung zu dem Server, der die Karten ausgibt, wird aus Gründen des Datenschutzes nicht authentifiziert.

Die Verbindung zwischen dem Zubehör und dem iCloud Remote Access Server erfolgt auf der Basis von HTTP/2 und ist abgesichert durch TLS 1.2 mit AES-128-GCM und SHA-256. Das Zubehör erhält seine Verbindung zum iCloud Remote Access Server aufrecht, um an es gerichtete Nachrichten empfangen und seinerseits Antworten und Benachrichtigungen an das iOS-Gerät senden zu können.

SiriKit

Siri nutzt den Mechanismus für iOS-Erweiterungen, um mit Apps anderer Anbieter zu kommunizieren. Siri hat zwar Zugriff auf iOS-Kontakte und den aktuellen Standort des Geräts, prüft aber dennoch die Berechtigung zum Zugriff auf geschützte iOS-Benutzerdaten der App, die die Erweiterung enthält, um festzustellen, ob die App Zugriff hat, bevor ihr diese Informationen bereitgestellt werden. Siri übergibt nur relevante Fragmente des Texts in der ursprünglichen Benutzeranfrage an die Erweiterung. Wenn die App keinen Zugriff auf iOS-Kontakte hat, würde Siri beispielsweise keine Beziehung in einer Benutzeranfrage auflösen wie „Sende meiner Mutter 10 Euro mit „Zahlungs-App““. In diesem Fall würde die App der Erweiterung nur „Mutter“ durch das Fragment der sprachlichen Äußerung sehen, das an sie übergeben wird. Hat die App jedoch Zugriff auf iOS-Kontakte, würde sie die iOS-Kontaktinformationen für die Mutter des Benutzers erhalten. Wenn im Haupttext der Nachricht auf einen Kontakt Bezug genommen wird, etwa „Erzähle meiner Mutter über „Nachrichten-App“, mein Bruder ist super“, würde Siri „mein Bruder“ nicht auflösen, unabhängig von den TTCs der App. Von der App präsentierte Inhalte können an den Server gesendet werden, damit Siri das von einem Benutzer in der App verwendete Vokabular versteht.

Bei Aussagen wie „Rufe mit <App-Name> ein Taxi für die Fahrt zum Haus meiner Mutter“, bei denen für die Anfrage Standortinformationen aus den Benutzerkontakten abgerufen werden müssen, stellt Siri diese Informationen der App-Erweiterung nur für diese Anfrage zur Verfügung, unabhängig vom Zugriff der App auf Orte oder Kontakte.

Siri erlaubt zur Laufzeit, dass die SiriKit-fähige App eine Reihe eigener Wörter bereitstellt, die für dieses Exemplar der App spezifisch sind. Diese eigenen Wörter sind mit der im Abschnitt „Siri“ in diesem Dokument beschriebenen zufälligen Kennung verknüpft und haben denselben Lebenszyklus.

HealthKit

HealthKit speichert und sammelt Daten von Gesundheits- und Fitness-Apps mit entsprechender Erlaubnis des Benutzers. HealthKit arbeitet direkt mit Gesundheit- und Fitnessgeräten zusammen, z. B. mit kompatiblen Bluetooth-Herzfrequenzmessgeräten und dem Motion-Coprozessor, der in viele iOS-Geräte integriert ist.

Gesundheitsdaten

HealthKit speichert und sammelt Gesundheitsdaten des Benutzers wie Größe, Gewicht, zurückgelegte Laufstrecken, Blutdruck usw. Diese Daten werden mit der Datenschutzklasse „Vollständiger Schutz“ gespeichert, sodass darauf nur zugegriffen werden kann, wenn der Benutzer seinen Code eingibt oder das Gerät mit Touch ID oder Face ID entsperrt.

HealthKit sammelt auch Verwaltungsdaten, wie Zugriffsrechte für Apps, Namen der mit HealthKit verbundenen Geräte und Planungsinformationen, die verwendet werden, um Apps zu starten, wenn neue Daten verfügbar werden. Diese Daten werden 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 die temporären Journal-Dateien in die Hauptdatenbank für Gesundheitsdaten importiert und anschließend gelöscht.

Gesundheitsdaten können in iCloud gespeichert werden. Wenn sie zum Speichern in iCloud konfiguriert sind, werden Gesundheitsdaten zwischen Geräten synchronisiert und durch Verschlüsselung gesichert, die die Daten bei der Übertragung und im Ruhezustand schützt. Gesundheitsdaten sind nur in verschlüsselten iTunes-Backups enthalten. In nicht verschlüsselten iTunes-Backups oder iCloud-Backups sind diese Daten nicht enthalten.

Datenintegrität

Die in der Datenbank gespeicherten Daten enthalten Metadaten, um die Herkunft der Datensätze zurückverfolgen zu können. Zu diesen Metadaten gehört 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 widerrufen.

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 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 nicht mit den Daten in den Gesundheitsdatenbanken synchronisiert.

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

ReplayKit

ReplayKit ist ein Framework, das es Entwicklern ermöglicht, ihre Apps mit Funktionen zur Aufzeichnung und Live-Übertragung auszustatten. Darüber hinaus ermöglicht es Benutzern, ihren Aufnahmen und Übertragungen mit der Frontkamera und dem Mikrofon des Geräts Kommentare hinzuzufügen.

Filmaufnahme

Das Aufnehmen eines Films umfasst verschiedene Sicherheitsstufen:

- **Berechtigungsfenster:** Vor dem Beginn der Aufnahme fordert ReplayKit den Benutzer in einem Hinweisfenster auf, zu bestätigen, dass der Bildschirm, das Mikrofon und die Frontkamera aufgezeichnet werden sollen. Dieser Hinweis erscheint einmal pro App-Prozess. Wenn die App länger als acht Minuten im Hintergrund bleibt, wird der Hinweis erneut angezeigt.
- **Bildschirm- und Audioaufzeichnung:** Bildschirm- und Audioaufzeichnungen erfolgen außerhalb des App-Prozesses im ReplayKit-Daemon *replayd*. Auf diese Weise ist gewährleistet, dass der aufgezeichnete Inhalt für den App-Prozess stets unzugänglich bleibt.
- **Filmerstellung und -sicherung:** Die Filmdatei wird in ein Verzeichnis geschrieben, das ausschließlich für ReplayKit-Subsysteme zugänglich ist und zu dem Apps keinerlei Zugang haben. Dadurch wird verhindert, dass Aufnahmen ohne Einwilligung des Benutzers von Dritten verwendet werden können.
- **Vorschau und Freigabe durch den Endbenutzer:** Mit der von ReplayKit verkauften Benutzeroberfläche kann der Benutzer den Film in einer Vorschau ansehen und ihn mit anderen teilen. Die Benutzeroberfläche wird außerhalb des Prozesses über die Infrastruktur für iOS-Erweiterungen angezeigt und hat Zugriff auf die generierte Filmdatei.

Übertragung

- **Bildschirm- und Audioaufzeichnung:** Der Mechanismus der Bildschirm- und Audioaufzeichnung während der Übertragung ist identisch mit der Filmaufnahme und erfolgt im *replayd*-Daemon.
- **Erweiterungen für die Übertragung:** Damit Dienste anderer Anbieter an der ReplayKit-Übertragung teilnehmen können, müssen sie zwei neue Erweiterungen erstellen, die mit dem Endpunkt `com.apple.broadcast-services` konfiguriert wurden:
 - Eine UI-Erweiterung, die es dem Benutzer ermöglicht, seine Übertragung einzurichten
 - Eine Upload-Erweiterung für das Hochladen von Video- und Audiodaten an die Back-End-Server des Dienstes

Durch die Architektur wird sichergestellt, dass die bereitstellenden Apps keinerlei Rechte an den übertragenen Video- und Audioinhalten haben – nur ReplayKit und die Broadcast-Erweiterungen des anderen Anbieters erhalten Zugriff.

- **Auswahl des Übertragungsdienstes:** Für die Auswahl des zu verwendenden Übertragungsdienstes stellt ReplayKit einen View-Controller (ähnlich dem `UIActivityViewController`) bereit, den der Entwickler in seiner App präsentieren kann. Der View-Controller wird mit der `UIRemoteViewController` SPI implementiert und ist eine Erweiterung, die sich innerhalb des ReplayKit-Frameworks befindet. Er läuft außerhalb des Prozesses der bereitstellenden App.
- **Upload-Erweiterung:** Die Upload-Erweiterung wird von Übertragungsdiensten anderer Anbieter implementiert, um die Video- und Audioinhalte während der Übertragung zu verarbeiten. Diese Erweiterung kann zwischen zwei Optionen für den Empfang der Inhalte wählen:
 - Kleine codierte MP4-Clips
 - Uncodierte RAW-Sample-Puffer
 - **Handhabung von MP4-Clips.** Bei diesem Handhabungsmodus werden vom *replayd*-Daemon kleine codierte MP4-Clips generiert und an einem privaten Speicherort abgelegt, der nur für die ReplayKit-Subsysteme zugänglich ist. Nachdem ein Filmclip generiert wurde, gibt der *replayd*-Daemon seine Speicherposition über die `NSExtension-Anfrage` SPI (XPC-basiert) an die Upload-Erweiterung des anderen Anbieters weiter. Der *replayd*-Daemon generiert außerdem ein einmal nutzbares `Sandbox-Token`, das ebenfalls an die Upload-Erweiterung weitergegeben wird und der Erweiterung während der Erweiterungsanfrage Zugriff auf diesen bestimmten Filmclip gewährt.
 - **Handhabung des Sample-Puffers.** Während dieses Handhabungsmodus werden Video- und Audiodaten serialisiert und über eine direkte XPC-Verbindung in Echtzeit an die Upload-Erweiterung des anderen Anbieters übergeben. Die Videodaten werden codiert, indem das `IOSurface`-Objekt aus dem Video-Sample-Puffer extrahiert, als XPC-Objekt sicher codiert, via XPC an die Erweiterung des anderen Anbieters gesendet und dann wieder sicher in ein `IOSurface`-Objekt decodiert wird.

Geschützte Notizen

Die App „Notizen“ bietet eine Funktion für geschützte Notizen, mit der Benutzer den Inhalt bestimmter Notizen schützen können. Geschützte Notizen werden mit einem vom Benutzer bereitgestellten Passwort verschlüsselt, das zum Anzeigen der Notizen in iOS, macOS und auf der iCloud-Website erforderlich ist.

Beim Schützen einer Notiz wird aus dem Passwort des Benutzers mittels PBKDF2 und SHA256 ein 16-Byte-Schlüssel abgeleitet. Der Inhalt der Notiz wird mit AES-GCM verschlüsselt. Neue Einträge werden in Core Data und CloudKit erstellt, um die verschlüsselte Notiz, das Tag und den Initialisierungsvektor zu speichern, und die Originaleinträge der Notiz werden gelöscht. Die verschlüsselten Daten werden nicht an gleicher Stelle geschrieben. Anhänge werden in gleicher Weise verschlüsselt. Zu den unterstützten Anhängen gehören Bilder, Skizzen, Tabellen, Karten und Websites. Notizen mit anderen Arten von Anhängen können nicht verschlüsselt werden. Es ist außerdem nicht möglich, zu geschützten Notizen Anhangsformate hinzuzufügen, die nicht unterstützt werden.

Wenn ein Benutzer zum Anzeigen oder zum Erstellen einer geschützten Notiz das Passwort erfolgreich eingibt, öffnet die Notizen-App eine sichere Sitzung. Solange diese Sitzung aktiv ist, muss der Benutzer das Passwort nicht eingeben oder Touch ID bzw. Face ID verwenden, um andere Notizen anzuzeigen oder zu schützen. Falls einige Notizen ein anderes Passwort verwenden, gilt die sichere Sitzung nur für die Notizen, die mit dem aktuellen Passwort geschützt sind. Die sichere Sitzung wird in folgenden Situationen geschlossen:

- Der Benutzer tippt in der App „Notizen“ auf die Taste „Jetzt sperren“.
- Die App „Notizen“ wird länger als drei Minuten in den Hintergrund geschoben.
- Das Gerät wird gesperrt.

Benutzer, die das Passwort vergessen haben, können geschützte Notizen dennoch anzeigen oder weitere Notizen schützen, wenn sie Touch ID oder Face ID auf ihren Geräten aktiviert haben. Nach drei erfolglosen Eingabeversuchen zeigt die Notizen-App zudem eine vom Benutzer bereitgestellte Passwort-Merkhilfe an. Der Benutzer muss das aktuelle Passwort kennen, um es ändern zu können.

Benutzer können das Passwort zurücksetzen, falls sie das aktuelle Passwort vergessen haben. Dadurch können sie zwar neue geschützte Notizen mit einem neuen Passwort erstellen, aber keine zuvor geschützten Notizen sehen. Zuvor geschützte Notizen können weiterhin angezeigt werden, wenn sich der Benutzer wieder an das alte Passwort erinnert. Zum Zurücksetzen des Passworts ist das Passwort für den iCloud-Account des Benutzers erforderlich.

Geteilte Notizen

Notizen können mit anderen geteilt werden. Geteilte Notizen sind nicht durchgehend (End-to-End) verschlüsselt. Apple verwendet den mit CloudKit verschlüsselten Datentyp für Texte oder Anhänge, die der Benutzer in einer Notiz hinzufügt. Medien werden immer mit einem Schlüssel aus dem Datensatz CKRecord verschlüsselt. Metadaten wie Erstellungs- und Änderungsdatum werden nicht verschlüsselt. CloudKit verwaltet den Prozess, durch den Teilnehmer die Daten der jeweils anderen Teilnehmer verschlüsseln/entschlüsseln können.

Apple Watch

Apple Watch verwendet für iOS entwickelte Sicherheitsfunktionen und -technologien zum Schutz der Daten auf dem Gerät und der Kommunikation mit einem gekoppelten 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 (siehe Beschreibung im Abschnitt „iMessage“ in diesem Dokument). 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 Bluetooth-, WLAN- und Mobilfunk-Links eine zusätzliche Verschlüsselungsebene bereitstellen. Das Abrufen der Schlüssel erfolgt in 15-Minuten-Intervallen, um das Zeitfenster für den Fall einzugrenzen, dass der Datenverkehr kompromittiert wird.

Zur Unterstützung von Apps, die Daten streamen, erfolgt die Verschlüsselung mithilfe der Methoden, die unter „FaceTime“ im Abschnitt „Internetdienste“ dieses Dokuments beschrieben werden. Hierzu wird der vom gekoppelten iPhone bereitgestellte IDS-Dienst oder eine direkte Internetverbindung verwendet.

Die Apple Watch implementiert eine hardwarebasierte Speicherverschlüsselung und einen klassenbasierten Schutz von Dateien und Schlüsselbundobjekten. Näheres hierzu findest du im Abschnitt über die Verschlüsselung und 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.

Wenn sich die Apple Watch außerhalb der Bluetooth-Reichweite befindet, kann alternativ eine WLAN- oder Mobilfunk-Verbindung verwendet werden. Die Apple Watch stellt erst dann eine Verbindung zu WLANs her, wenn die entsprechenden Anmeldedaten (die zuvor mit der Apple Watch synchronisiert wurden) auf dem gekoppelten iPhone vorhanden sind. Befindet sich die Apple Watch außerhalb der iPhone-Reichweite, stehen auf dem iPhone verfügbare neue Netzwerkanmeldedaten auf der Apple Watch nicht zur Verfügung.

Die Apple Watch kann durch Drücken der Seitentaste manuell gesperrt werden. Darüber hinaus wird Bewegungsheuristik verwendet, um das Gerät kurz nach dem Entfernen vom Handgelenk automatisch zu sperren. Wenn die Apple Watch gesperrt ist, kann Apple Pay nur verwendet werden, nachdem der Code der Uhr eingegeben wurde. Die Handgelenkerkennung wird mit der App „Apple Watch“ auf dem iPhone ausgeschaltet. Diese Einstellung kann mithilfe einer MDM-Lösung (Mobile Device Management) erzwungen werden.

Sofern die Uhr getragen wird, kann sie auch mit dem gekoppelten iPhone entsperrt werden. Dies geschieht, indem eine Verbindung hergestellt wird, die durch die beim Koppeln 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.

Die Apple Watch kann jeweils nur mit einem iPhone gekoppelt werden. Das iPhone kommuniziert Anleitungen zum Löschen aller Inhalte und Daten von der Apple Watch, wenn die Koppelung aufgehoben wird.

Wird „Mein iPhone suchen“ auf dem gekoppelten iPhone eingeschaltet, kann automatisch auch die Aktivierungssperre auf der Apple Watch verwendet werden. Die Aktivierungssperre erschwert unbefugten Personen den Verkauf oder die Verwendung einer Apple Watch, die verloren oder gestohlen wurde. Die Aktivierungssperre bewirkt, dass die Apple-ID des Benutzers und das Passwort eingegeben werden müssen, um die Kopplung aufzuheben, die Daten zu löschen oder die Apple Watch neu zu aktivieren.

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 nutzt bewährte Technologien und die aktuellen Standards sowohl für WLAN als auch für Mobilfunk-Datenverbindungen, 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 die Beschränkung der Listening-Ports und das Entfernen unnötiger Netzwerkdienstprogramme wie Telnet, Shells oder Webserver reduziert wird.

TLS

iOS unterstützt Transport Layer Security (TLS 1.0, TLS 1.1, TLS 1.2) und DTLS. Es unterstützt AES-128 und AES-256 und bevorzugt Cipher-Suites mit perfekter Forward Secrecy. Safari, Kalender, Mail und weitere Internetprogramme verwenden dieses Protokoll 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. CFNetwork lässt SSLv3 nicht zu, und Apps wie Safari, die vom WebKit Gebrauch machen, ist es verwehrt, eine SSLv3-Verbindung herzustellen.

Ab iOS 11 und macOS High Sierra können SHA-1-Zertifikate nicht mehr für TLS-Verbindungen verwendet werden, wenn der Benutzer sie nicht als vertrauenswürdig einstuft. Zertifikate mit RSA-Schlüsseln mit weniger als 2048 Bit sind ebenfalls nicht zulässig. Die RC4-Cipher-Suite für die symmetrische Verschlüsselung findet in iOS 10 und macOS Sierra keine Anwendung. Standardmäßig sind RC4-Cipher-Suites auf TLS-Clients oder -Servern, die mit SecureTransport APIs implementiert wurden, nicht aktiviert und können keine Verbindung herstellen, wenn es sich bei RC4 um die einzige verfügbare Cipher-Suite handelt. Dienste oder Apps, die RC4 erfordern, sollten für die Verwendung von modernen, sicheren Cipher-Suites aktualisiert werden, um ein höheres Maß an Sicherheit zu erzielen.

Transportsicherheit für Apps

App Transport Security umfasst Anforderungen für Standardverbindungen, um bei Verwendung von NSURLConnection-, CFURL- oder NSURLSession-APIs sicherzustellen, dass von den Apps die besten Verfahren für sichere Verbindungen eingehalten werden. Standardmäßig beschränkt App Transport Security die Cipher-Auswahl ausschließlich auf solche Cipher-Suites, die Forward Secrecy bereitstellen, insbesondere ECDHE_ECDSA_AES und ECDHE_RSA_AES im GCM- oder CBC-Modus. Apps sind in der Lage, die Forward Secrecy-Erfordernis pro Domain zu deaktivieren. In diesem Fall wird RSA_AES zur Gruppe der verfügbaren Ciphers hinzugefügt.

Server müssen TLS 1.2 und Forward Secrecy unterstützen und Zertifikate müssen auf der Basis von SHA-256 (oder neuer) validiert und signiert sein, wobei ein 2048-Bit-RSA-Schlüssel oder ein 256-Bit-ECC-Schlüssel (elliptischer Kurven-Schlüssel) die Mindestanforderung darstellt.

Netzwerkverbindungen, die diese Anforderungen nicht erfüllen, kommen nicht zustande, es sei denn, die App setzt die in App Transport Security definierten Anforderungen für die Transportsicherheit außer Kraft. Ungültige Zertifikate führen in jedem Fall zu einem nicht behebbaren Fehler (Hard Failure) und zum Nichtzustandekommen der Verbindung. App Transport Security wird automatisch auf alle Apps angewendet, die für iOS 9 (oder neuer) kompiliert werden.

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:

- IKEv2/IPSec mit Authentifizierung per Shared Secret, RSA-Zertifikate, ECDSA-Zertifikate, EAP-MSCHAPv2 oder EAP-TLS
- SSL-VPN unter Verwendung der entsprechenden Client-App aus dem App Store
- Cisco IPSec mit Benutzerauthentifizierung per Passwort, RSA SecurID oder CRYPTOCARD, sowie die Systemauthentifizierung über gemeinsame geheime Schlüssel (Shared Secret) und Zertifikate
- L2TP/IPSec mit Benutzerauthentifizierung per MS-CHAPv2-Passwort, RSA SecurID oder CRYPTOCARD, sowie die Systemauthentifizierung über gemeinsame geheime Schlüssel (Shared Secret)

iOS unterstützt Folgendes:

- **VPN On Demand** für Netzwerke, die eine zertifikatsbasierte Authentifizierung verwenden. IT-Richtlinien legen über das verwendete VPN-Konfigurationsprofil fest, für welche Domains eine VPN-Verbindung benötigt wird.
- **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 spezielle Domains in Safari bestimmt werden. So lässt sich sicherstellen, dass nur sichere Daten und keine privaten Daten des Benutzers in das Unternehmensnetzwerk hinein- und daraus hinausgelangen.
- Die Funktion „**VPN immer eingeschaltet**“, die für Geräte konfiguriert werden kann, die über MDM verwaltet und mit Apple Configurator 2, dem Programm zur Geräteregistrierung (DEP) oder mit Apple School Manager betreut werden. Auf diese Weise müssen die Benutzer die Sicherung über VPN nicht manuell aktivieren, wenn sie sich mit Mobilfunk- oder 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 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.

Neben dem Schutz für Daten erweitert iOS den Schutz auf WPA2-Ebene auf Unicast- und Multicast-Management-Frames über den PMF-Service (Protected Management Frame), auf den in 802.11w Bezug genommen wird. PMF-Unterstützung ist auf dem iPhone 6 und iPad Air 2 (oder neuer) verfügbar.

iOS verwendet eine zufällige MAC-Adresse (Media Access Control) für WLAN-Scans, wenn keine Verbindung zu einem WLAN-Netzwerk besteht. Solche Scans könnten ausgeführt werden, um ein bevorzugtes WLAN-Netzwerk zu finden und eine Verbindung dazu herzustellen oder um Ortungsdienste für Apps zu unterstützen, die Geofences verwenden, etwa ortsbasierte Erinnerungen, oder zum Fixieren eines Standorts in der Apple-App „Karten“. Zu beachten ist, dass WLAN-Scans, die bei einem Verbindungsversuch zu einem bevorzugten WLAN-Netzwerk erfolgen, nicht zufällig sind.

iOS 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 verwendet, die Geofences nutzen, 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 von einem WLAN getrennt wird, 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. Apple hat WLAN-Hersteller darüber informiert, dass iOS-WLAN-Scans eine zufällige MAC-Adresse verwenden und dass weder Apple noch der jeweilige Hersteller Aussagen darüber treffen können, welche zufälligen MAC-Adressen verwendet werden. Das iPhone 4s (oder früher) bietet keine Unterstützung für die Verwendung zufälliger MAC-Adressen in WLANs.

Auf dem iPhone 6s (oder neuer) ist das verdeckte Eigentum eines WLAN-Netzwerks bekannt und wird automatisch aktualisiert. Wenn die SSID (Service Set Identifier) eines WLAN-Netzwerks übertragen wird, sendet das iOS-Gerät mit der in der Anfrage enthaltenen SSID keinen Test mit. Dadurch wird verhindert, dass das Gerät den Netzwerknamen von nicht versteckten Netzwerken sendet.

Um das Gerät vor Lücken in der Firmware von Netzwerkprozessoren zu schützen, haben Netzwerkschnittstellen (einschließlich WLAN und Baseband) eingeschränkten Zugriff auf den Speicher des Anwendungsprozessors. Wenn USB oder SDIO als Schnittstelle zum Netzwerkprozessor genutzt wird, kann der Netzwerkprozessor keine DMA-Transaktionen (Direct Memory Access) mit dem Anwendungsprozessor starten. Wenn PCIe verwendet wird, befindet sich jeder Netzwerkprozessor auf einem eigenen isolierten PCIe-Bus. Der DMA-Zugriff des Netzwerkprozessors auf jedem PCIe-Bus wird durch eine IOMMU (I/O Memory Management Unit) auf Seiten im Speicher eingeschränkt, die Netzwerkpakete und Kontrollstrukturen enthalten.

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 1.5)
- Phone Book Access Profile (PBAP)
- Message Access Profile (MAP)
- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP)
- Personal Area Network Profile (PAN)
- Human Interface Device Profile (HID)
- Die Unterstützung für diese Profile hängt vom jeweiligen Gerät ab.

Weitere Informationen findest du unter:

<https://support.apple.com/de-de/HT204387>

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 zertifikatsbasierte Authentifizierung (PKINIT) wird ebenfalls unterstützt.

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 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 von SSO unterstützt iOS ein neues Konfigurationsprofil, das es MDM-Lösungen 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 und 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 Informationen auf Geräte in der Nähe zu senden, zu denen auch Mac-Computer mit AirDrop-Unterstützung und OS X 10.11 (oder neuer) 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 in AirDrop aber 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. Wenn aber 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.

Für Geräte und Apps, die mit einer Lösung für das Mobile Device Management (MDM) verwaltet werden, kann die Verwendung von AirDrop eingeschränkt werden.

WLAN-Passwortfreigabe

iOS-Geräte, die die WLAN-Passwortfreigabe unterstützen, verwenden ein AirDrop ähnliches Verfahren, um ein WLAN-Passwort zwischen zwei Geräten zu senden.

Wenn ein Benutzer ein WLAN-Netzwerk (Antragsteller) auswählt und nach dem WLAN-Passwort gefragt wird, startet das Apple-Gerät eine Bluetooth Low Energy-Ankündigung, die angibt, dass ein WLAN-Passwort erforderlich ist. Andere aktive Apple-Geräte in der Nähe, die das Passwort für das ausgewählte WLAN-Netzwerk haben, stellen eine Bluetooth Low Energy-Verbindung mit dem anfordernden Gerät her.

Das Gerät mit dem WLAN-Passwort (Aussteller) fordert die Kontaktinformationen des Antragstellers an und dieser muss seine Identität mit einem AirDrop ähnlichen Verfahren nachweisen. Nachdem die Identität nachgewiesen wurde, sendet der Aussteller dem Antragsteller den sicheren privaten Schlüssel mit 64 Zeichen, der unter anderem zum Beitritt zum Netzwerk verwendet werden kann.

Für Geräte und Apps, die mit einer Lösung für das Mobile Device Management (MDM) verwaltet werden, kann die Verwendung der WLAN-Passwortfreigabe eingeschränkt werden.

Apple Pay

Mit Apple Pay können Benutzer mit einem der unterstützten iOS-Geräte und mit der Apple Watch schnell, sicher und geheim in Stores, Apps und im Web in Safari bezahlen. Dies ist nicht nur praktisch, sondern bietet auch in Hard- und Software integrierte Sicherheit.

Apple Pay ist auch dafür ausgelegt, die persönlichen Daten des Nutzers zu schützen. Apple Pay sammelt keine Informationen über Transaktionen, die mit dem Benutzer verknüpft werden können. Die Transaktion findet zwischen dem Benutzer, dem Händler und dem Kartenaussteller statt.

Apple Pay-Komponenten

Secure Element: Secure Element ist ein zertifizierter Chip, der die Branchenstandards erfüllt und auf dem die Java Card-Plattform ausgeführt wird, die mit den Anforderungen für elektronische Zahlungen der Finanzbranche kompatibel ist.

NFC-Controller: Der NFC-Controller verarbeitet die NFC-Protokolle (Near Field Communication) und steuert die Kommunikation zwischen Anwendungsprozessor und Secure Element bzw. zwischen Secure Element und Kassenterminal.

Wallet: Wallet wird verwendet, um Kredit-, Debit-, Kunden- und Bonuskarten hinzuzufügen und zu verwalten und Zahlungen mit Apple Pay abzuwickeln. Der Benutzer kann in Wallet seine Karten, Zusatzinformationen über den Kartenaussteller und dessen Datenschutzrichtlinien, die letzten Zahlungen und weitere Informationen anzeigen lassen. Außerdem können im Systemassistenten und in den Einstellungen Karten für Apple Pay hinzugefügt werden.

Secure Enclave: Auf einem iPhone, einem iPad und einer Apple Watch führt die Secure Enclave das Authentifizierungsverfahren durch und ermöglicht die Abwicklung der Zahlung.

Die Apple Watch muss entsperrt sein und der Benutzer muss die Seitentaste doppelklicken. Der Doppelklick wird erkannt und direkt ohne Umweg über den Anwendungsprozessor an das Secure Element oder die Secure Enclave (sofern verfügbar) weitergeleitet.

Apple Pay-Server: Die Apple Pay-Server verwalten die Konfiguration und Bereitstellung der Kredit- und Debitkarten in Wallet und die im Secure Element gespeicherten Device Account Numbers. Sie kommunizieren sowohl mit dem Gerät als auch mit den Zahlungsservern im Netzwerk. Die Apple Pay Server sind auch für die Neuverschlüsselung der Zahlungsdaten in den Apps zuständig.

So nutzt Apple Pay das Secure Element

Das Secure Element nutzt ein extra entwickeltes Applet zur Verwaltung von Apple Pay. Außerdem besitzt es Zahlungs-Applets, die von den Zahlungsnetzwerken zertifiziert werden. Die Daten der Kredit-, Debit- oder Prepaid-Karte werden vom Zahlungsnetzwerk oder dem Kartenaussteller in verschlüsselter Form an diese Zahlungs-Applets gesendet, wobei Schlüssel verwendet werden, die nur das Zahlungsnetzwerk und die Sicherheits-Domain des Applets kennen. Diese Daten werden in den Zahlungs-Applets gespeichert und mit den Sicherheitsfunktionen des Secure Element geschützt. Bei einer Transaktion kommuniziert das Terminal über den NFC-Controller und einen eigenen Hardware-Bus direkt mit dem Secure Element.

So nutzt Apple Pay den NFC-Controller

Als Tor zum Secure Element sorgt der NFC-Controller dafür, dass alle kontaktlosen Zahlungen über ein Kassenterminal in unmittelbarer Nähe des Geräts durchgeführt werden. Nur Zahlungsanfragen von Terminals in unmittelbarer Nähe werden von NFC-Controller als kontaktlose Zahlungen markiert.

Nachdem die Zahlung vom Besitzer der Karte per Touch ID oder Code oder auf einer entsperrten Apple Watch durch Doppelklicken auf die Seitentaste in der Secure Enclave autorisiert wurde, werden die kontaktlosen Antworten der Zahlungs-Applets im Secure Element über den Controller exklusiv an das NFC-Feld geleitet. Daher bleiben die Daten der Zahlungsanweisung für kontaktlose Transaktionen im lokalen NFC-Feld und haben keinen Kontakt zum Anwendungsprozessor. Die Daten von Zahlungsanweisungen in Apps und im Web werden dagegen über den Anwendungsprozessor an den Apple Pay Server gesendet, allerdings nach Verschlüsselung durch das Secure Element.

Bereitstellen von Kredit-, Debit- und Prepaid-Karten

Wenn der Benutzer in Apple Pay eine Kredit-, Debit- oder Prepaid-Karte (Kundenkarten inbegriffen) hinzufügt, sendet Apple die Kartendaten zusammen mit anderen Informationen über den Benutzeraccount und das Gerät über eine sichere Verbindung an den jeweiligen Kartenaussteller oder den autorisierten Dienstleister des Kartenausstellers. Anhand dieser Informationen bestimmt der Kartenaussteller, ob die Karte für Apple Pay hinzugefügt werden kann.

Apple Pay verwendet zum Zweck der Kartenbereitstellung drei serverseitige Aufrufe, um mit dem Kartenaussteller oder dem Netzwerk zu kommunizieren: *Required Fields*, *Check Card* und *Link and Provision*. Der Kartenaussteller bzw. das Netzwerk verwenden diese Aufrufe, um Karten für Apple Pay zu verifizieren, zu bestätigen und hinzuzufügen. Diese Client-Server-Sitzungen werden mit TLS v1.2 verschlüsselt.

Die vollständige Kartennummer wird weder auf dem Gerät noch auf den Servern von Apple gespeichert. Stattdessen wird eine eindeutige Device Account Number erstellt, verschlüsselt und im Secure Element gespeichert. Diese eindeutige Device Account Number wird so verschlüsselt, dass Apple nicht darauf zugreifen kann. Diese Device Account Number ist eindeutig und unterscheidet sich von anderen Kredit- oder Debitkartennummern darin, dass der Kartenaussteller verhindern kann, dass sie auf einer Magnetkarte, am Telefon oder auf Websites verwendet wird. Die Device Account Number im Secure Element ist von iOS und watchOS isoliert, wird nie auf den Apple-Servern gespeichert und nie in iCloud gesichert.

Karten zur Verwendung mit der Apple Watch werden für Apple Pay mit der App „Apple Watch“ auf dem iPhone aktiviert. Um eine Karte für die Apple Watch vorzubereiten, muss sich die Uhr im Bluetooth-Kommunikationsradius befinden. Karten, die speziell für die Verwendung mit der Apple Watch aktiviert wurden, haben eine eigene Device Account Number, die im Secure Element auf der Apple Watch gespeichert ist. Es gibt drei Möglichkeiten, eine Kredit-, Debit- oder Prepaid-Karte in Apple Pay bereitzustellen:

- Karte manuell zu Apple Pay hinzuzufügen
- Kredit- oder Debitkarten, die für einen iTunes Store Account registriert wurden und verwendet werden, zu Apple Pay hinzuzufügen
- Karten aus der App des Kartenausstellers zu Apple Pay hinzuzufügen

Kredit- oder Debitkarte manuell zu Apple Pay hinzuzufügen

Beim manuellen Hinzufügen einer Karte (Kundenkarten inbegriffen) werden für den Bereitstellungsprozess der Name des Kartenhalters, die Kreditkartennummer, das Ablaufdatum und die CVV-Nummer verwendet. Der Benutzer kann die Informationen in den Einstellungen, in der App „Wallet“ oder in der App „Apple Watch“ eintippen oder mithilfe der Kamera des Geräts erfassen. Wenn die Kamera die Karteninformationen erfasst, versucht Apple, Name, Kartennummer und Ablaufdatum einzugeben. Das Foto wird nicht auf dem Gerät oder in der Foto-Mediathek gespeichert. Nachdem alle Felder ausgefüllt wurden, überprüft der „Check Card“-Prozess alle Felder mit Ausnahme der CVV-Nummer. Sie werden verschlüsselt und an den Apple Pay-Server gesendet.

Wenn der „Check Card“-Prozess eine ID für die Geschäftsbedingungen umfasst, lädt Apple die Geschäftsbedingungen des jeweiligen Kartenausstellers und zeigt sie dem Benutzer an. Wenn der Benutzer die Geschäftsbedingungen annimmt, übergibt Apple die ID der Geschäftsbedingungen sowie die CVV an den „Link and Provision“-Prozess. Apple stellt als Teil des „Link and Provision“-Prozesses auch Informationen über das Gerät für den Kartenaussteller oder das Netzwerk bereit: u. a. Informationen über Aktivitäten deines iTunes- und App Store-Accounts (z. B. die Anzahl der Transaktionen in iTunes), Informationen über das Gerät (z. B. Telefonnummer, Name und Modell des Geräts sowie weitere für das Einrichten von Apple Pay erforderliche iOS-Geräte) und deinen ungefähren Standort zu der Zeit, zu der du die Karte hinzugefügt hast, sofern du die Ortungsdienste aktiviert hast. Anhand dieser Informationen bestimmt der Kartenaussteller, ob die Karte für Apple Pay hinzugefügt werden kann.

Der Prozess „Link and Provision“ löst zwei Ereignisse aus:

- Das Gerät beginnt mit dem Laden der Wallet-Kartendatei für die jeweilige Kredit- bzw. Debitkarte.
- Das Gerät bindet die Karte an das Secure Element.

Die Kartendatei enthält URLs zum Laden von Kartengrafiken, Metadaten zur Karte wie Kontaktinformationen, zugehörige App des Kartenausstellers und unterstützte Funktionen. Außerdem enthält sie den Kartenstatus, der Informationen darüber enthält, ob das Secure Element vollständig personalisiert wurde, ob die Karte aktuell durch den Kartenaussteller gesperrt ist oder ob eine zusätzliche Verifizierung nötig ist, bevor in Apple Pay mit dieser Karte gezahlt werden kann.

Kredit- oder Debitkarten, die für einen iTunes Store Account verwendet werden, zu Apple Pay hinzufügen

Bei Kredit- oder Debitkarten, die bereits für iTunes verwendet werden, muss der Benutzer das Passwort für seine Apple-ID erneut eingeben. Die Kreditkartennummer wird von iTunes abgerufen und der Prozess „Check Card“ wird gestartet. Wenn die Karte für Apple Pay verwendet werden kann, lädt das Gerät die Geschäftsbedingungen, zeigt sie an und übergibt die dazugehörige ID und den Kartensicherheitscode an den Prozess „Link and Provision“. Für im iTunes-Account vorhandene Karten werden möglicherweise zusätzliche Überprüfungen durchgeführt.

Kredit- oder Debitkarten aus der App des Kartenausstellers zu Apple Pay hinzufügen

Wenn eine App für die Verwendung mit Apple Pay registriert wird, werden Schlüssel für die App und den Server des Händlers erstellt. Diese Schlüssel werden verwendet, um die Karteninformationen zu verschlüsseln, die an den Händler gesendet werden. Auf diese Weise wird verhindert, dass diese Informationen vom iOS-Gerät gelesen werden. Der Datenfluss für die Bereitstellung ähnelt dem Datenfluss beim manuellen Hinzufügen von Karten (siehe oben). Die einzige Abweichung besteht darin, dass anstelle der CVV-Nummer ein nur einmal nutzbares Passwort verwendet wird.

Zusätzliche Überprüfung

Der Kartenaussteller kann festlegen, ob für eine Kredit- oder Debitkarte eine zusätzliche Überprüfung notwendig ist. Der Benutzer kann für die Überprüfung aus verschiedenen Methoden wählen, abhängig davon, welche vom Kartenaussteller angeboten werden, z. B. SMS, E-Mail, Anruf beim Kundendienst oder über eine App eines anderen Anbieters. Bei SMS- oder E-Mail-Nachrichten kann der Benutzer aus den beim Kartenaussteller hinterlegten Kontaktdaten auswählen. Der Benutzer erhält einen Code, der in der App „Wallet“, in den Einstellungen oder in der App „Apple Watch“ eingegeben werden muss. Beim Kundendienst oder bei der Überprüfung über eine App führt der Aussteller einen eigenen Kommunikationsprozess durch.

Autorisierung von Zahlungen

Auf Geräten mit einer Secure Enclave lässt das Secure Element eine Zahlung erst dann zu, wenn es die Autorisierung von der Secure Enclave erhalten hat. Auf dem iPhone oder iPad erfordert dies die Bestätigung, dass sich der Benutzer per Touch ID, Face ID oder Gerätecode authentifiziert hat. Die Standardmethode ist Touch ID oder Face ID, es kann aber auch jederzeit der Code verwendet werden. Der Code wird automatisch angeboten, wenn dreimal in Folge kein registrierter Fingerabdruck erkannt wurde oder zwei Versuche zum Gesichtsabgleich nicht erfolgreich waren; nach fünf Fehlversuchen muss der Code eingegeben werden. Der Code wird auch benötigt, wenn Touch ID oder Face ID nicht konfiguriert oder nicht für Apple Pay aktiviert wurde. Damit auf der Apple Watch eine Zahlung erfolgen kann, muss der Benutzer das Gerät mittels Code entsperren und einen Doppelklick auf die Seitentaste ausführen.

Die Kommunikation zwischen der Secure Enclave und dem Secure Element findet über eine serielle Schnittstelle statt, wobei das Secure Element mit dem NFC-Controller verbunden ist, der wiederum mit dem Anwendungsprozessor verbunden ist. Obwohl sie nicht direkt verbunden sind, können Secure Enclave und Secure Element über ein gemeinsames Schlüsselpaar, das bei der Fertigung festgelegt wird, sicher kommunizieren. Die Verschlüsselung und die Authentifizierung der Kommunikation basieren auf AES, wobei beide Seiten zum Schutz vor Replay-Angriffen kryptografische Nonces verwenden. Der Kopplungsschlüssel wird in der Secure Enclave aus dem UID Schlüssel und der eindeutigen Kennung des Secure Element erzeugt. Der Kopplungsschlüssel wird dann bei der Fertigung von der Secure Enclave sicher an

ein Hardwaresicherheitsmodul (HSM) übertragen, das die benötigten Schlüsselmaterialien besitzt, um den Kopplungsschlüssel mit dem Secure Element zu verknüpfen.

Wenn der Benutzer eine Transaktion autorisiert, sendet die Secure Enclave signierte Daten über die Art der Authentifizierung und Details zu der Transaktion (kontaktlos oder in Apps) zusammen mit einem Zufallswert für die Authentifizierung (AR) an das Secure Element. Der AR wird in der Secure Enclave erzeugt, wenn der Benutzer zum ersten Mal eine Kreditkarte hinzufügt, und bleibt erhalten, solange Apple Pay aktiviert ist. Er wird von Mechanismen der Secure Enclave für Verschlüsselung und Anti-Rollback geschützt. Er wird mithilfe des Kopplungsschlüssels sicher an das Secure Element übertragen. Bei Erhalt eines neuen AR-Werts löscht das Secure Element alle zuvor hinzugefügten Karten.

Dem Secure Element hinzugefügte Kredit-, Debit- und Prepaid-Karten können nur verwendet werden, wenn das Secure Element eine Autorisierung mit dem Kopplungsschlüssel und dem AR-Wert, die beim Hinzufügen der Karte verwendet wurden, erhält. Dadurch kann iOS die Secure Enclave anweisen, Karten unbrauchbar zu machen, indem sie ihre Kopie des AR unter den folgenden Umständen als ungültig kennzeichnet:

- Der Code ist deaktiviert.
- Der Benutzer meldet sich bei iCloud ab.
- Der Benutzer löscht alle Inhalte und Einstellungen.
- Das Gerät wird aus dem Wartungsmodus wiederhergestellt.

Bei der Apple Watch werden Karten unter den folgenden Umständen als ungültig kennzeichnet:

- Der Code für die Uhr wurde deaktiviert.
- Die Kopplung von Uhr und iPhone wurde aufgehoben.
- Die Handgelenkerkennung wurde deaktiviert.

Mit dem Kopplungsschlüssel und der Kopie des aktuellen AR-Werts überprüft das Secure Element die Autorisierung der Secure Enclave, bevor das Zahlungs-Applet die kontaktlose Zahlung abwickeln kann. Dieser Prozess gilt auch, wenn verschlüsselte Zahlungsdaten von einem Zahlungs-Applet bei Transaktionen in einer App abgerufen werden.

Transaktionsspezifischer dynamischer Sicherheitscode

Alle Zahlungsvorgänge aus Zahlungs-Applets umfassen einen transaktions-spezifischen dynamischen Sicherheitscode zusammen mit einer Device Account Number. Dieser einmal nutzbare Code wird mit einem Zähler, der für jede neue Transaktion erhöht wird, und einem Schlüssel berechnet, der im Zahlungs-Applet während der Personalisierung bereitgestellt wird und dem Zahlungsnetzwerk und/oder dem Kartenaussteller bekannt ist. Abhängig vom Zahlungssystem werden auch andere Daten für die Berechnung dieser Codes verwendet, darunter:

- eine vom Zahlungs-Applet erzeugte Zufallszahl,
- eine vom Terminal erzeugte Zufallszahl (bei NFC-Transaktionen)
oder
- eine vom Server erzeugte Zufallszahl (bei In-App-Transaktionen).

Diese Sicherheitscodes werden an das Zahlungsnetzwerk und den Kartenaussteller weitergegeben, sodass beide die Transaktion überprüfen können. Die Länge dieser Sicherheitscodes hängen von der jeweiligen Transaktion ab.

Kontaktlose Zahlungen mit Apple Pay

Wenn das iPhone eingeschaltet ist und ein NFC-Feld erkennt, bietet es dem Benutzer entweder die relevante Kredit-, Debit- oder Prepaid-Karte oder alternativ die Standardkarte an, die in den Einstellungen festgelegt wurde. Der Benutzer kann auch die App „Wallet“ öffnen und eine Kredit- oder Debitkarte auswählen oder falls das Gerät gesperrt ist, zweimal auf die Home-Taste drücken.

Anschließend muss sich der Benutzer mit Touch ID, Face ID oder dem Code authentifizieren, bevor die Zahlungsinformationen übertragen werden. Wenn die Apple Watch nicht gesperrt ist, kann durch Doppelklicken auf die Seitentaste die Standardkarte zur Bezahlung aktiviert werden. Ohne Benutzerauthentifizierung werden keine Zahlungsinformationen gesendet. Nach der Authentifizierung werden die Device Account Number und der transaktionsspezifische dynamische Sicherheitscode für die Zahlungsabwicklung verwendet. Weder Apple noch das Gerät des Benutzers sendet die vollständige Kartenummer an den Händler. Apple kann anonymisierte Informationen zur Zahlung erhalten, z. B. ungefähre Zeit und Ort der Transaktion, mit denen Apple Pay und andere Produkte und Dienste von Apple verbessert werden können.

Mit Apple Pay in Apps zahlen

Apple Pay kann auch für Zahlungen in iOS-Apps und in Apple Watch-Apps (ab watchOS 3) verwendet werden. Wenn der Benutzer in einer App mit Apple Pay zahlt, erhält Apple verschlüsselte Informationen zur Transaktion und sendet diese an den Entwickler oder Händler, nachdem sie erneut mit einem entwicklerspezifischen Schlüssel verschlüsselt wurden. Apple Pay speichert anonymisierte Informationen zu Transaktionen, wie den ungefähren Betrag. Diese Informationen können nicht mit dem Benutzer verknüpft werden und enthalten keine Angaben dazu, wofür der Benutzer gezahlt hat.

Wenn in einer App eine Apple Pay Transaktion gestartet wird, erhalten die Apple Pay Server die verschlüsselte Transaktion von dem Gerät, bevor sie an den Händler gesendet wird. Die Apple Pay Server verschlüsseln sie dann erneut mit einem händlerspezifischen Schlüssel, bevor sie die Transaktion weiterleiten.

Wenn eine App eine Zahlung anfordert, ruft sie eine API auf, um festzustellen, ob das Gerät Apple Pay unterstützt und der Benutzer Kredit- oder Debitkarten besitzt, mit denen auf einem vom Händler akzeptierten Zahlungsnetzwerk Zahlungen abgewickelt werden können. Die App fragt an, welche Informationen für die Transaktion benötigt werden, z. B. Rechnungs- und Lieferadresse oder Kontaktdaten. Danach ergeht an iOS die Aufforderung, die Apple Pay-Seite anzuzeigen, um Informationen für die App und andere benötigte Informationen abzufragen, z. B. welche Karte verwendet werden soll.

Zu diesem Zeitpunkt erhält die App die Adressdaten (Ort, Staat/Land und PLZ), um die Versandkosten berechnen zu können. Die App erhält erst dann alle Informationen, wenn der Benutzer die Zahlung mit Touch ID, Face ID oder dem Code bestätigt. Ist die Zahlung autorisiert, werden die Informationen der Apple Pay-Seite an den Händler übertragen.

Wenn der Benutzer eine Zahlung bestätigt, erfolgt ein Aufruf an die Apple Pay-Server, um eine kryptografische Nonce anzufordern, die eine ähnliche Funktion erfüllt, wie der vom NFC-Terminal gesendete Wert im Laden. Die Nonce wird zusammen mit anderen Daten zur Transaktion an das Secure Element weitergegeben, um die Zahlungsdaten zu erzeugen, die mit einem Apple Schlüssel verschlüsselt werden. Wenn die verschlüsselten Zahlungsdaten vom Secure Element ausgegeben werden, werden sie an die Apple Pay Server weitergegeben, die die Daten entschlüsseln, die enthaltene Nonce mit der vom Secure Element gesendeten Nonce abgleichen und die Zahlungsdaten mit dem Händlerschlüssel zu der Händler-ID erneut verschlüsseln. Sie werden anschließend an das Gerät zurückgegeben, das sie über die API an

die App zurückgibt. Die App überträgt sie dann an das Händlersystem zur Bearbeitung. Der Händler kann nun die Zahlungsdaten mit seinem privaten Schlüssel zur Bearbeitung entschlüsseln. Dadurch und mit der Signatur der Server von Apple kann der Händler überprüfen, ob diese Transaktion für ihn bestimmt ist.

Die APIs benötigen eine Berechtigung, die die unterstützten Händler-IDs angibt. Eine App kann auch zusätzliche Daten an das Secure Element zum Signieren senden, zum Beispiel eine Bestellnummer oder Kundenidentität, sodass die Transaktionen keinem falschen Kunden zugeordnet werden können. Der App-Entwickler kann „applicationData“ bei der „PKPaymentRequest“ angeben. Ein Hash-Wert dieser Daten wird dann mit den anderen Zahlungsdaten verschlüsselt. Der Händler muss nun überprüfen, ob sein „applicationData“ Hash-Wert mit dem in den Zahlungsdaten übereinstimmt.

Mit Apple Pay im Web oder mit Handoff zahlen

Apple Pay kann für Zahlungen auf Websites verwendet werden. In iOS 10 oder neuer können Transaktionen mit Apple Pay im Web mit dem iPhone und iPad ausgeführt werden. In macOS Sierra oder neuer können Apple Pay-Transaktionen auf einem Mac begonnen und dann unter Verwendung desselben iCloud-Accounts auf einem iPhone oder einer Apple Watch, auf dem bzw. der Apple Pay aktiviert ist, abgeschlossen werden.

Apple Pay im Web setzt voraus, dass alle teilnehmenden Websites bei Apple registriert sind. Die Apple-Server validieren den Domain-Namen und geben ein TLS-Client-Zertifikat aus. Websites mit Apple Pay-Unterstützung müssen ihre Inhalte über HTTPS bereitstellen. Für jede Zahlungstransaktion müssen Websites mit dem von Apple ausgestellten TLS-Client-Zertifikat eine sichere und eindeutige Händlersitzung mit einem Apple-Server starten. Die Daten der Händlersitzung werden von Apple signiert. Nachdem die Signatur der Händlersitzung verifiziert wurde, kann eine Website abfragen, ob der Benutzer über ein Apple Pay-fähiges Gerät verfügt und ob eine Kredit-, Debit- oder Prepaid-Karte auf dem Gerät aktiviert ist. Eine Freigabe weiterer Details erfolgt nicht. Wenn der Benutzer seine Informationen nicht teilen möchte, kann er Apple Pay-Anfragen in den Datenschutzeinstellungen von Safari in iOS und macOS deaktivieren.

Nach der Validierung einer Händlersitzung sind alle Datenschutz- und Sicherheitsmaßnahmen identisch mit denen, die bei Zahlungsvorgängen des Benutzers innerhalb einer App gelten.

Beim Handoff vom Mac zum iPhone oder zur Apple Watch nutzt Apple Pay das End-to-End-verschlüsselte IDS-Protokoll, um zahlungsrelevante Informationen zwischen dem Mac des Benutzers und dem autorisierenden Gerät zu übertragen. IDS nutzt die Geräteschlüssel des Benutzers für die Verschlüsselung, sodass kein anderes Gerät diese Informationen entschlüsseln kann. Diese Schlüssel stehen Apple nicht zur Verfügung. Die Geräteerkennung für Apple Pay-Handoff umfasst den Typ und die eindeutige Kennung der Kreditkarten des Benutzers sowie einige Metadaten. Die gerätespezifische Accountnummer der Karte des Benutzers wird nicht preisgegeben und verbleibt sicher auf dem iPhone oder der Apple Watch des Benutzers. Apple sorgt mit dem iCloud-Schlüsselbund für die sichere Übertragung der kürzlich verwendeten Kontakt-, Liefer- und Rechnungsadressen des Benutzers.

Nachdem der Benutzer die Zahlung per Touch ID, Face ID oder Code auf dem iPhone oder durch Doppelklicken auf die Seitentaste der Apple Watch autorisiert hat, wird ein für das Händlerzertifikat jeder Website verschlüsseltes Zahlungs-Token auf sichere Weise vom iPhone bzw. von der Apple Watch des Benutzers an seinen Mac übertragen und dann der Website des Händlers bereitgestellt.

Nur Geräte, die sich in unmittelbarer Nähe zueinander befinden, können eine Zahlung anfordern und abschließen. Die Nähe wird durch Bluetooth Low Energy-Ankündigungen bestimmt.

Kundenkarten

In iOS 9 oder neuer unterstützt Apple Pay das Protokoll Value Added Service (VAS) für die Übertragung von Daten von händlerspezifischen Kundenkarten auf kompatible NFC-Terminals. Das VAS-Protokoll kann auf den Terminals des Händlers implementiert werden; für die Kommunikation mit den unterstützten Apple-Geräten wird NFC verwendet. Das VAS-Protokoll funktioniert nur über kurze Distanzen; es wird verwendet, um im Zuge einer Apple Pay-Transaktion komplementäre Dienste wie das Übertragen der Informationen von einer Kunden- oder Bonuskarte zu ermöglichen.

Das NFC-Terminal initiiert den Empfang der Karteninformationen, indem es eine Anfrage bezüglich der Karte sendet. Ist der Benutzer im Besitz einer Karte mit der Kennung des jeweiligen Anbieters, wird er aufgefordert, die Verwendung dieser Karte zu autorisieren. Wenn der Händler das Verschlüsseln der Daten unterstützt, wird ein Verschlüsselungscode für die Kartendaten generiert, mit dem dann die an das Terminal gesendeten Daten verschlüsselt werden. Basis für diesen Verschlüsselungscode sind die Karteninformationen, ein Zeitstempel und ein nur einmal nutzbarer ECDH P-256-Zufallscode. Wenn der Händler das Verschlüsseln der Daten nicht unterstützt, wird der Benutzer aufgefordert, das Gerät ein weiteres Mal vor das Terminal zu halten, bevor die Informationen von der Kundenkarte übertragen werden.

Apple Pay Cash

Ab iOS 11.2 und watchOS 4.2 kann Apple Pay auf einem iPhone, iPad oder auf einer Apple Watch verwendet werden, um Geld an andere Benutzer zu senden bzw. von diesen zu empfangen oder anzufordern. Wenn ein Benutzer Geld empfängt, wird es zu einem Apple Pay Cash-Konto hinzugefügt, auf das in Wallet oder unter „Einstellungen“ > „Wallet & Apple Pay“ auf allen berechtigten Geräten zugegriffen werden kann, auf denen sich der Benutzer mit seiner Apple-ID angemeldet hat.

Um Apple Pay Cash und Zahlungen zwischen Personen zu verwenden, muss ein Benutzer mit einem iCloud-Account auf einem Apple Pay Cash-fähigen Gerät angemeldet sein und die Zwei-Faktor-Authentifizierung im iCloud-Account eingerichtet haben.

Wenn du Apple Pay Cash einrichtest, können dieselben Informationen, die du beim Hinzufügen einer Kredit- oder Debitkarte eingibst, mit unserer Partnerbank Green Dot Bank und mit Apple Payments Inc. geteilt werden. Dies ist ein eigenes Tochterunternehmen, das zum Schutz deiner Privatsphäre gegründet wurde, indem Informationen getrennt von den restlichen Apple-Funktionen gespeichert und verarbeitet werden und zwar so, dass sie in den restlichen Apple-Funktionen nicht bekannt sind. Die Informationen werden nur bei der Fehlerbehebung, zur Verhinderung von Betrug und zu Regulierungszwecken verwendet.

Geldanforderungen und -überweisungen zwischen Benutzern werden in der App „Nachrichten“ oder über Siri gestartet. Wenn ein Benutzer versucht, Geld zu senden, wird in iMessage die Apple Pay-Seite angezeigt. Das Apple Pay Cash-Guthaben wird immer zuerst verwendet. Falls erforderlich, wird ein zusätzlicher Betrag von einer zweiten Kredit- oder Debitkarte abgebucht, die der Benutzer in Wallet hinzugefügt hat.

Die Apple Pay Cash-Karte in Wallet kann mit Apple Pay verwendet werden, um in Stores, in Apps und im Web zu bezahlen. Das Geld auf dem Apple Pay Cash-Konto kann auch auf ein Bankkonto überwiesen werden. Zusätzlich zum

Geld, das du von anderen Benutzern erhältst, kann dem Apple Pay Cash-Konto auch Geld über eine Debit- oder Prepaid-Karte in Wallet hinzugefügt werden.

Nach dem Abschluss einer Transaktion speichert Apple Payments Inc. deine Transaktionsdaten möglicherweise für die Fehlerbehebung, zur Verhinderung von Betrug und zu Regulierungszwecken. Die restlichen Apple-Funktionen wissen nicht, an wen du Geld gesendet hast, von wem du Geld erhalten hast oder wo du mit deiner Apple Pay Cash-Karte eingekauft hast.

Wenn ein Benutzer Geld mit Apple Pay sendet, einem Apple Pay Cash-Konto Geld hinzufügt oder Geld auf ein Bankkonto überweist, wird ein Aufruf an die Apple Pay-Server gesendet, um eine kryptografische Nonce anzufordern, die dem Wert ähnelt, der in Apps für Apple Pay zurückgegeben wird. Die Nonce wird zusammen mit anderen Daten zur Transaktion an das Secure Element weitergegeben, um eine Zahlungssignatur zu erzeugen. Die Zahlungssignatur wird vom Secure Element an die Apple Pay-Server weitergeleitet. Die Apple Pay-Server überprüfen die Authentifizierung, Integrität und Korrektheit der Transaktion über die Zahlungssignatur und die Nonce. Anschließend wird die Geldüberweisung gestartet und der Benutzer über den Abschluss der Transaktion benachrichtigt.

Die Transaktion kann eine Kredit- oder Debitkarte für folgende Zwecke enthalten:

- Geld zu Apple Pay Cash hinzufügen oder
- Geld an einen anderen Benutzer senden oder
- zusätzliches Geld bereitstellen, wenn das Apple Pay Cash-Guthaben nicht ausreicht

In diesem Fall werden zusätzlich zu der oben beschriebenen Überweisungssignatur auch verschlüsselte Zahlungsdaten generiert und an die Apple Pay-Server gesendet. Diese sind den Daten ähnlich, die in Apps und auf Websites für Apple Pay verwendet werden.

Sobald das Guthaben auf dem Apple Pay Cash-Konto einen bestimmten Betrag überschreitet oder wenn ungewöhnliche Aktivitäten festgestellt werden, wird der Benutzer aufgefordert, seine Identität zu verifizieren. Bereitgestellte Informationen zum Verifizieren der Benutzeridentität, wie die Sozialversicherungsnummer oder Antworten auf Fragen (z. B. die Bestätigung des Namens einer Straße, in der du früher gewohnt hast), werden sicher an den Apple-Partner übertragen und mit seinem Schlüssel verschlüsselt. Apple kann diese Daten nicht entschlüsseln.

Suica-Karten

In Japan können Benutzer auf unterstützten iPhone- oder Apple Watch-Modellen eine Suica-Karte zu Apple Pay in Wallet hinzufügen. Dazu werden entweder der Wert und der ÖPNV-Ausweis einer physischen Karte in die digitale Wallet-Darstellung übertragen oder über die Suica-App eine neue Suica-Karte in Wallet bereitgestellt. Nachdem Suica-Karten in Wallet hinzugefügt wurden, kann der Benutzer mit einer anonymen Suica-Karte, einer MySuica-Karte oder einer Karte, die einen ÖPNV-Ausweis enthält, in Stores bezahlen oder den Nahverkehr nutzen.

Hinzugefügte Suica-Karten werden mit dem iCloud-Account eines Benutzers verknüpft. Wenn der Benutzer mehr als eine Karte in Wallet hinzufügt, können Apple oder der Nahverkehrsbetrieb die persönlichen Daten des Benutzers und die verknüpften Accountdaten möglicherweise zwischen Karten verbinden. MySuica-Karten können beispielsweise mit anonymen

Suica-Karten verbunden werden. Suica-Karten und Transaktionen werden durch eine Reihe von hierarchischen kryptografischen Schlüsseln geschützt.

Bei einer anonymen Suica-Karte müssen Benutzer während der Übertragung des Guthabens von der physischen Karte in Wallet die letzten vier Stellen der Kartenseriennummer eingeben. Bei einer MySuica-Karte oder bei einer Karte, die einen ÖPNV-Ausweis enthält, müssen die Benutzer auch ihr Geburtsdatum eingeben, um zu beweisen, dass sie im Besitz der Karte sind. Wenn du einen Ausweis von deinem iPhone auf die Apple Watch überträgst, müssen beide Geräte online sein.

Das Guthaben kann über Beträge von einer Kredit- oder Prepaid-Karte in Wallet oder über die Suica-App aufgeladen werden. Die Sicherheit beim Aufladen des Guthabens mit Apple Pay wird im Abschnitt „Mit Apple Pay in Apps zahlen“ in diesem Dokument beschrieben.

Der Prozess zum Bereitstellen der Suica-Karte in der Suica-App wird im Abschnitt „Kredit- oder Debitkarten aus der App des Kartenausstellers zu Apple Pay hinzufügen“ in diesem Dokument beschrieben.

Der Nahverkehrsbetrieb besitzt die kryptografischen Schlüssel, die erforderlich sind, um die physische Karte zu authentifizieren und die eingegebenen Benutzerdaten zu überprüfen. Nach der Überprüfung kann das System eine Gerätekontonummer für Secure Element erstellen und den neu hinzugefügten Ausweis mit dem übertragenen Guthaben in Wallet aktivieren. Nachdem die Bereitstellung abgeschlossen ist, wird die physische Karte deaktiviert.

Zum Abschluss beider Arten der Bereitstellung wird das Suica-Guthaben verschlüsselt und in einem spezifischen Applet im Secure Element gespeichert. Der Nahverkehrsbetrieb besitzt die Schlüssel, um kryptografische Operationen für Guthabentransaktionen mit den Kartendaten auszuführen.

Den Benutzern stehen standardmäßig die Vorteile von „Express-ÖPNV“ zur Verfügung, damit sie ohne Touch ID, Face ID oder einen Code zahlen oder den Nahverkehr nutzen können. Wenn der Expressmodus aktiviert ist, kann an kontaktlosen Kartenlesegeräten auf Informationen wie kürzlich besuchte Stationen, letzte Transaktionen und zusätzliche Fahrscheine zugegriffen werden. Benutzer können die Anforderung einer Autorisierung per Touch ID, Face ID oder Code in der Einstellung „Wallet & Apple Pay“ aktivieren, indem sie „Express-ÖPNV“ deaktivieren.

Wie bei anderen Apple Pay-Karten können Benutzer Suica-Karten wie folgt sperren oder entfernen:

- Das Gerät mit „Mein iPhone suchen“ fernlöschen
- Modus „Verloren“ mit „Mein iPhone suchen“ aktivieren
- Per MDM fernlöschen
- Alle Karten auf der Accountseite der Apple-ID entfernen
- Alle Karten auf iCloud.com entfernen
- Alle Karten in Wallet entfernen

Die Apple Pay-Server benachrichtigen den Nahverkehrsbetrieb, um diese Suica-Karten zu deaktivieren. Wenn ein Gerät beim Löschversuch offline ist, können Suica-Karten an einigen Terminals möglicherweise noch bis 0:01 Uhr JST am Folgetag genutzt werden.

Wenn Benutzer ihre Suica-Karten entfernen, kann das Guthaben erstattet werden. Die Karten können ab 5:00 Uhr JST am Folgetag auf einem Gerät hinzugefügt werden, das mit derselben Apple-ID angemeldet ist.

Suica-Karten können nicht gesperrt werden, wenn dein Gerät offline ist.

Karten sperren, entfernen und löschen

Benutzer können Apple Pay auf iPhone, iPad und Apple Watch mit watchOS 3 sperren, indem sie ihr Gerät über „Mein iPhone suchen“ in den Modus „Verloren“ versetzen. Karten können über „Mein iPhone suchen“, über iCloud.com oder direkt auf den Geräten in der App „Wallet“ aus Apple Pay entfernt und gelöscht werden. Bei einer Apple Watch lassen sich Karten mit den iCloud-Einstellungen, der App „Apple Watch“ auf dem iPhone oder direkt auf der Watch entfernen. Die Möglichkeit, auf dem Gerät per Karte zu bezahlen, kann für Apple Pay vom Kartenaussteller oder dem Zahlungsnetzwerk gesperrt oder entfernt werden, auch wenn das Gerät offline und nicht mit einem Mobilfunk- oder WLAN-Netzwerk verbunden ist. Benutzer können sich auch an ihren Kartenaussteller wenden, um Karten für Apple Pay sperren oder entfernen zu lassen.

Wenn ein Benutzer das Gerät in „Mein iPhone suchen“ mit „Inhalte & Einstellungen löschen“ komplett löscht oder das Gerät im Wartungsmodus wiederherstellt, weist iOS das Secure Element an, alle Karten als gelöscht zu markieren. Dadurch werden die Karten sofort in Apple Pay unbrauchbar, bis die Apple Pay-Server angewiesen werden können, die Karten völlig aus dem Secure Element zu löschen. Unabhängig davon markiert die Secure Enclave den AR als ungültig, sodass keine weiteren Zahlungen für die registrierten Karten mehr autorisiert werden können. Wenn das Gerät online ist, kontaktiert es die Apple Pay-Server, um sicherzustellen, dass alle Karten im Secure Element gelöscht werden.

Internetdienste

Erstellen eines sicheren Passworts für die Apple-ID

Apple-IDs werden für die Anmeldung bei verschiedenen Diensten wie iCloud, FaceTime und iMessage verwendet. Alle Passwörter für neue Accounts müssen die folgenden Attribute aufweisen, 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-Vorschläge, iCloud, iCloud-Backup und iCloud-Schlüsselbund.

Diese Internetdienste wurden mit denselben Sicherheitszielen entwickelt, die auf der gesamten iOS-Plattform 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 ist der Account, der verwendet wird, um sich bei verschiedenen Apple-Diensten wie iCloud, iMessage, FaceTime, iTunes Store, iBooks Store, App Store und anderen anzumelden. Es ist wichtig, dass Benutzer ihre Apple-ID sicher aufbewahren, um einen 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 aufweisen dürfen und nicht mit einem häufig genutzten Passwort übereinstimmen dürfen. Es wird Benutzern nachdrücklich empfohlen, über diese Richtlinien hinaus weitere Zeichen und Interpunktionszeichen hinzuzufügen, um ein noch sichereres Passwort zu erhalten. Apple verlangt außerdem, Antworten auf drei Sicherheitsfragen bereitzustellen, anhand derer die Identität eines Benutzers überprüft werden kann, wenn dieser Änderungen an seinen Accountdaten vornehmen oder ein vergessenes Passwort zurücksetzen will.

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 verdächtig erscheint, sollten sie sofort das Passwort für ihre Apple-ID ändern.

Apple nutzt darüber hinaus eine Vielzahl von Richtlinien und Verfahren, die darauf abzielen, Benutzeraccounts zu schützen. Dazu gehören die Beschränkung der Eingabeversuche beim Anmelden oder Zurücksetzen des Passworts, eine aktive Betrugsüberwachung, um Angriffe sofort beim Auftreten zu entdecken, sowie regelmäßige Überprüfungen der Richtlinien, die es Apple ermöglichen, diese an neue Informationen anzupassen, die Auswirkungen auf die Sicherheit der Kunden haben könnten.

Zwei-Faktor-Authentifizierung

Um Benutzern dabei zu helfen, ihre Accounts noch besser zu sichern, bietet Apple eine Zwei-Faktor-Authentifizierung an – eine weitere Sicherheitsebene für Apple-IDs. Damit soll sichergestellt werden, dass nur der Accountinhaber auf den Account zugreifen kann, auch wenn eine andere Person das Passwort kennt.

Mit der Zwei-Faktor-Authentifizierung ist der Zugriff auf einen Benutzeraccount nur auf vertrauenswürdigen Geräten möglich, zum Beispiel auf dem iPhone, iPad oder Mac des Benutzers. Für die erste Anmeldung auf einem neuen Gerät sind zwei Informationen erforderlich: zum einen das Passwort der Apple-ID und zum anderen ein sechsstelliger Bestätigungscode, der automatisch auf

den vertrauenswürdigen Geräten des Benutzers angezeigt oder an eine vertrauenswürdige Telefonnummer gesendet wird. Durch die Eingabe des Codes bestätigt der Benutzer, dass er diesem neuen Gerät vertraut und dass die Anmeldung sicher ist. Da ein Passwort allein nicht mehr ausreicht, um auf einen Benutzeraccount zuzugreifen, trägt die Zwei-Faktor-Authentifizierung dazu bei, die Sicherheit der Apple-ID des Benutzers und aller seiner bei Apple gespeicherten persönlichen Daten zu verbessern. Sie ist direkt in iOS, macOS, tvOS, watchOS und in die auf den Apple-Websites verwendeten Authentifizierungssysteme integriert.

Näheres zur Zwei-Faktor-Authentifizierung findest du unter:
<https://support.apple.com/de-de/HT204915>.

Zweistufige Überprüfung

Seit 2013 bietet Apple mit der *zweistufigen Überprüfung* bereits eine ähnliche Sicherheitsmethode an. Bei aktivierter zweistufiger Überprüfung muss die Identität des Benutzers mithilfe eines temporären Codes bestätigt werden, der an eines der vertrauenswürdigen Geräte gesendet wird, bevor von einem neuen Gerät aus die Apple-ID-Accountdaten geändert werden können, der Benutzer sich bei iCloud, iMessage, FaceTime oder Game Center anmelden oder Einkäufe im iTunes Store, iBooks Store oder App Store tätigen kann. Die Benutzer erhalten zudem einen 14-stelligen Wiederherstellungsschlüssel zur sicheren Aufbewahrung für den Fall, dass sie ihr Passwort vergessen oder nicht mehr auf die vertrauenswürdigen Geräte zugreifen können. Obwohl die meisten neuen Benutzer zur Verwendung der Zwei-Faktor-Authentifizierung aufgefordert werden, gibt es immer noch einige Situationen, in denen stattdessen die zweistufige Überprüfung empfohlen wird.

Näheres zur zweistufigen Überprüfung für die Apple-ID findest du unter:
<https://support.apple.com/de-de/HT204152>

Verwaltete Apple-IDs

Verwaltete Apple-IDs funktionieren ähnlich wie eine Apple-ID, befinden sich aber im Eigentum einer Bildungseinrichtung oder unterliegen deren Kontrolle. Die Einrichtung kann Passwörter zurücksetzen, die Kaufmöglichkeiten und Kommunikationsfunktionen wie FaceTime und Nachrichten einschränken und rollenbasierte Zugriffsrechte für Mitarbeiter, Lehrkräfte und Schüler festlegen.

Einige Apple-Dienste wie Apple Pay, iCloud-Schlüsselbund, HomeKit und die Funktion „Mein iPhone suchen“ sind für verwaltete Apple-IDs deaktiviert.

Näheres über verwaltete Apple-IDs findest du unter:
<https://help.apple.com/schoolmanager/>

Überprüfung verwalteter Apple-IDs

Verwaltete Apple-IDs unterstützen zudem Überprüfungen, sodass Bildungseinrichtungen gesetzliche Vorschriften und Datenschutzregelungen einhalten können. Den Accounts von Administratoren, Managern oder Lehrkräften können Überprüfungsrechte für bestimmte verwaltete Apple-IDs zugewiesen werden. Die Prüfer können nur Accounts überwachen, die in der Hierarchie der Einrichtung unter ihnen stehen. Das heißt, Lehrkräfte können Schüler/Studenten überwachen, Manager können Lehrkräfte überprüfen und Administratoren können wiederum Manager, Lehrkräfte und Schüler/Studenten überprüfen.

Wenn mithilfe von Apple School Manager Überprüfungsanmeldedaten angefordert werden, wird ein spezieller Account angelegt, der nur auf die verwaltete Apple-ID zugreifen kann, für die eine Überprüfung angefragt wurde. Der Überprüfungszugriff läuft nach sieben Tagen ab. Während dieses Zeitraums kann der Prüfer die in iCloud- oder in CloudKit-fähigen Apps gespeicherten Inhalte des Benutzers lesen und Änderungen daran vornehmen. Alle Anfragen für einen solchen Überprüfungszugriff werden in Apple School Manager protokolliert. Die Protokolle zeigen den Namen des Prüfers, die Apple-ID, auf die der Prüfer Zugriff angefordert hat, die Uhrzeit der Anfrage und die Angabe, ob die Überprüfung stattgefunden hat oder nicht.

Verwaltete Apple-IDs und persönliche Geräte

Verwaltete Apple-IDs können auch mit persönlichen iOS-Geräten und Mac-Computern verwendet werden. Die Schüler/Studenten melden sich dazu bei iCloud mit der von der Einrichtung ausgegebenen verwalteten Apple-ID und einem zusätzlichen privaten Passwort an, das als zweite Stufe für die Zwei-Faktor-Authentifizierung der Apple-ID dient. Während der Verwendung einer verwalteten Apple-ID auf einem persönlichen Gerät ist der iCloud-Schlüsselbund nicht verfügbar und die Nutzung bestimmter Funktionen wie FaceTime oder Nachrichten kann eingeschränkt sein. Alle iCloud-Dokumente, die Schüler/Studenten erstellen, wenn sie angemeldet sind, können wie zuvor in diesem Abschnitt beschrieben überprüft werden.

iMessage

iMessage ist ein Messaging-Dienst für iOS-Geräte, Apple Watch 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) in vollem Umfang. 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 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. Bei einigen Netzwerken muss eine SMS verwendet werden. (Dem Benutzer wird ein Bestätigungsdialog angezeigt, wenn die SMS nicht kostenfrei ist.) Eine Verifizierung der Telefonnummer kann neben iMessage für einige Systemdienste wie FaceTime und iCloud erforderlich sein. Außerdem zeigen alle registrierten Geräte eines Benutzers eine Warnung an, wenn ein neues Gerät, eine neue Telefonnummer oder eine neue E-Mail-Adresse hinzugefügt wird.

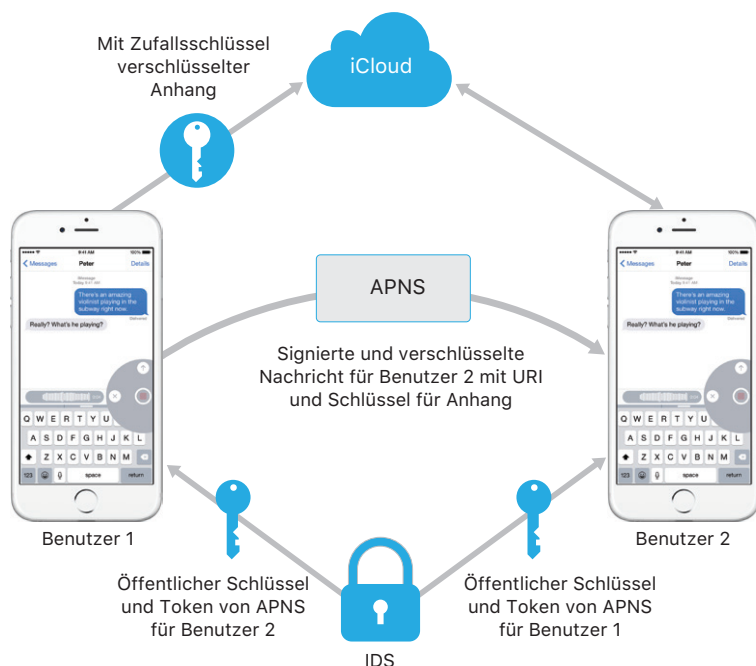
So sendet und empfängt iMessage Nachrichten

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 88-Bit-Wert und verwendet diesen als HMAC-SHA256-Schlüssel, um einen 40-Bit-Wert zu erstellen, der vom öffentlichen Schlüssel des Senders und Empfängers und vom Klartext abgeleitet wird. Durch Verkettens des 88-Bit-Werts mit dem 40-Bit-Wert ergibt sich ein

128-Bit-Wert, der die Nachricht mittels AES im CTR-Modus verschlüsselt. Der 40-Bit-Wert wird auf Seiten des Empfängers verwendet, um die Integrität des entschlüsselten Klartexts zu überprüfen. 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 Sendegeräts 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 Routing-Informationen des APNS 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, sein 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 Diagramm gezeigt 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 aus 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 Empfängergerät im Offline-Modus befindet. Nachrichten werden derzeit für maximal 30 Tage gespeichert.

FaceTime

FaceTime ist der Dienst für Video- und Audioanrufe von Apple. Für FaceTime-Anrufe wird, genau wie für iMessage, der APNS (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.

Die erste FaceTime-Verbindung wird über die Apple-Serverinfrastruktur hergestellt, die Datenpakete zwischen den registrierten Geräten der Benutzer weiterleitet. Die Geräte verifizieren ihre Identitätszertifikate und erstellen ein Shared Secret für jede Sitzung mittels APNS-Benachrichtigungen und STUN-Nachrichten (Session Traversal Utilities for NAT) über die weitergeleitete Verbindung. Das Shared Secret wird verwendet, um Sitzungsschlüssel für Medienkanäle abzuleiten, die über SRTP (Secure Real-time Transport Protocol) gestreamt werden. SRTP-Pakete werden mit AES-256 im Counter Mode und mit HMAC-SHA1 verschlüsselt. Falls möglich, verwendet FaceTime nach der ersten Verbindung und der Sicherheitskonfiguration STUN und ICE (Internet Connectivity Establishment) zum Erstellen einer Peer-To-Peer-Verbindung zwischen den Geräten.

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 MDM-Konfigurationsprofile deaktiviert werden. Der Dienst behandelt alle Dateiinhalte identisch, als eine Ansammlung von Byte.

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. S3 und Google Cloud Platform gespeichert.

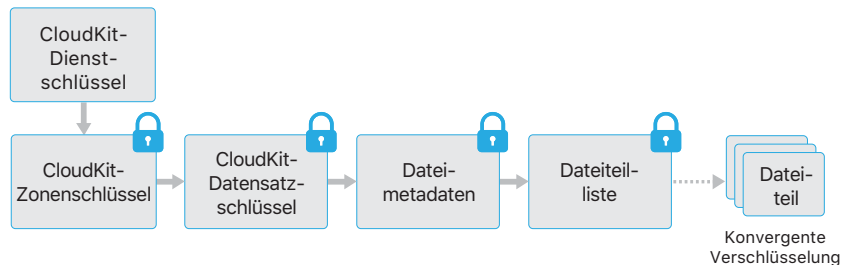
iCloud Drive

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 anzuzeigen.

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.



End-to-End-Verschlüsselung in CloudKit

Apple Pay Cash, Benutzerschlüsselwörter, Siri Intelligence und Hey Siri verwenden die End-to-End-Verschlüsselung in CloudKit mit einem CloudKit-Dienstschlüssel, der durch die Synchronisierung mit dem iCloud-Schlüsselbund geschützt ist. Die Schlüsselhierarchie für diese CloudKit-Container ist im iCloud-Schlüsselbund verankert und hat somit dieselben Sicherheitsmerkmale wie der iCloud-Schlüsselbund. Die Schlüssel sind nur auf den als vertrauenswürdig eingestuften Geräten des Benutzers und nicht für Apple oder einen anderen Anbieter verfügbar. Wenn der Zugriff auf die Daten des iCloud-Schlüsselbunds verloren geht (siehe Abschnitt „Escrow-Sicherheit“ weiter unten im Dokument), werden die Daten in CloudKit zurückgesetzt. Wenn Daten auf dem als vertrauenswürdig eingestuften lokalen Gerät verfügbar sind, werden sie erneut in CloudKit geladen.

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:

- Einträge für gekaufte Musik, Filme, TV-Sendungen, Apps und Bücher. Das iCloud-Backup eines Benutzers umfasst Informationen über gekaufte Inhalte, die auf dem iOS-Gerät des Benutzers vorhanden sind, nicht jedoch die gekauften Inhalte selbst. Wenn der Benutzer ein iCloud-Backup wiederherstellt, werden seine gekauften Inhalte automatisch aus dem iTunes Store, iBooks Store oder App Store geladen. Einige Arten von Inhalten werden nicht in allen Ländern und Regionen automatisch heruntergeladen und bereits getätigte Einkäufe sind möglicherweise nicht verfügbar, wenn sie zurückerstattet wurden oder im jeweiligen Store nicht mehr angeboten werden. Der vollständige Einkaufsverlauf ist der Apple-ID des Benutzers zugewiesen.
- Fotos und Videos auf iOS-Geräten des Benutzers. Hinweis: Aktiviert ein Benutzer die iCloud-Fotomediathek auf seinem iOS-Gerät (iOS 8.1 oder neuer) oder Mac (OS X 10.10.3 oder neuer), sind seine Fotos und Videos bereits in iCloud gespeichert, d. h., sie werden nicht in das iCloud-Backup des Benutzers einbezogen.
- Kontakte, Kalender-Ereignisse, Erinnerungen und Notizen
- Geräteeinstellungen
- App-Daten
- Anrufliste und Klingeltöne
- Home-Bildschirm und Anordnung der Apps
- HomeKit-Konfiguration
- HealthKit-Daten
- iMessage, Text- (SMS) und MMS-Nachrichten (erfordert die SIM-Karte, die während des Backups verwendet wurde)
- Visual Voicemail-Passwort (erfordert die SIM-Karte, die während des Backups verwendet wurde)

Wenn Dateien in Datensicherheitsklassen erstellt werden, auf die bei gesperrtem Gerät nicht zugegriffen werden kann, werden ihre pro Datei erzeugten Schlüssel mit den Klassenschlüsseln des iCloud-Backup-Keybags verschlüsselt. Die Dateien 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 findest du unter „Schutz von Schlüsselbunddaten“ im Abschnitt „Verschlüsselung und Datensicherheit“ in diesem Dokument.

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, ist aber nach wie vor durch einen 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 zufällige, kryptografisch sichere Zeichenfolgen als Passwörter für Websites erzeugen, die im Schlüsselbund gesichert und mit 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

Mit dem iCloud-Schlüsselbund können Benutzer ihre Passwörter sicher zwischen iOS-Geräten und Mac-Computern synchronisieren, ohne die Informationen für Apple offenzulegen. Zusätzlich zu Datenschutz und Sicherheit waren Benutzerfreundlichkeit und Wiederherstellbarkeit eines Schlüsselbunds weitere Ziele, die sich nachhaltig auf Konzeption und Architektur des iCloud-Schlüsselbunds ausgewirkt haben. Der iCloud-Schlüsselbund umfasst zwei Dienste: Schlüsselbund synchronisierung 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 wird kompromittiert.
- iCloud wird von einem Angreifer von außen oder einem Mitarbeiter kompromittiert.
- Ein anderer Anbieter greift auf Benutzeraccounts zu.

Schlüsselbund synchronisierung

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 (P-256) 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 das neue Gerät 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 Mitgliedern, von denen jedes den öffentlichen Schlüssel des anderen besitzt. Sie tauschen nun untereinander einzelne Schlüsselbundobjekte über den iCloud-Schlüssel/Wert-Speicher aus oder speichern sie bei Bedarf in CloudKit. Ist ein Objekt bei beiden vorhanden, wird das zuletzt geänderte Objekt synchronisiert. Wenn das andere Mitglied das gleiche

Objekt besitzt und dieses zuletzt zum gleichen Zeitpunkt geändert wurde, wird das Objekt übersprungen. Jedes synchronisierte Objekt wird verschlüsselt. Es kann also nur von einem Gerät im Circle of Trust des Benutzers entschlüsselt werden. Weder andere Geräte noch Apple können es entschlüsseln.

Dieser Prozess wird für jedes neue Gerät im Circle wiederholt. Wenn z. B. ein drittes Gerät beitrifft, 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. Auch 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 mit einem sicheren Code verschlüsselt, und der Treuhanddienst stellt nur dann eine Kopie des Schlüsselbunds bereit, wenn eine Reihe strikter Bedingungen erfüllt ist.

Wenn der iCloud-Schlüsselbund und die Zwei-Faktor-Authentifizierung für den Benutzeraccount aktiviert sind, wird der Code des Geräts verwendet, um einen hinterlegten (escrow) Schlüsselbund wiederherzustellen. Wenn die Zwei-Faktor-Authentifizierung nicht eingerichtet ist, wird der Benutzer gebeten, einen iCloud-Sicherheitscode zu erstellen, indem er einen sechsstelligen Code eingibt. Benutzer ohne Zwei-Faktor-Authentifizierung 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 Schlüsselbunds des Benutzers wobei die Schlüssel in einem asymmetrischen Keybag verpackt werden und in den iCloud-Schlüssel/Wert-Speicher des Benutzers gelegt werden. Der Keybag wird mit dem iCloud-Sicherheitscode des Benutzers und dem öffentlichen Schlüssel des HSM-Clusters (Hardware-Sicherheitsmodul) verpackt, der den Escrow-Eintrag speichert. 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 (Escrow) sichergestellt, dass nur autorisierte Benutzer und Geräte eine Wiederherstellung durchführen können. Hinter iCloud stehen HSM-Cluster, mit denen die Escrow-Einträge geschützt werden. Jedes besitzt einen Schlüssel, mit dem die geschützten Escrow-Einträge verschlüsselt werden, wie bereits in diesem Dokument 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 SMS authentifizieren. Anschließend muss der Benutzer seinen iCloud-Sicherheitscode eingeben. Der HSM-Cluster überprüft mit dem 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 (siehe nachfolgende Beschreibung). 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 wenden. Nach dem 10. fehlgeschlagenen Versuch 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 erhält der Eigentümer jedes mit dem Cluster geschützten Schlüsselbunds eine Mitteilung, dass der Escrow-Eintrag gelöscht wurde. Er kann 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 Benutzerinformationen an den Server. Dazu gehören Informationen zur 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 genauere Standort benötigt, z. B. um Kinos in der Nähe zu finden, bittet der Server das Gerät um genauere Ortsdaten. Dies 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 zuvor beschrieben. Statt die Benutzerinformationen erneut zu senden, senden die Anfragen auch die Siri-Kennung für 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 diktieren 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 von Siri nutzen kann. Ein kleiner Teil von Aufzeichnungen, Abschriften und zugehörigen Daten ohne Kennungen kann von Apple über diese 2 Jahre hinaus zur kontinuierlichen Verbesserung und Qualitätssicherung von Siri weiter verwendet werden. 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 Spracherkennungsserver von Apple zur weiteren Bearbeitung gesendet.

Integration

Integration 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 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. Mit diesem Schlüssel können die Bluetooth Low Energy-Ankündigungen, die den anderen in iCloud gekoppelten Geräten die aktuelle Aktivität des Geräts mitteilen, verschlüsselt und authentifiziert werden, 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 APNS (Apple-Dienst für Push-Benachrichtigungen) anstatt über Bluetooth Low Energy versendet. Die 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 findest du unter „Zugriff auf gesicherte Passwörter in Safari“ im Abschnitt „Verschlüsselung und Datensicherheit“ in diesem Dokument. Das System muss die Kontrolle der App über den Domain-Namen validieren, bevor sie Handoff für die Benutzeraktivität verwenden kann.

Ursprung für das 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 zuvor in diesem Abschnitt 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 beschränkt, deren Eigentümer der Entwickler der nativen App ist.

Handoff größerer 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.

Universelle Zwischenablage

Die universelle Zwischenablage nutzt Handoff für die sichere Übertragung der Inhalte der Zwischenablage zwischen Geräten. Dadurch ist es möglich, etwas auf einem Gerät zu kopieren und auf einem anderen Gerät einzusetzen. Der Inhalt wird dabei auf gleiche Weise wie andere Handoff-Daten geschützt und standardmäßig mit der universellen Zwischenablage geteilt, es sei denn, der App-Entwickler lässt die Freigabe nicht zu.

Die Daten in der Zwischenablage sind für Apps zugänglich, unabhängig davon, ob der Benutzer die Zwischenablage in die App eingesetzt hat. Mit der universellen Zwischenablage wird dieser Datenzugriff auf Apps ausgeweitet, die auf anderen Geräten des Benutzers ausgeführt werden (wie bei der iCloud-Anmeldung festgelegt).

Automatisches Entsperrn

Mac-Computer mit Unterstützung für das automatische Entsperrn verwenden Bluetooth Low Energy und Peer-To-Peer-WLAN-Technologie, um der Apple Watch des Benutzers das sichere Entsperrn des Mac zu erlauben. Jeder fähige Mac und jede fähige Apple Watch, der bzw. die einem iCloud-Account zugeordnet ist, muss die Zwei-Faktor-Authentifizierung (TFA) verwenden.

Wird eine Apple Watch zum Entsperrn eines Mac aktiviert, wird mit Auto Unlock Identities ein sicherer Link erstellt. Der Mac erstellt ein zufälliges und nur einmal nutzbares Unlock Secret und überträgt dieses über den Link an die Apple Watch. Das Secret wird auf der Apple Watch gesichert und ist nur dann zugänglich, wenn die Apple Watch entsperrt ist (siehe Abschnitt „Datenschutzklassen“). Weder die Master-Entropie noch das neue Secret ist mit dem Passwort des Benutzers identisch.

Während eines Entsperrens verwendet der Mac Bluetooth Low Energy, um eine Verbindung zur Apple Watch herzustellen. Zwischen den beiden Geräten wird ein sicherer Link hergestellt, wobei die gemeinsamen Schlüssel zum Einsatz kommen, die bei der ersten Aktivierung verwendet wurden. Mac und Apple Watch nutzen dann Peer-To-Peer-WLAN und einen sicheren Schlüssel, der vom sicheren Link stammt, um die Entfernung zwischen beiden Geräten zu bestimmen. Befinden sich die Geräte in Reichweite, wird der sichere Link verwendet, um das Pre-Shared Secret zum Entsperrn des Mac zu übertragen. War der Entsperrvorgang erfolgreich, ersetzt der Mac das aktuelle Unlock Secret durch ein neues nur einmal nutzbares Unlock Secret und überträgt das neue Unlock Secret über den Link an die Apple Watch.

iPhone-Mobilanrufumleitung

Wenn ein Mac, iPad oder iPod touch des Benutzers dasselbe WLAN wie sein iPhone verwendet, können über die Mobilfunkverbindung des iPhone Telefonanrufe gestartet oder empfangen werden. Für die Konfiguration müssen die Geräte mit demselben Apple-ID-Account bei iCloud und FaceTime angemeldet sein.

Bei einem eingehenden Anruf werden alle konfigurierten Geräte über den Apple-Dienst für Push-Benachrichtigungen 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.

Wird ein Anruf auf einem Gerät entgegengenommen, wird das Klingeln von über iCloud gekoppelten Geräten in der Nähe durch eine kurze Ankündigung via Bluetooth Low Energy 4.0 beendet. Die Ankündigungsbytes werden mit derselben Methode wie Handoff-Ankündigungen verschlüsselt.

Ausgehende Anrufe werden ebenfalls über den Apple-Dienst für Push-Benachrichtigungen umgeleitet und die Audiodaten werden in ähnlicher Weise 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 iPad-, iPod touch- und Mac-Geräte des Benutzers übertragen. Alle Geräte müssen beim Dienst „iMessage“ mit derselben Apple-ID angemeldet sein. Wenn die Nachrichtenweiterleitung und die Zwei-Faktor-Authentifizierung aktiviert sind, erfolgt die Registrierung auf Geräten im Circle of Trust eines Benutzers automatisch. Andernfalls 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 (oder neuer) 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-Nachricht verschlüsselt. Das Gerät sendet dann über dieselbe Bluetooth Low Energy-Verbindung mit derselben nachrichtenspezifischen Verschlüsselung die Verbindungsinformationen für den persönlichen Hotspot.

Safari-Vorschläge, Siri-Vorschläge in „Suchen“, „Nachschlagen“, #images, App „News“ und News-Widget in Ländern ohne die App „News“

Safari-Vorschläge, Siri-Vorschläge in „Suchen“, „Nachschlagen“, #images, App „News“ und News-Widget in Ländern ohne App „News“ präsentieren Benutzern Vorschläge, die über ihre Geräte hinausgehen und von Quellen stammen wie etwa Wikipedia, dem iTunes Store, lokalen Nachrichten, den Ergebnissen der App „Karten“ und dem App Store. Diese Vorschläge werden bereits angeboten, bevor der Benutzer mit dem Schreiben beginnt.

Wenn ein Benutzer mit der Eingabe in der Safari-Adressleiste beginnt, Siri-Vorschläge in „Suchen“ öffnet oder verwendet, „Nachschlagen“ verwendet, #images öffnet, „Suchen“ in der App „News“ verwendet oder das News-Widget in Ländern ohne die App „News“ verwendet, wird der folgende Kontext mit HTTPS verschlüsselt an Apple gesendet, um dem Benutzer relevante Ergebnisse anzuzeigen:

- Eine Kennung, die alle 15 Minuten geändert wird, um die Daten zu schützen
- Die Suchanfrage des Benutzers
- Die wahrscheinlichste vollständige Abfrage basierend auf dem Kontext und früheren Suchvorgängen im lokalen Zwischenspeicher
- Der ungefähre Standort des Geräts, sofern darauf die Ortungsdienste für ortsbasierte Vorschläge aktiviert sind. Wie stark der Standort verschleiert wird, ist abhängig von der ungefähren Benutzerdichte am Standort des Geräts. Beispielsweise erfolgt in ländlichen Gegenden, in denen Benutzer sich geografisch weiter auseinander befinden können, eine stärkere Verschleierung als in Innenstädten, wo sich mehr Benutzer auf engerem Raum aufhalten. Der Benutzer kann in den Systemeinstellungen festlegen, dass keine Ortsdaten an Apple gesendet werden sollen, indem er die Ortungsdienste für ortsabhängige Vorschläge deaktiviert. Wenn die Ortungsdienste deaktiviert sind, kann Apple den ungefähren Standort des Geräts aus der IP-Adresse des Geräts ableiten.
- Die Art des Geräts und ob die Suche in den Siri-Vorschlägen in „Suchen“, Safari, der App „Nachschlagen“, der App „News“ oder der App „Nachrichten“ erfolgt
- Die Art der Verbindung
- Informationen zu den drei zuletzt auf dem Gerät genutzten Apps (für zusätzlichen Kontext in der Suchanfrage). Es werden nur Apps aufgenommen, die in einer von Apple gepflegten Positivliste beliebter Apps aufgeführt sind und in den letzten drei Stunden verwendet wurden.
- Eine Liste beliebter Apps auf dem Gerät
- Regionale Sprache, Sprachumgebung und Einstellungen für die Eingabe
- Wenn das Gerät des Benutzers auf Dienste für Musik- oder Videoabonnements zugreifen kann, können Informationen wie Namen der Abonnementdienste und Arten der Abonnements an Apple gesendet werden. Accountname, Nummer und Passwort des Benutzers werden nicht an Apple gesendet.
- Eine zusammengefasste gesammelte Darstellung der Themen von Interesse

Wenn ein Benutzer ein Ergebnis auswählt oder die App verlässt, ohne ein Ergebnis ausgewählt zu haben, werden einige Informationen an Apple gesendet, um die Qualität künftiger Suchergebnisse zu verbessern. Diese Informationen sind lediglich mit derselben 15-Minuten-Sitzungskennung verknüpft, nicht mit einem bestimmten Benutzer. Das Feedback enthält einige der zuvor beschriebenen Kontextinformationen sowie u. a. folgende Informationen zur Interaktion:

- Die vergangene Zeit zwischen Interaktionen und über das Netzwerk gesendeten Suchanfragen
- Rangliste und Anzeigereihenfolge von Vorschlägen
- Die Kennung von Ergebnis und Aktion, die ausgewählt wurden, wenn es sich nicht um ein lokales Ergebnis handelt, oder die Kategorie des ausgewählten Ergebnisses, wenn das Ergebnis ein lokales Ergebnis ist
- Eine Markierung, die angibt, ob der Benutzer das Ergebnis ausgewählt hat

Apple speichert die Protokolle für Vorschläge mit Anfragen, Kontext und Feedback 18 Monate lang. Ein Teil der Protokolle wird bis zu fünf Jahre aufbewahrt, z. B. Anfragen, Sprachumgebung (Locale), Domain, ungefähre Standort und aggregierte Messwerte.

Unter bestimmten Umständen können Vorschläge Anfragen für häufige Wörter und Ausdrücke an einen qualifizierten Partner weiterleiten, um die Suchergebnisse des Partners erhalten und anzeigen zu können. Apple versieht die Anfragen mit einem Proxy, sodass Partner keine IP-Adressen der Benutzer und kein Feedback zur Suche erhalten. Die Kommunikation mit dem Partner wird mit HTTPS verschlüsselt. Für häufig auftretende Abfragen gibt Apple Stadt, Gerätetyp und Sprache des Clients als Suchkontext an den Partner weiter, um die Suchleistung zu optimieren. In iOS 11 werden Siri-Vorschläge für Anfragen in „Suchen“ nicht an Partner gesendet.

Die folgenden Informationen werden ohne Sitzungskennung protokolliert, um die Leistung von Vorschlägen besser zu verstehen und geografisch sowie übergreifend über verschiedene Netzwerktypen zu optimieren:

- Unvollständige IP-Adresse (ohne das letzte Oktett bei IPv4-Adressen; ohne die letzten 80 Bit bei IPv6-Adressen)
- Der ungefähre Standort
- Die ungefähre Zeit der Anfrage
- Latenz/Transferrate
- Größe der Antwort
- Art der Verbindung
- Sprachumgebung
- Gerätetyp und die anfragende App

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

Der Gerätecode des Benutzers ist standardmäßig eine aus vier Ziffern bestehende PIN. Auf Geräten mit Touch ID oder Face ID muss der Gerätecode mindestens sechs Ziffern umfassen. Bei anderen Geräten liegt die Mindestlänge bei vier Ziffern. Unter „Einstellungen“ > „Touch ID & Code“ kann ein Benutzer über die Option „Eigener alphanumerischer Code“ einen längeren, alphanumerischen Code festlegen. Längere und komplexere Codes sind schwerer zu erraten oder anzugreifen und werden 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 oder Face ID erlauben

Administratoren finden Details zu den einzelnen Richtlinien unter:
<https://help.apple.com/deployment/ios/#/apd4D6A472A-A494-4DFD-B559-D59E63167E43>

Entwickler finden Details zu den einzelnen Richtlinien 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 diese vertrauenswürdige Verbindung, um zusätzliche Funktionen mit dem verbundenen Host (z. B. Synchronisation von Daten) zu aktivieren.

In iOS 9 können Dienste, die eine Kopplung erfordern, erst gestartet werden, nachdem die Gerätesperre durch den Benutzer aufgehoben wurde.

Darüber hinaus setzen unter iOS 10 einige Dienste, darunter die Synchronisierung von Fotos, voraus, dass das Gerät entsperrt wird, damit der Vorgang gestartet werden kann.

Ab iOS 11 werden Dienste nur gestartet, wenn das Gerät vor Kurzem entsperrt wurde.

Für die Kopplung muss der Benutzer das Gerät entsperren und die Anfrage des Hosts annehmen. Ab iOS 11 muss der Benutzer auch den Code eingeben. Daraufhin tauschen Host und Gerät öffentliche 2048-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“ in diesem Dokument). 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. In iOS 9 verfällt ein Datensatz für die Kopplung, wenn er länger als sechs Monate nicht verwendet wurde. Dieser Zeitrahmen wird in iOS 11 auf 30 Tage verkürzt.

Weitere Informationen findest du unter:

<https://support.apple.com/de-de/HT203034>

Bestimmte Dienste (z. B. com.apple.pcapd) sind nur über USB möglich. Der Dienst com.apple.file_relay benötigt außerdem ein von Apple signiertes Konfigurationsprofil, um installiert werden zu können.

In iOS 11 kann das Apple TV das SRP-Protokoll (Secure Remote Password) verwenden, um drahtlos eine Kopplungsbeziehung zu erstellen.

Der Benutzer kann die Liste vertrauenswürdiger Hosts mit der Option „Netzwerkeinstellungen zurücksetzen“ oder „Standort & Datenschutz zurücksetzen“ löschen.

Weitere Informationen findest du unter:

<https://support.apple.com/de-de/HT202778>

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 WLAN- und Datenzugriff verknüpfen. Beispielsweise kann ein Konfigurationsprofil für die E-Mail-Konfiguration 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

Eine aktuelle Liste für Administratoren findest du auf folgender Website:
<https://help.apple.com/deployment/ios/#/cad5370d089>

Eine aktuelle Liste für Entwickler findest du auf folgender Website:
<https://developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/>

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 Eigentümer ihrer iOS-Geräte sind, können Konfigurationsprofile, die ein Gerät fest mit einer MDM-Lösung verbinden, entfernt werden; dabei werden aber alle verwalteten Konfigurationsinformationen, Daten und Apps entfernt.

Benutzer können Konfigurationsprofile mit Apple Configurator 2 direkt auf ihren Geräten installieren oder sie in Safari oder aus einer Mail laden oder von einer MDM-Lösung zusenden lassen. Wenn ein Benutzer ein Gerät im Device Enrollment Program (Programm zur Geräteregistrierung, DEP) oder in Apple School Manager einrichtet, lädt das Gerät ein Profil für die MDM-Registrierung und installiert es.

Mobile Device Management (MDM)

Die MDM-Unterstützung in iOS ermöglicht Unternehmen die sichere Konfiguration und Verwaltung skalierter Implementierungen von iPhone, iPad, Apple TV und Mac im Unternehmen. MDM-Funktionen basieren auf vorhandenen iOS-Technologien wie den Konfigurationsprofilen, der kabellosen Registrierung und dem Apple-Dienst für Push-Benachrichtigungen (APNS). So wird beispielsweise APNS verwendet, um den Ruhezustand eines Geräts zu beenden, damit es über eine sichere Verbindung direkt mit einer MDM-Lösung 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 findest du unter:

<https://www.apple.com/de/iphone/business/it/management.html>

Geteiltes iPad

Die Option „Geteiltes iPad“ ist ein Mehr-Benutzer-Modus für iPad-Implementierungen in Bildungseinrichtungen. Damit können sich Schüler/Studenten ein iPad teilen, ohne Dokumente und Daten teilen zu müssen. Jeder Schüler/Student erhält ein eigenes Benutzerverzeichnis, das als APFS-Datenträger erstellt und durch die Benutzerdaten geschützt wird. Für die Option „Geteiltes iPad“ ist eine verwaltete Apple-ID erforderlich, die von der Bildungseinrichtung ausgegeben wird und deren Kontrolle unterliegt. Schüler/Studenten können sich so an jedem Gerät im Eigentum der Organisation anmelden, das für die Nutzung durch mehrere Personen konfiguriert ist.

Die Daten der Schüler sind in separate Benutzerverzeichnisse in eigenen Datensicherheits-Domains aufgeteilt, von denen jede durch UNIX-Zugriffsrechte und Sandboxing geschützt ist. Bei der Anmeldung eines Schülers wird die verwaltete Apple-ID mithilfe des SRP-Protokolls bei den Identitätsservern von Apple authentifiziert. Ist der Vorgang erfolgreich, wird ein gerätespezifisches vorübergehendes Zugangstoken zugewiesen. Wenn der Schüler das Gerät bereits zuvor verwendet hat, hat er bereits einen lokalen Benutzeraccount, der mit denselben Daten entsperrt wird. Hat der Schüler/Student das Gerät zuvor noch nicht verwendet, werden eine neue UNIX-Benutzerkennung, ein APFS-Datenträger mit dem Benutzerverzeichnis und ein logischer Schlüsselbund hinzugefügt. Wenn das Gerät nicht mit dem Internet verbunden ist (z. B. weil der Schüler/Student auf einem Ausflug ist), kann die Authentifizierung während einer begrenzten Anzahl von Tagen mit dem lokalen Account ausgeführt werden. In diesem Fall können sich nur Benutzer anmelden, die bereits einen lokalen Account haben. Nach dem Ablauf der Zeitbegrenzung müssen sich Schüler/Studenten online authentifizieren, auch wenn bereits ein lokaler Account vorhanden ist.

Nachdem der lokale Account des Schülers/Studenten entsperrt oder erstellt wurde (bei Authentifizierung per Fernzugriff), wird das von den Apple-Servern ausgegebene vorübergehende Token in ein iCloud-Token umgewandelt, das eine Anmeldung bei iCloud erlaubt. Als Nächstes werden die Einstellungen der Schüler/Studenten wiederhergestellt und ihre Dokumente und Daten von iCloud synchronisiert.

Solange die Schülersitzung aktiv ist und das Gerät online bleibt, werden erstellte oder geänderte Dokumente und Daten in iCloud gespeichert. Ein Synchronisierungsmechanismus, der im Hintergrund abläuft, sorgt zudem dafür, dass Änderungen an iCloud gepusht werden, nachdem sich die Schüler abmelden. Nachdem die Synchronisierung im Hintergrund für den Benutzer abgeschlossen ist, wird der APFS-Datenträger des Benutzers deaktiviert und kann nicht mehr aktiviert werden, ohne die Benutzerdaten bereitzustellen.

Wenn ein geteiltes iPad von einer iOS-Version vor 10.3 auf 10.3 (oder neuer) aktualisiert wird, findet einmalig eine Umwandlung des Dateisystems der Datenpartition mit HFS+ in einen APFS-Datenträger statt. Wenn zu diesem Zeitpunkt Benutzerverzeichnisse auf dem System vorhanden sind, verbleiben sie auf dem Hauptdatenträger und werden nicht in einzelne APFS-Datenträger umgewandelt. Wenn sich weitere Schüler/Studenten anmelden, werden ihre Benutzerverzeichnisse ebenfalls auf dem Hauptdatenträger angelegt. Wie zuvor beschrieben werden neue Benutzeraccounts nicht mit eigenen APFS-Datenträgern erstellt, solange nicht alle Benutzeraccounts auf dem Hauptdatenträger gelöscht wurden. Um sicherzustellen, dass den Benutzern der zusätzliche Schutz und die Kontingente von APFS bereitgestellt werden, sollte das iPad deshalb mit einer Neuinstallation in Verbindung mit dem Löschen auf 10.3 (oder neuer) aktualisiert werden. Alternativ können auch alle Benutzeraccounts mit dem MDM-Befehl „Benutzer löschen“ gelöscht werden.

Apple School Manager

Apple School Manager ist ein Dienst für Bildungseinrichtungen, der es diesen Einrichtungen ermöglicht, Inhalte zu kaufen, eine automatische Geräteregistrierung in MDM-Lösungen zu konfigurieren, Accounts für Lernende und Mitarbeiter anzulegen und iTunes U-Kurse einzurichten. Apple School Manager ist über das Web zugänglich und für Technologiemanager und IT-Administratoren sowie für Mitarbeiter und Lehrkräfte konzipiert.

Weitere Informationen zu Apple School Manager findest du unter:
<https://help.apple.com/schoolmanager/>

Geräteregistrierung

Das Programm zur Geräteregistrierung (Device Enrollment Program, DEP) ist Teil von Apple School Manager und Apple-Implementierungsprogrammen und bietet ein schnelles, optimiertes Verfahren, um iOS-Geräte zu implementieren, die eine Organisation direkt bei Apple oder bei teilnehmenden, autorisierten Apple-Vertriebspartnern und Anbietern gekauft hat. iOS-Geräte mit iOS 11 oder neuer können mit Apple Configurator 2 auch nach dem Kauf zum DEP hinzugefügt werden.

Organisationen können Geräte automatisch im MDM registrieren, ohne sie vor der Ausgabe an die Benutzer in die Hand nehmen oder vorbereiten zu müssen. Nach der Registrierung im Programm melden sich Administratoren auf der Programm-Website an und verknüpfen das Programm mit ihrer MDM-Lösung. Die gekauften Geräte können dann über MDM den Benutzern zugewiesen werden. Sobald die Zuweisung zu einem Benutzer erfolgt ist, werden in MDM festgelegte Konfigurationen, Einschränkungen und Steuerungen automatisch installiert. Die gesamte Kommunikation zwischen den Geräten und den Apple-Servern wird mit HTTPS (SSL) verschlüsselt.

Darüber hinaus lässt sich der Konfigurationsprozess für Benutzer weiter vereinfachen, indem bestimmte Schritte im Systemassistenten entfernt werden, sodass die Benutzer schnell mit ihrer Arbeit beginnen können. Administratoren können auch steuern, ob der Benutzer das MDM-Profil von seinem Gerät löschen kann, und außerdem sicherstellen, dass die Geräteeinschränkungen vom ersten Moment an greifen. Nach dem Entpacken und Aktivieren des Geräts wird es im MDM der Organisation registriert und alle verwalteten Einstellungen, Apps und Bücher werden installiert.

Weitere Informationen für Unternehmen findest du unter:
<https://help.apple.com/deployment/business/>

Weitere Informationen für Bildungseinrichtungen findest du unter:
<https://help.apple.com/schoolmanager/>

Hinweis: Die Geräteregistrierung ist nicht in allen Ländern und Regionen verfügbar.

Apple Configurator 2

Neben der Nutzung von MDM erleichtert Apple Configurator 2 für macOS auch die Einrichtung und Vorabkonfiguration von iOS-Geräten und Apple TV vor der Verteilung an die Benutzer. Mit Apple Configurator 2 können Geräte schnell mit Apps, Daten, Einschränkungen und Einstellungen vorkonfiguriert werden.

Apple Configurator 2 erlaubt die Verwendung von Apple School Manager (im Bildungsbereich) oder des Device Enrollment Program (in Unternehmen), um Geräte in einer MDM-Lösung zu registrieren, ohne dass die Benutzer den Systemassistenten verwenden müssen. Apple Configurator 2 kann auch verwendet werden, um iOS-Geräte und Apple TV nach dem Kauf zu Apple School Manager oder zum Device Enrollment Program hinzuzufügen.

Weitere Informationen zu Apple Configurator 2 findest du unter:
<https://help.apple.com/configurator/mac/>

Betreuung

Beim Einrichten eines Geräts kann eine Organisation festlegen, dass das Gerät betreut werden soll. Die Betreuung impliziert, dass sich das Gerät im Eigentum der Organisation befindet und die Konfiguration und die Einschränkungen für das Gerät in noch umfassenderer Weise gesteuert werden können. Die Konfiguration von Geräten als betreute Geräte kann mit Apple School Manager, mit dem Device Enrollment Program (Programm zur Geräteregistrierung, DEP) oder mit Apple Configurator 2 erfolgen. Für die Betreuung eines Geräts muss es gelöscht und das Betriebssystem neu installiert werden.

Weitere Informationen zum Konfigurieren und Verwalten von Geräten mithilfe von MDM oder Apple Configurator 2 findest du unter:
<https://help.apple.com/deployment/ios/>

Einschränkungen

Einschränkungen können von Administratoren aktiviert (oder in einigen Fällen deaktiviert) werden, um zu verhindern, dass Benutzer auf bestimmte Apps, Dienste oder Funktionen des Geräts zugreifen. Einschränkungen werden in einer entsprechenden Payload, die an ein Konfigurationsprofil angehängt wird, an die Geräte gesendet. Einschränkungen können auf Geräte mit iOS, tvOS und macOS angewendet werden. Bestimmte Einschränkungen auf einem verwalteten iPhone können auf eine gekoppelte Apple Watch gespiegelt werden.

Eine aktuelle Liste für IT-Manager kannst du auf der folgenden Website anzeigen: <https://help.apple.com/deployment/ios/#/aüddbd6309354>

Fernlöschen

iOS-Geräte können von einem Administrator oder Benutzer per Fernzugriff gelöscht werden. Die sofortige Fernlöschung wird dadurch erreicht, 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.

Modus „Verloren“

Bei Verlust oder Diebstahl eines Geräts kann ein MDM-Administrator auf betreuten Geräten mit iOS 9.3 (oder neuer) per Fernzugriff den Modus „Verloren“ aktivieren. Wenn der Modus „Verloren“ aktiviert wird, wird der aktuelle Benutzer abgemeldet und das Gerät kann nicht entsperrt werden. Auf dem Bildschirm erscheint eine Mitteilung, die vom Administrator angepasst werden kann. Sie kann beispielsweise eine Telefonnummer für den Finder des Geräts zeigen. Wird das Gerät in den Modus „Verloren“ versetzt, kann der Administrator den aktuellen Standort des Geräts abfragen und optional einen Ton abspielen. Der Modus „Verloren“ kann nur durch einen Administrator deaktiviert und beendet werden. Ist dies der Fall, wird der Benutzer mit einer entsprechenden Mitteilung auf dem Sperrbildschirm oder einem Hinweis auf dem Home-Bildschirm darüber informiert.

Aktivierungssperre

Wenn die Option „Mein iPhone suchen“ aktiviert ist, kann das Gerät nicht neu aktiviert werden, ohne dass die Apple-ID-Anmeldedaten des Eigentümers oder der vorherige Code des Geräts eingegeben werden.

Bei Geräten, die sich im Eigentum einer Organisation befinden, empfiehlt es sich, die Geräte zu betreuen. Dadurch wird die Aktivierungssperre von der Organisation verwaltet und diese muss sich nicht darauf verlassen, dass die einzelnen Benutzer ihre Apple-ID-Anmeldedaten eingeben, um die Geräte neu zu aktivieren.

Eine kompatible MDM-Lösung kann auf betreuten Geräten einen Umgehungscode speichern, wenn die Aktivierungssperre eingeschaltet wird, oder diesen später verwenden, um die Aktivierungssperre automatisch aufzuheben, wenn das Gerät gelöscht und einem neuen Benutzer bereitgestellt werden soll.

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

Bei Geräten, die in Bildungseinrichtungen mit einer über Apple School Manager erstellten, verwalteten Apple-ID verwendet werden, kann die Aktivierungssperre statt mit der Apple-ID des Benutzers mit der Apple-ID eines Administrators verknüpft oder mit dem Umgehungscode des Geräts deaktiviert werden.

Datenschutzeinstellungen

Apple respektiert die Privatsphäre seiner Kunden und hat diverse Einstellungen und Optionen integriert, mit denen 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 Analysedaten aufgenommen werden, die Apple verwendet, um iOS zu verbessern, und dass ortsbasierte Siri-Informationen, ortsbasierter Kontext für Siri-Vorschläge in „Suchen“, lokale Verkehrsbedingungen und wichtige zuvor 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 gehört Zugriff auf:

- Kontakte
- Kalender
- Erinnerungen
- Fotos
- Aktivitätsdaten & Fitness
- Ortungsdienste
- Apple Music
- Deine Musik- und Videoaktivität
- Accounts in sozialen Netzwerken wie Twitter oder Facebook
- Mikrophon
- Kamera
- HomeKit
- Health
- Spracherkennung
- Bluetooth-Freigabe
- Deine Mediathek

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 den von einer MDM-Lösung und den vom Benutzer installierten Apps und Accounts verhindern.

Datenschutzrichtlinie

Die Datenschutzrichtlinie von Apple findest du auf folgender Website:
<https://www.apple.com/legal/privacy/de-ww/>

Apple Security Bounty

Personen, die Apple auf kritische Probleme hinweisen, erhalten von Apple eine Belohnung. Diese Personen müssen durch einen eindeutigen Bericht und einen stichhaltigen Wirksamkeitsnachweis belegen, dass sie Anspruch auf eine solche Belohnung haben. Die gefundene Sicherheitslücke muss die aktuell ausgelieferte iOS-Version betreffen sowie die neueste Hardware, sofern relevant. Die Höhe der Zahlung wird nach der Prüfung durch Apple festgelegt. Zu den Kriterien gehören Neuheit, Entdeckungswahrscheinlichkeit und Grad der erforderlichen Benutzerinteraktion.

Nach Erhalt entsprechender Informationen hat es für Apple oberste Priorität, bestätigte Probleme so schnell wie möglich zu lösen. Apple gewährt den betreffenden Personen dann öffentliche Anerkennung, sofern diese nichts anderes beantragt haben.

Kategorie	Maximale Zahlung (USD)
Sichere Boot-Firmware-Komponenten	200.000 US-Dollar
Extrahieren vertraulicher, durch die Secure Enclave geschützter Materialien	100.000 US-Dollar
Ausführen von beliebigem Code mit Kernel-Rechten	50.000 US-Dollar
Unbefugter Zugriff auf iCloud-Accountdaten auf Apple-Servern	50.000 US-Dollar
Zugriff über einen Sandbox-Prozess auf Benutzerdaten außerhalb dieser Sandbox	25.000 US-Dollar

Fazit

Der Sicherheit verpflichtet

Mit führenden Technologien für Datenschutz und Sicherheit, die dafür entwickelt wurden, persönliche Daten zu schützen, und umfassenden Methoden zur Datensicherheit 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 löschen, das verkauft oder übertragen werden soll. Die Diagnosedaten werden ebenfalls anonym gesammelt.

Apple legt bei der Entwicklung seiner Apps für iOS besonderes Augenmerk auf die Sicherheit. iMessage und FaceTime bieten beispielsweise eine Verschlüsselung zwischen den Clients. Bei Apps von Drittanbietern wird über eine Kombination aus obligatorischer Codesignierung, Sandbox und Berechtigungen dafür gesorgt, dass der Benutzer 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 von iOS optimal nutzen zu 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 and Security Teams).

Weitere Informationen, wie Problembereiche an Apple gesendet und Sicherheitsmitteilungen abonniert werden können, findest du auf folgender Website: <https://www.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.
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.
Datensicherheit	Schutzmechanismen für Dateien und Schlüsselbundobjekte in iOS. Kann sich auch auf APIs beziehen, die Apps zum Schutz von Dateien und Schlüsselbundobjekten verwenden.
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. Du musst dieses iPad wiederherstellen, bevor es mit iTunes verwendet werden kann.“
ECID	Eine eindeutige 64-Bit-Kennung für jeden Prozessor eines iOS-Geräts. Wird ein Anruf auf einem Gerät entgegengenommen, wird das Klingeln von über iCloud gekoppelten Geräten in der Nähe durch eine kurze Ankündigung via Bluetooth Low Energy 4.0 beendet. Die Ankündigungsbytes werden mit derselben Methode wie Handoff-Ankündigungen verschlüsselt. 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.
Hardware sicherheitsmodule (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	<p>Eine Datenstruktur, die zum Speichern von Klassenschlüsselsammlungen verwendet wird. Alle Keybag-Arten (User, Device, Backup, Escrow oder iCloud Backup) besitzen dasselbe Format:</p> <ul style="list-style-type: none"> • Ein Header mit: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – Schlüssel UUID – Klasse (Datensicherheitsklasse der Datei bzw. des Schlüsselbunds) – 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 nutzt die Methode NIST AES Key Wrapping in Übereinstimmung mit RFC 3394.
Low-Level Bootloader (LLB)	Code, der im Rahmen des sicheren Startvorgangs vom Boot-ROM abgerufen wird 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.
Smart Card	Integrierter, eingebetteter Schaltkreis, der eine sichere Identifikation, Authentifizierung und Speicherung der Daten bietet.
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 in der 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, das den Code eines Benutzers in einen kryptografischen Schlüssel konvertiert, 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 nutzt.
XNU	Der Kernel im Zentrum der Betriebssysteme iOS und macOS. Er wird als vertrauenswürdig eingestuft und setzt Sicherheitsmaßnahmen wie Code-Signierung, Sandbox, Überprüfen von Berechtigungen und ASLR durch.

Dokumentrevisionsen

Datum	Zusammenfassung
Januar 2018	<p>Aktualisiert für iOS 11.2</p> <ul style="list-style-type: none">• Apple Pay Cash <p>Aktualisiert für iOS 11.1</p> <ul style="list-style-type: none">• Sicherheitszertifizierungen und -programme• Touch ID/Face ID• Geteilte Notizen• End-to-End-Verschlüsselung in CloudKit• TLS• Apple Pay, Mit Apple Pay im Web zahlen• Siri-Vorschläge• Geteiltes iPad• Weitere Informationen über sicherheitsrelevante Inhalte von iOS 11 findest du unter: https://support.apple.com/de-de/HT208112
Juli 2017	<p>Aktualisiert für iOS 10.3</p> <ul style="list-style-type: none">• System Enclave• Sicherheit von Dateidaten• Keybags• Sicherheitszertifizierungen und -programme• SiriKit• HealthKit• Netzwerksicherheit• Bluetooth• Geteiltes iPad• Modus „Verloren“• Aktivierungssperre• Datenschutzeinstellungen• Weitere Informationen über sicherheitsrelevante Inhalte von iOS 10.3 findest du unter: https://support.apple.com/de-de/HT207617
März 2017	<p>Aktualisiert für iOS 10</p> <ul style="list-style-type: none">• Systemsicherheit• Datensicherheitsklassen• Sicherheitszertifizierungen und -programme• HomeKit, ReplayKit, SiriKit• Apple Watch• WLAN, VPN• Single-Sign-On• Apple Pay, Mit Apple Pay im Web zahlen• Bereitstellung von Kredit-, Debit- und Prepaid-Karten• Safari-Vorschläge• Weitere Informationen über sicherheitsrelevante Inhalte von iOS 10 findest du unter: https://support.apple.com/de-de/HT207143

Datum	Zusammenfassung
Mai 2016	<p>Aktualisiert für iOS 9.3</p> <ul style="list-style-type: none"> • Verwaltung Apple-ID • Zwei-Faktor-Authentifizierung für Apple-IDs • Keybags • Sicherheitszertifizierungen • Modus „Verloren“, Aktivierungssperre • Geschützte Notizen • Apple School Manager, Geteiltes iPad • Weitere Informationen über sicherheitsrelevante Inhalte von iOS 9.3 findest du unter: https://support.apple.com/de-de/HT206166.
September 2015	<p>Aktualisiert für iOS 9</p> <ul style="list-style-type: none"> • Aktivierungssperre für Apple Watch • Coderichtlinien • API-Unterstützung für Touch ID • Datenschutz bei A8 per AES-XTS • Keybags für unbeaufsichtigte Softwareaktualisierungen • Aktualisierungen für Zertifizierung • Modell der Vertrauenswürdigkeit von Unternehmens-Apps • Datenschutz für Safari-Lesezeichen • Transportsicherheit für Apps • VPN-Spezifikation • iCloud Remote-Zugriff für HomeKit • Kunden-/Bonuskarten für Apple Pay, App des Kartenausstellers für Apple Pay • Gerätebasierte Indexierung für Spotlight • iOS-Kopplungsmodell • Apple Configurator 2 • Einschränkungen • Weitere Informationen über sicherheitsrelevante Inhalte von iOS 9 findest du unter: https://support.apple.com/de-de/HT205212

© 2018 Apple Inc. Alle Rechte vorbehalten.

Apple, das Apple-Logo, AirDrop, AirPlay, Apple Music, Apple Pay, Apple TV, Apple Watch, Bonjour, CarPlay, Face ID, FaceTime, Handoff, iMessage, iPad, iPad Air, iPhone, iPod touch, iTunes, iTunes U, Keychain, Lightning, Mac, macOS, OS X, Safari, Siri, Spotlight, Touch ID, watchOS und Xcode sind Marken der Apple Inc., die in den USA und weiteren Ländern eingetragen sind.

HealthKit, HomeKit, SiriKit und tvOS sind Marken der Apple Inc.

AppleCare, App Store, CloudKit, iBooks Store, iCloud, iCloud Drive, iCloud Keychain und iTunes Store sind Dienstleistungsmarken der Apple Inc., die in den USA und weiteren Ländern eingetragen sind.

iOS ist eine Marke oder eingetragene Marke von Cisco in den USA und weiteren Ländern und wird in Lizenz verwendet.

Bluetooth®-Wortmarke und -Logos sind eingetragene Marken der Bluetooth SIG, Inc., die von Apple in Lizenz verwendet werden.

Java ist eine eingetragene Marke von Oracle und/oder ihren Tochtergesellschaften.

Andere hier genannte Produkt- und Herstellernamen sind möglicherweise Marken ihrer jeweiligen Rechtsinhaber. Änderungen der Produktspezifikationen vorbehalten.

Januar 2018