Apple-plattformsikkerhet

Mai 2021

Innhold

Introduksjon til Apple-plattformsikkerhet

Apple bygger inn sikkerhet i kjernen av plattformene sine. Apple bygger videre på erfaringene fra å ha laget verdens mest avanserte mobiloperativsystem og har laget sikkerhetsarkitekturer som oppfyller de unike kravene som stilles til hver mobil, klokke, stasjonære enhet og hvert hjem.

Alle Apple-enheter kombinerer *maskinvare*, *programvare* og *tjenester* som er utviklet for å fungere sammen for maksimal sikkerhet og en åpen brukeropplevelse, med hensyn til det endelige målet om å beskytte personopplysninger. For eksempel driver Apples spesialutviklede sikkerhetsmaskinvare viktige sikkerhetsfunksjoner. Programvarebeskyttelser arbeider for å holde operativsystemet og tredjepartsapper beskyttet. Tjenester sørger for en mekanisme for sikre programvareoppdateringer til riktig tid, driver et beskyttet økosystem for apper og tilrettelegger for sikker kommunikasjon og betalinger. Det fører til at Apple-enheter ikke bare beskytter enheten og dataene, men hele økosystemet inkludert alt brukerne gjør lokalt, på nettverk og med viktige internettjenester.

Vi designer produktene våre til å være enkle, intuitive og kraftige – på samme måte som vi designer dem til å være sikre. Viktige sikkerhetsfunksjoner, som maskinvarebasert enhetskryptering, kan ikke deaktiveres ved et uhell. Andre funksjoner, for eksempel Touch ID og Face ID, forbedrer brukeropplevelsen ved å gjøre det enklere og mer intuitivt å gjøre enheten sikker. Og fordi mange av disse funksjonene er slått på som standard, trenger ikke brukere eller IT-avdelinger å utføre omfattende konfigurasjoner.

Denne dokumentasjonen inneholder detaljert informasjon om hvordan sikkerhetsteknologi og -funksjoner er implementert i Apple-plattformer. Den hjelper også organisasjoner med å kombinere sikkerhetsteknologi og -funksjoner på Apple-plattformene med egne regler og prosedyrer, slik at de kan dekke bedriftens spesifikke sikkerhetsbehov.

Innholdet er delt inn i følgende emner:

- **Maskinvaresikkerhet og biometri:** Brikkene og maskinvaren som danner grunnlaget for sikkerhet på Apple-enheter, inkludert Secure Enclave, en dedikert AES-kryptografimotor, Touch ID og Face ID
- **Systemsikkerhet:** De integrerte maskinvare- og programvarefunksjonene som sørger for trygg oppstart, oppdatering og kontinuerlig drift av Apples operativsystemer.
- **Kryptering og databeskyttelse:** Arkitekturen og utformingen som beskytter brukerdata hvis enheten er blitt borte eller stjålet, eller hvis en uautorisert person eller prosess forsøker å bruke eller endre enheten.
- **Appsikkerhet:** Programvaren og tjenestene som leverer et trygt økosystem for apper, slik at apper kan kjøres sikkert og uten å risikere plattformintegriteten.
- **Sikkerhetstjenester:** Apples identifikasjonstjenester, passordadministrering, betalinger, kommunikasjon og å finne mistede enheter.

- **Nettverkssikkerhet:** Nettverksprotokoller som følger bransjestandardene og som sørger for sikker autentisering og kryptering ved dataoverføring.
- **Sikkerhet for utviklerrammeverk:** Rammeverk for sikker og privat administrering av hjem og helse, samt utvidelse av Apple-enheten og tjenestefunksjoner til tredjepartsapper.
- **Sikker administrering av enheter:** Metoder som gir adgang til administrasjon av Apple-enheter, bidrar til å forhindre uautorisert bruk og gjør det mulig å fjernslette innholdet på en enhet hvis den blir borte eller stjålet.

Et engasjement for sikkerhet

Apple er opptatt av å hjelpe til med å beskytte kundene gjennom ledende personvern- og sikkerhetsteknologi, utformet for å verne personopplysninger, samt omfattende metoder, for å bidra til å beskytte bedriftsinformasjon i et bedriftsmiljø. Ved å tilby sikkerhetsbelønning fra Apple, belønner Apple personer for jobben de gjør ved å avdekke sårbarheter. Se mer informasjon om programmet og belønningskategorier på https://developer.apple.com/security-bounty/.

Vi har til enhver tid et eget sikkerhetsteam som tar hånd om alle Apple-produktene. Teamet utfører sikkerhetskontroller og testing av produkter, både under utvikling og etter lansering. Apple-teamet leverer også sikkerhetsverktøy og -opplæring og følger aktivt med på om det kommer trusler og rapporter om nye sikkerhetsproblemer. Apple er medlem av Forum of Incident Response and Security Teams (FIRST).

Apple fortsetter å flytte grensene for hva som er mulig innen sikkerhet og personvern. I år bruker Apple-enheter i alle produktserier med Apple SoC, fra Apple Watch til iPhone og iPad og nå Mac, spesialutviklede brikker til både effektiv beregning og sikkerhet. Apple-brikker danner grunnlaget for sikker oppstart, Touch ID, Face ID og databeskyttelse, i tillegg til systemintegritetsfunksjoner som er nye på Mac, inkludert Kernel Integrity Protection, Pointer Authentication Codes (PAC-er) og Raske rettighetsrestriksjoner. Disse integritetsfunksjonene hjelper med å forhindre vanlige angrepsteknikker som går etter minnet, manipulerer instruksjoner og bruker javaskript på nettet. De jobber sammen for å sørge for at skaden som angriperkoden klarer å gjøre hvis den i det hele tatt får kjørt, reduseres dramatisk.

For å få mest mulig ut av de omfattende sikkerhetsfunksjonene i plattformene våre, oppfordres organisasjoner til å gjennomgå IT- og sikkerhetsregelsett for å sikre full utnyttelse av alle lagene med sikkerhetsteknologi som plattformene tilbyr.

Du finner mer informasjon om hvordan du rapporterer problemer til Apple og abonnerer på sikkerhetsvarsler på: Rapporter en sikkerhets- eller personvernssårbarhet.

Apple mener at personvern er en grunnleggende menneskerett og har en rekke innebygde kontroller og valg som gir brukere muligheten til å bestemme når og hvordan apper bruker informasjonen deres, samt hvilken informasjon som brukes. Hvis du vil vite mer om Apples tilnærming til personvern, kontroller for personvern på Apple-enheter og Apples retningslinjer for personvern, kan du gå til https://www.apple.com/no/privacy**.**

Merk: Med mindre annet er oppgitt dekker denne dokumentasjonen følgende versjoner av operativsystemene: iOS 14.5, iPadOS 14.5, macOS 11.3, tvOS 14.5 og watchOS 7.4.

Maskinvaresikkerhet og biometri

Oversikt over maskinvaresikkerhet

For at programvare skal være sikker, må den støtte seg på maskinvare med innebygd sikkerhet. Det er derfor Apple-enheter som kjører iOS, iPadOS, macOS, watchOS eller tvOS, har sikkerhetsfunksjoner integrert i chipene. Disse funksjonene inkluderer en prosessor som driver systemsikkerhetsfunksjoner, i tillegg til kretser som er dedikert til sikkerhetsfunksjoner. Sikkerhetsfokusert maskinvare følger prinsippet om å støtte begrensede og tydelig definerte funksjoner for å kunne minimere angrepsflaten. Slike komponenter inkluderer en oppstart-ROM, som danner en «root of trust» i maskinvaren for sikker oppstart, dedikerte AES-motorer for effektiv og sikker kryptering og dekryptering samt en Secure Enclave. *Secure Enclave* er et System on Chip (SoC) som er inkludert på alle nyere iPhone-, iPad-, Apple Watch-, Apple TV- og HomePod-enheter, på alle Apple Silicon-baserte Macer og på Macer med Apple T2-sikkerhetsbrikken. Selve Secure Enclave følger samme designprinsipp som SoC-en gjør, med egen oppstart-ROM og AES-motor. Secure Enclave danner også grunnlaget for sikker generering og lagring av nøklene som er nødvendig for å kryptere lagrede data, og den beskytter og evaluerer de biometriske dataene for Touch ID og Face ID.

Lagringskryptering må være raskt og effektivt. Samtidig kan det ikke eksponere data (eller *nøkkelmaterialet*) som brukes til å etablere kryptografiske nøkkelforhold. AES-maskinvaremotoren løser dette problemet ved å utføre rask, integrert kryptering og dekryptering *når filer skrives eller leses*. En spesialkanal fra Secure Enclave gir nødvendig nøkkelmateriale til AES-motoren uten å eksponere denne informasjonen for applikasjonsprosessoren (eller prosessoren) eller hele operativsystemet. Dette bidrar til å sikre at Apple-teknologiene databeskyttelse og FileVault beskytter brukernes filer uten å avsløre krypteringsnøkler med lang levetid.

Apple har designet sikker oppstart for å beskytte de laveste nivåene av programvare mot å tukles med, og at kun godkjent operativsystemprogramvare fra Apple lastes inn ved oppstart. Sikker oppstart starter i uforanderlig kode som kalles oppstart-ROM, som implementeres under produksjon av Apple SoC-en og er kjent som maskinvarens *root of trust*. På Macer med en T2-brikke, starter godkjenning for sikker macOS-oppstart med T2-brikken. (Både T2-brikken og Secure Enclave kjører i tillegg sine egne sikre oppstartsprosesser ved hjelp av sin egen atskilte oppstart-ROM. Dette er en nøyaktig parallell til hvordan sikker oppstart utføres på brikkene i A-serien og M1-brikkene.)

Secure Enclave behandler også fingeravtrykk og ansiktsinformasjon fra Touch ID- og Face ID-sensorer i Apple-enheter. Dette sørger for sikker autentisering samtidig som

biometriske data er beskyttet. Dette gjør også at brukere kan dra nytte av sikkerheten til lengre og mer komplekse koder og passord i kombinasjon med rask autentisering for tilgang eller kjøp.

Apple SoC-sikkerhet

Apple-designede brikker danner en felles arkitektur på tvers av alle Apple-produkter og driver nå Mac og iPhone, iPad, Apple TV og Apple Watch. I over ti år har Apples fremragende brikkedesignteam bygget og finpusset Apples System on Chip (SoC). Resultatet er en skalerbar arkitektur utviklet for alle enhetene som er ledende i bransjen innen sikkerhetsfunksjoner. Dette fellesgrunnlaget for sikkerhetsfunksjoner er kun mulig fra et selskap som designer egne brikker som fungerer sammen med programvaren.

Apple Silicon er utformet og produsert spesielt for å muliggjøre systemsikkerhetsfunksjonene beskrevet under.

Funksjon	A10	A11, S3	A12, S4	A13, S5	A14, S6	M1
Kernel Integrity Protection	✓	✓	✓	✓	✓	✓
Raske rettighetsrestriksjoner		✓	✓	✓	✓	✓
System Coprocessor Integrity Protection			✓	✓	✓	✓
Pointer Authentication Codes			✓	✓	✓	✓
Sidebeskyttelseslag		✓	✓	✓	✓	Se notat nedenfor.

Merk: PPL (Page Protection Layer) krever at plattformen *kun* kjører signert og godkjent kode. Dette er en sikkerhetsmodell som ikke er gjeldende for macOS.

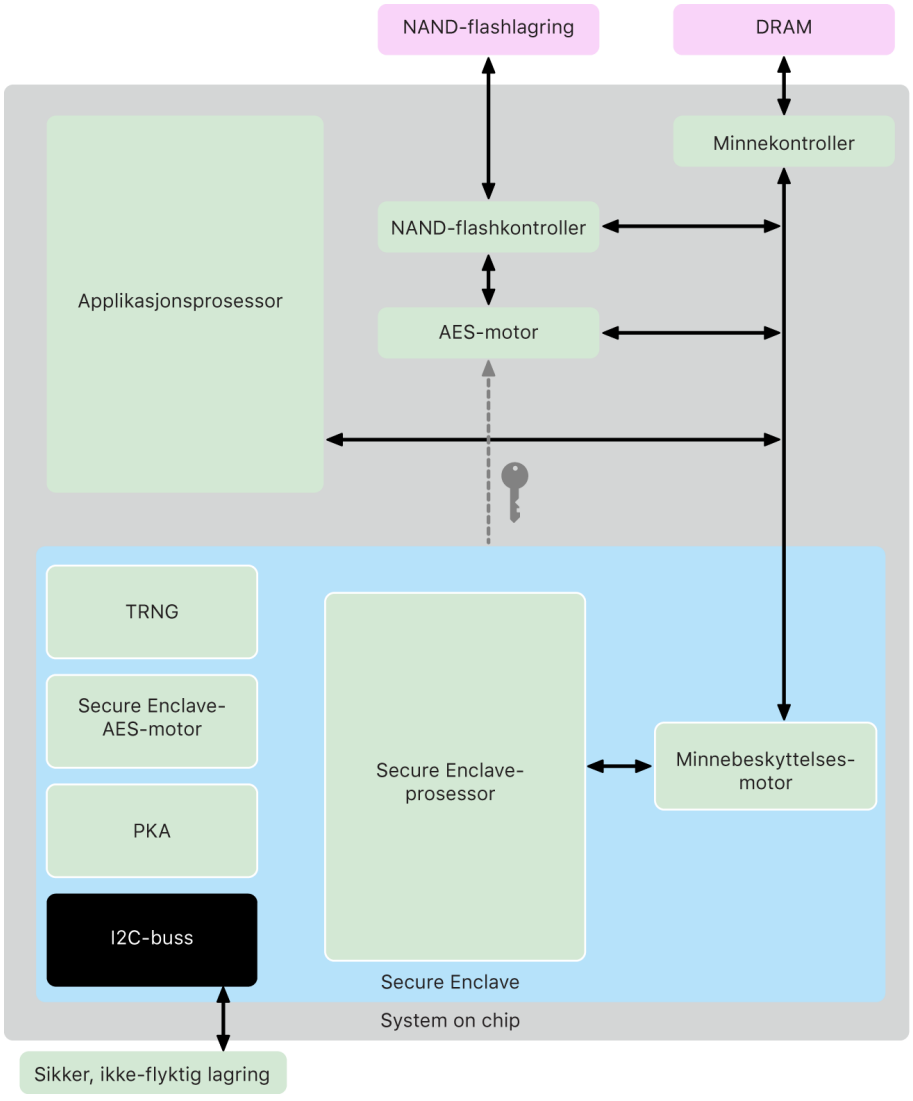
Apple-designede brikker muliggjør også spesifikt databeskyttelsesfunksjonene beskrevet under.

Funksjon	A10	A11, S3	A12, S4	A13, S5	A14, M1, S6
Sealed Key Protection (SKP)	✓	✓	✓	✓	✓
recoveryOS – alle databeskyttelsesklasser er beskyttet	✓	✓	✓	✓	✓
Alternative oppstarter av DFU, diagnostikk og oppdatering – databeskyttelse i klasse A, B og C			✓	✓	✓

Secure Enclave

Oversikt

Secure Enclave er et reservert, sikkert undersystem som er integrert i Apples System on Chip (SoC-er). Secure Enclave er isolert fra hovedprosessoren for å gi et ekstra sikkerhetslag og er utviklet for å holde sensitive brukerdata sikre selv når kjernen til applikasjonsprosessoren kompromitteres. Den følger samme designprinsipper som SoC ved å ha en oppstart-ROM for å etablere en «root of trust» i maskinvaren, en AES-motor for effektive og sikre kryptografiske operasjoner og beskyttet minne. Selv om Secure Enclave ikke inkluderer lagring, har den likevel en mekanisme for å lagre informasjon sikkert på tilkoblet lagring atskilt fra NAND-flash brukt av applikasjonsprosessoren og operativsystemet.



Secure Enclave-komponentene.

Secure Enclave er en maskinvarefunksjon i de fleste versjoner av iPhone, iPad, Mac, Apple TV, Apple Watch og HomePod, det vil si:

- iPhone 5s eller nyere
- iPad Air eller nyere
- MacBook Pro med Touch Bar (2016 og 2017) med Apple T1-brikken
- Intel-baserte Macer med Apple T2-sikkerhetsbrikken
- Mac-maskiner med Apple-chiper
- Apple TV HD eller nyere
- Apple Watch Series 1 eller nyere
- HomePod og HomePod mini

Secure Enclave-prosessor

Secure Enclave-prosessoren leverer den primære datakraften for Secure Enclave. Secure Enclave-prosessoren er utelukkende dedikert til bruk av Secure Enclave for å gi den sterkeste isolasjonen. Dette bidrar til å forhindre angrep fra sidekanaler som er avhengig av at ondsinnet programvare deler den samme utførende kjernen som målprogramvaren under angrep.

Secure Enclave-prosessoren kjører en Apple-tilpasset versjon av L4-mikrokjernen. Den er laget for å fungere effektivt ved en lavere klokkefrekvens, noe som bidrar til å beskytte den mot tids- og effektangrep. Secure Enclave-prosessoren, som starter med A11 og S4, inkluderer en minnebeskyttelsesmotor og krypteringsminne med anti-repetisjonsfunksjoner, sikker oppstart, en dedikert generator for tilfeldige tall og en egen AES-motor.

Minnebeskyttelsesmotor

Secure Enclave drives fra et reservert område av enhetens DRAM-minne. Flere sikkerhetslag isolerer det Secure Enclave-beskyttede minnet fra applikasjonsprosessoren.

Når enheten starter, genererer oppstart-ROM for Secure Enclave en tilfeldig, kortvarig minnebeskyttelsesnøkkel for minnebeskyttelsesmotoren. Når Secure Enclave skriver til det avgrensede minneområdet, krypterer minnebeskyttelsesmotoren minneblokken ved hjelp av AES i Mac XEX-modus (xor-encrypt-xor) og beregner en CMAC-autentiseringsetikett (Cipher-based Message Authentication Code) for minnet. Minnebeskyttelsesmotoren lagrer autentiseringsetiketten sammen med det krypterte minnet. Når Secure Enclave leser minnet, verifiserer minnebeskyttelsesmotoren autentiseringsetiketten. Hvis autentiseringsetiketten samsvarer, dekrypterer minnebeskyttelsesmotoren minneblokken. Hvis etiketten ikke samsvarer, signaliserer minnebeskyttelsesmotoren om en feil til Secure Enclave. Etter en minneautentiseringsfeil slutter Secure Enclave å akseptere forespørsler frem til systemet startes på nytt.

Fra og med Apple A11- og S4-SoC-er legger minnebeskyttelsesmotoren til repetisjonsbeskyttelse for Secure Enclave-minne. For å bidra til å hindre repetisjon av sikkerhetskritiske data lagrer minnebeskyttelsesmotoren en nonce-verdi for minneblokken sammen med autentiseringsetiketten. Nonce-verdien brukes som en ekstra finjustering for CMAC-autentiseringsetiketten. Nonce-verdiene for alle minneblokker beskyttes ved hjelp av et integritetstre som er rotfestet i separat SRAM i Secure Enclave. For skriving *oppdaterer* minnebeskyttelsesmotoren nonce-verdien og hvert nivå av integritetstreet opp til SRAM. For lesing *verifiserer* minnebeskyttelsesmotoren nonce-verdien og hvert nivå av integritetstreet opp til SRAM. Manglende samsvar av nonce-verdier håndteres likt som manglende samsvar av autentiseringsetiketter.

På Apple A14, M1 og nyere SoC-er støtter minnebeskyttelsesmotoren to kortvarige minnebeskyttelsesnøkler. Den første brukes for data som er privat for Secure Enclave, og den andre brukes for data som er delt med Secure Neural Engine.

Minnebeskyttelsesmotoren er integrert med og fungerer transparent med Secure Enclave. Secure Enclave leser og skriver minne som om det var en vanlig ukryptert DRAM, mens en observatør utenfor Secure Enclave kun ser den krypterte og autentiserte versjonen av minnet. Resultatet er sterk minnebeskyttelse uten å gå på kompromiss med ytelse eller programvarekompleksitet.

Oppstart-ROM for Secure Enclave

Secure Enclave inkluderer en separat oppstart-ROM for Secure Enclave. I likhet med oppstart-ROM-en for applikasjonsprosessoren, er oppstart-ROM-en for Secure Enclave en uforanderlig kode som etablerer maskinvarens «root of trust» for Secure Enclave.

Ved oppstart tilordner iBoot et separat minneområde til Secure Enclave. Før minnet brukes, initialiserer oppstart-ROM for Secure Enclave minnebeskyttelsesmotoren for å gi kryptografisk beskyttelse av det Secure Enclave-beskyttede minnet.

Applikasjonsprosessoren sender deretter sepOS-diskfilen til oppstart-ROM for Secure Enclave. Når sepOS-diskfilen er kopiert til det Secure Enclave-beskyttede minnet, sjekker oppstart-ROM for Secure Enclave den kryptografiske hashen og signaturen til diskfilen for å verifisere at sepOS er autorisert til å kjøre på enheten. Hvis sepOS-diskfilen er tilstrekkelig godkjent til å kjøre på enheten, overfører oppstart-ROM for Secure Enclave kontrollen til sepOS. Hvis signaturen ikke er gyldig, er oppstart-ROM for Secure Enclave utviklet for å forhindre all videre bruk av Secure Enclave frem til neste nullstilling av brikken.

På Apple A10 og nyere SoC-er låser oppstart-ROM for Secure Enclave en hash av sepOS i et separat register dedikert til dette formålet. Public Key Accelerator bruker denne hashen for nøkler bundet av operativsystemet.

Oppstartsovervåking for Secure Enclave

På Apple A13 og nyere SoC-er, har Secure Enclave en oppstartsovervåking som er utviklet for å sikre sterkere integritet på hashen av det startede sepOS-et.

Ved oppstart av systemet bidrar Secure Enclave-prosessorens SCIP-konfigurasjon (System Coprocessor Integrity Protection) til å forhindre at Secure Enclave-prosessoren kjører noen som helst kode bortsett fra oppstart-ROM for Secure Enclave. Oppstartsovervåking bidrar til å forhindre at Secure Enclave endrer SCIP-konfigurasjonen direkte. For å gjøre det lastede sepOS-et kjørbart, sender oppstart-ROM for Secure Enclave en forespørsel til oppstartsovervåkingen med adressen og størrelsen på det lastede sepOS-et. Når oppstartsovervåkingen mottar forespørselen, nullstiller den Secure Enclave-prosessoren, hasher det lastede sepOS-et, oppdaterer SCIP-innstillingene for å tillate kjøring av det lastede sepOS-et og starter kjøring i den nylig lastede koden. Når systemet fortsetter å starte opp, brukes denne samme prosessen hver gang en ny kode gjøres kjørbar. Hver gang oppdaterer oppstartsovervåkingen en kjørehash av oppstartsprosessen. Oppstartsovervåking inkluderer også kritiske sikkerhetsparametere i kjørehashen.

Når oppstarten fullføres, sluttfører oppstartsovervåkingen kjørehashen og sender den til Public Key Accelerator for bruk med nøkler bundet av operativsystemet. Denne prosessen er designet slik at operativsystemnøkkelbindingen ikke kan omgås selv med en sårbarhet i oppstart-ROM for Secure Enclave.

True Random Number Generator

True Random Number Generator (TRNG) brukes til å generere sikre, tilfeldige data. Secure Enclave bruker TRNG når den genererer en tilfeldig kryptografisk nøkkel, tilfeldig nøkkelinitialverdi eller annen entropi. TRNG er basert på flere ringoscillatorer som etterbehandles med CTR_DRBG (en algoritme basert på blokk-koder i tellemodus).

Rotkryptografinøkler

Secure Enclave inkluderer en rotkryptografinøkkel for unik ID (UID). UID er unik for hver individuelle enhet og er ikke knyttet til noen annen identifikator på enheten.

En tilfeldig generert UID forenes i SoC ved produksjon. Fra og med A9-SoC-er genereres UID-en av Secure Enclave TRNG ved produksjon og skrives til konfigureringene ved hjelp av en programvareprosess som kjører kun i Secure Enclave. Denne prosessen beskytter UID-en fra å være synlig utenfor enheten under produksjon, og er derfor ikke er tilgjengelig for tilgang eller lagring av Apple eller noen av leverandørene.

sepOS bruker UID-en til å beskytte enhetsspesifikke hemmeligheter. UID-en gjør at data kan knyttes kryptografisk til en bestemt enhet. For eksempel inneholder nøkkelhierarkiet som beskytter filsystemet, UID-en. Så hvis den interne SSD-lagringen flyttes fysisk fra én enhet til en annen, blir filene utilgjengelige. Andre beskyttede enhetsspesifikke hemmeligheter inkluderer Touch ID- eller Face ID-data. På en Mac er det kun fullstendig intern lagring knyttet til AES-motoren som har dette krypteringsnivået. For eksempel: verken eksterne lagringsenheter som kobles til via USB, eller PCIe-basert lagring lagt til Mac Pro fra 2019, krypteres på denne måten.

Secure Enclave har også en enhetsgruppe-ID (GID), som er felles for alle enheter som bruker en gitt SoC (for eksempel alle enheter som bruker Apple A14-SoC, deler samme GID).

UID-en og GID-en er ikke tilgjengelige via JTAG eller andre feilsøkingsgrensesnitt.

AES-motor for Secure Enclave

AES-motoren for Secure Enclave er en maskinvareblokk som brukes til å utføre symmetrisk kryptografi basert på AES-koden. AES-motoren er designet for å motstå informasjonslekkasje ved hjelp av timing og Static Power Analysis (SPA). Fra og med A9 SoC inkluderer også AES-motoren DPA-mottiltak (Dynamic Power Analysis).

AES-motoren støtter maskinvare- og programvarenøkler. Maskinvarenøkler avledes fra Secure Enclaves UID eller GID. Disse nøklene blir værende i AES-motoren og er ikke synlige selv for sepOS-programvare. Selv om programvare kan be om krypterings- og dekrypteringsoperasjoner med maskinvarenøklene, kan den ikke trekke ut nøklene.

På Apple A10 og nyere SoC-er inkluderer AES-motoren låsbare seed-biter som utleder nøkler avledet fra UID-en eller GID-en. Dette gjør det mulig å betinge datatilgang på

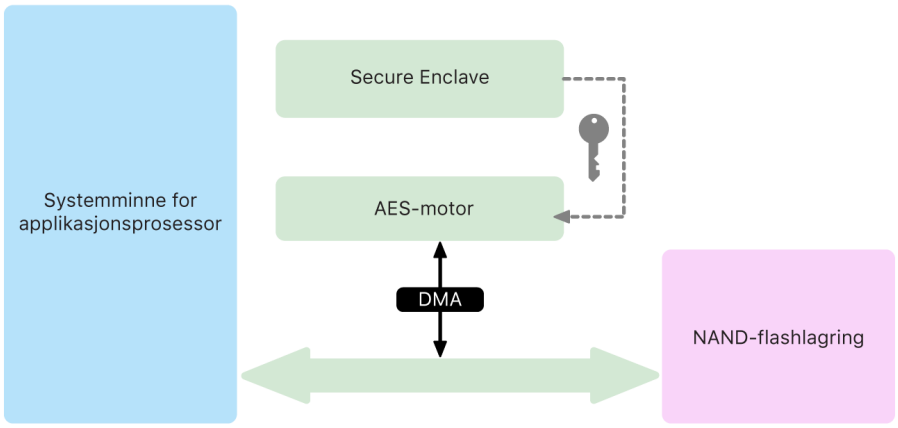
enhetens driftsmodus. Låsbare seed-biter brukes for eksempel til å avslå tilgang til passordbeskyttede data ved oppstart fra DFU-modus (Device Firmware Update). Du finner mer informasjon om dette under Koder og passord.

AES-motor

Alle Apple-enheter med Secure Enclave har også en dedikert AES256-krypteringsmotor («AES-motor») innebygd i banen for direkte minnetilgang (DMA) mellom NAND-flashlagringen (ikke-flyktig) og systemets hovedminne, som gir svært effektiv filkryptering. På A9-prosessorer eller nyere prosessorer i A-serien ligger undersystemet med flashlagringen på en isolert databuss som kun gis tilgang til minne som inneholder brukerdata via DMA-krypteringsmotoren.

Ved oppstart genererer sepOS en kortvarig innpakningsnøkkel ved bruk av TRNG. Secure Enclave sender denne nøkkelen til AES-motoren gjennom separate ledere som er utviklet for å forhindre at programvare utenfor Secure Enclave har tilgang til den. sepOS kan deretter bruke den kortvarige innpakkingsnøkkelen til å pakke inn filnøkler til bruk i filsystemdriveren til applikasjonsprosessoren. Når filsystemdriveren leser eller skriver en fil, sender den den innpakkede nøkkelen til AES-motoren, som pakker ut nøkkelen. AES-motoren eksponerer aldri den utpakkede nøkkelen for programvare.

Merk: AES-motoren er en komponent som er atskilt fra både Secure Enclave og AES-motoren for Secure Enclave, men driften er nært knyttet til Secure Enclave som vist nedenfor.



AES-motoren støtter sanntidskryptering på DMA-banen for effektiv kryptering og dekryptering av data når det skrives og leses til lagring.

Public Key Accelerator

Public Key Accelerator (PKA) er en maskinvareblokk som brukes til å utføre asymmetriske kryptografioperasjoner. PKA støtter signerings- og krypteringsalgoritmer for RSA og ECC (elliptisk kurve-kryptografi). PKA er utformet for å motstå informasjonslekkasje ved hjelp av timing- og sidekanalsangrep, som SPA og DPA.

PKA støtter programvare- og maskinvarenøkler. Maskinvarenøkler avledes fra Secure Enclaves UID eller GID. Disse nøklene blir værende i PKA og er ikke synlige selv for sepOS-programvare.

PKAs krypteringsimplementeringer har siden A13 SoC blitt bevist å være matematisk riktige ved bruk av formelle verifiseringsteknikker.

På Apple A10 og nyere SoC-er støtter PKA nøkler bundet av operativsystemet, også kalt Sealed Key Protection eller SKP. Disse nøklene genereres ved å bruke en kombinasjon av enhetens UID og hashen av sepOS-et som kjører på enheten. Hashen gis av oppstart-ROM for Secure Enclave, eller av oppstartsovervåking for Secure Enclave på Apple A13 og nyere SoC-er. Disse nøklene brukes også til å verifisere sepOS-versjonen når det sendes forespørsler til enkelte Apple-tjenester, og de brukes også til å forbedre sikkerheten til kodebeskyttede data ved å bidra til å forhindre tilgang til nøkkelmateriale hvis kritiske endringer gjøres i systemet uten brukergodkjenning.

Sikker, ikke-flyktig lagring

Secure Enclave er utstyrt med en separat, sikker, ikke-flyktig lagringsenhet. Den sikre, ikke-flyktige lagringen er koblet til Secure Enclave ved hjelp av en dedikert I2C-buss, slik at den kun er tilgjengelig for Secure Enclave. Alle krypteringsnøkler for brukerdata er rotfestet i entropi lagret i Secure Enclaves ikke-flyktige lagring.

I enheter med A12-SoC, S4-SoC og nyere SoC-er kobles Secure Enclave sammen med en komponent for sikker lagring for entropilagring. Selve komponenten for sikker lagring er utviklet med uforanderlig ROM-kode, en maskinvarebasert generator for tilfeldige tall, en unik, kryptografisk enhetsnøkkel, kryptografimotorer og gjenkjenning av fysisk manipulasjon. Secure Enclave og komponenten for sikker lagring kommuniserer ved hjelp av en kryptert og autentisert protokoll som sørger for eksklusiv tilgang til entropien.

Enheter som ble lansert for første gang høsten 2020 eller nyere er utstyrt med en komponent for sikker lagring (andre generasjon). Komponenten for sikker lagring (andre generasjon) legger til tellerlåskasser. Hver tellerlåskasse lagrer et 128-bit-salt, en 128-bit-kodeverifikator, en 8-bit-teller og en 8-bit-verdi for maksimalt antall forsøk. Tilgang til tellerlåskassene bruker en kryptert og autentisert protokoll.

Tellerlåskasser inneholder entropien som trengs for å låse opp kodebeskyttede brukerdata. For å få tilgang til brukerdataene må den parede Secure Enclave avlede riktig kodeentropiverdi fra brukerens kode og Secure Enclaves UID. Brukerens kode kan ikke læres gjennom opplåsingsforsøk som sendes fra en kilde annet enn den parede Secure Enclave. Hvis grensen for kodeforsøk overstiges (for eksempel 10 forsøk på iPhone), vil de kodebeskyttede dataene slettes i sin helhet fra komponenten for sikker lagring.

For å opprette en tellerlåskasse sender Secure Enclave kodeentropiverdien og verdien for maksimalt antall forsøk til komponenten for sikker lagring. Komponenten for sikker lagring genererer saltverdien ved hjelp av generatoren for tilfeldige tall. Deretter avleder den en kodeverifiseringsverdi og en låskasseentropiverdi fra den gitte kodeentropien, den unike kryptografiske nøkkelen til komponenten for sikker lagring og saltverdien.

Komponenten for sikker lagring initialiserer tellerlåskassen med en telling på 0, den gitte verdien for maksimalt antall forsøk, den avledede kodeverifiseringsverdien og saltverdien. Komponenten for sikker lagring returnerer deretter den genererte låskasseentropiverdien til Secure Enclave.

For å hente låskasseentropiverdien fra en tellerlåskasse senere sender Secure Enclave kodeentropien til komponenten for sikker lagring. Komponenten for sikker lagring øker først telleren for låskassen trinnvis. Hvis den trinnvis økte telleren overskrider verdien for maksimalt antall forsøk, ødelegger komponenten for sikker lagring tellerlåskassen. Hvis telleren for maksimalt antall forsøk ikke er nådd, forsøker komponenten for sikker lagring å avlede kodeverifiseringsverdien og låskasseentropiverdien med samme algoritme som ble brukt til å opprette den tellerlåskassen. Hvis den avledede kodeverifiseringsverdien samsvarer med den lagrede kodeverifiseringsverdien, returnerer komponenten for sikker lagring låskasseentropiverdien til Secure Enclave og nullstiller telleren til 0.

Nøklene som brukes for å få tilgang til passordbeskyttede data, er rotfestet i entropien lagret i tellerlåskasser. Du finner mer informasjon om dette under Oversikt over databeskyttelse.

Den sikre, ikke-flyktige lagringen brukes til alle anti-repetisjonstjenester i Secure Enclave. Anti-repetisjonstjenester på Secure Enclave brukes til tilbakekalling av data over hendelser som merker anti-repetisjonsgrenser, inkludert, men ikke begrenset til, følgende:

- endring av passord
- aktivering eller deaktivering av Touch ID eller Face ID
- legge til eller fjerne et Touch ID-fingeravtrykk eller Face ID-ansikt
- nullstilling av Touch ID eller Face ID
- legge til eller fjerne et Apple Pay-kort
- sletting av alt innhold og alle innstillinger

På arkitekturer som ikke har en komponent for sikker lagring, brukes EEPROM (Electrically Erasable Programmable Read-Only Memory) for å gi tjenester for sikker lagring til Secure Enclave. Akkurat som komponentene for sikker lagring, er EEPROM tilknyttet og kun tilgjengelig fra Secure Enclave, men det inneholder ikke dedikerte funksjoner for maskinvaresikkerhet og garanterer heller ikke eksklusiv tilgang til entropi (bortsett fra de fysiske tilknytningsegenskapene) eller tellerlåskassefunksjonalitet.

Secure Neural Engine

På enheter med Face ID konverterer Secure Neural Engine 2D-bilder og dybdekart til en matematisk representasjon av en brukers ansikt.

På A11 til og med A13 SoC-er er Secure Neural Engine integrert i Secure Enclave. Secure Neural Engine bruker direkte minnetilgang (DMA) for høy ytelse. En IOMMU (Input-Output Memory Management Unit) som styres av sepOS-kjernen, begrenser denne direkte tilgangen til autoriserte minneområder.

Fra og med A14 og M1 implementeres Secure Neural Engine som en sikker modus i applikasjonsprosessorens Neural Engine. En dedikert maskinvaresikkerhetskontroller bytter mellom applikasjonsprosessoroppgaver og Secure Enclave-oppgaver og nullstiller Neural Engine-tilstanden for hver overgang for å holde Face ID-data sikre. En dedikert

motor anvender minnekryptering, autentisering og tilgangskontroll. Samtidig bruker den en separat kryptografisk nøkkel og et separat minneområde til å begrense Secure Neural Engine til autoriserte minneområder.

Effekt- og klokkeovervåking

All elektronikk er utformet for å operere innenfor et begrenset spennings- og frekvensområde. Når den opererer utenfor dette området, kan elektronikken feile og sikkerhetskontrollene kan omgås. For å bidra til å sikre at spenningen og frekvensen er innenfor et trygt område, er Secure Enclave designet med overvåkingskretser. Disse overvåkingskretsene er utformet for å ha et mye større operasjonsområde enn resten av Secure Enclave. Hvis overvåkingen oppdager et ulovlig operasjonspunkt, stopper klokkene i Secure Enclave automatisk og starter ikke på nytt før neste SoC-nullstilling.

Oppsummering av Secure Enclave-funksjonen

Merk: A12-, A13-, S4- og S5-produkter lansert for første gang høsten 2020 har en komponent for sikker lagring (andre generasjon), mens eldre produkter basert på disse SoC-ene har komponenter for sikker lagring (første generasjon).

SoC	Minnebeskyttelsesmotor	Sikker lagring	AES-motor	PKA
A8	Kryptering og autentisering	EEPROM	Ja	Nei
A9	Kryptering og autentisering	EEPROM	DPA-beskyttelse	Ja
A10	Kryptering og autentisering	EEPROM	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
A11	Kryptering, autentisering og repetisjonsforhindring	EEPROM	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
A12 (Apple-enheter lansert før høsten 2020)	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 1	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
A12 (Apple-enheter lansert etter høsten 2020)	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 2	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
A13 (Apple-enheter lansert før høsten 2020)	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 1	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet og oppstartsovervåking
A13 (Apple-enheter lansert etter høsten 2020)	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 2	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet og oppstartsovervåking
A14	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 2	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet og oppstartsovervåking
S3	Kryptering og autentisering	EEPROM	DPA-beskyttelse og låsbare seed-biter	Ja
S4	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 1	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet

SoC	Minnebeskyttelsesmotor	Sikker lagring	AES-motor	PKA
S5 (Apple-enheter lansert før høsten 2020)	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 1	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
S5 (Apple-enheter lansert etter høsten 2020)	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 2	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
S6	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 2	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
T2	Kryptering og autentisering	EEPROM	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet
M1	Kryptering, autentisering og repetisjonsforhindring	Komponent for sikker lagring gen. 2	DPA-beskyttelse og låsbare seed-biter	Nøkler bundet av operativsystemet og oppstartsovervåking

Touch ID og Face ID

Touch ID- og Face ID-sikkerhet

Koder og passord er avgjørende for sikkerheten i Apple-enheter. Samtidig trenger brukerne enkel tilgang til enhetene sine, gjerne mer enn hundre ganger hver dag. Biometrisk autentisering gir en mulighet til å beholde sikkerheten med sterke passord – eller til og med å styrke koden eller passordet, siden det ikke trenger å angis manuelt – mens det gjør det raskt og enkelt å låse opp med et fingertrykk eller et blikk. Touch ID og Face ID erstatter ikke en kode eller passord, men de gir tilgang raskere og enklere i de fleste situasjoner.

Apples arkitektur for biometrisk sikkerhet er basert på streng adskillelse mellom ansvaret til den biometriske sensoren og til Secure Enclave, samt en sikker forbindelse mellom dem. Sensoren fanger det biometriske bildet og overfører det til Secure Enclave på en sikker måte. Under registreringen behandler, krypterer og arkiverer Secure Enclave tilsvarende maldata for Touch ID og Face ID. Under gjenkjenning sammenligner Secure Enclave data fra den biometriske sensoren med de lagrede malene for å avgjøre om enheten skal låses opp, eller for å svare at gjenkjenningen er gyldig (for Apple Pay, bruk i apper og andre bruksområder for Touch ID og Face ID). Arkitekturen støtter enheter som har både sensoren og Secure Enclave (for eksempel iPhone, iPad og mange Mac-systemer), og den gjør det også mulig å fysisk skille sensoren ut i en ekstern enhet som er sikkert sammenkoblet med Secure Enclave i en Mac med Apple Silicon.

Touch ID

Touch ID er fingeravtrykkssystemet som gir sikker tilgang til støttede Apple-enheter på en raskere og enklere måte. Denne teknologien kan lese fingeravtrykksdata fra alle vinkler, og den lærer mer om brukerens fingeravtrykk over tid. Sensoren fortsetter å utvide kartet over fingeravtrykk etter hvert som det registreres flere overlappende noder hver gang den brukes.

Apple-enheter med en Touch ID-sensor kan låses opp ved hjelp av et fingeravtrykk. Touch ID erstatter ikke behovet for en enhetskode eller et brukerpassord, som fortsatt

kreves når enheten starter opp, starter på nytt eller hvis brukeren logger av (på en Mac). I noen apper kan Touch ID også brukes i stedet for en enhetskode eller et brukerpassord, for eksempel for å låse opp passordbeskyttede notater i Notater-appen, for å låse opp nøkkelringbeskyttede nettsteder eller for å låse opp støttede app-passord. I enkelte situasjoner kreves det imidlertid alltid en enhetskode eller et brukerpassord (for eksempel ved endring av eksisterende enhetskode eller brukerpassord, fjerning av eksisterende fingeravtrykksregistreringer og oppretting av nye).

Når fingeravtrykkssensoren gjenkjenner en fingerberøring, utløser dette det avanserte bildesystemet som skanner fingeren og sender bildet til Secure Enclave. Kanalen som brukes til å sikre denne tilkoblingen, varierer avhengig av om Touch ID-sensoren er innebygd i enheten med Secure Enclave eller sitter i en ekstern enhet.

Mens fingeravtrykkbildet vektoriseres for analyse, lagres rasterbildet midlertidig i kryptert minne inne i Secure Enclave for deretter å bli forkastet. Analysen bruker en prosess der mønsterlinjer i underhuden tegnes opp. I denne prosessen går ørsmå biter med informasjon tapt, informasjon som hadde vært nødvendig om brukerens fingeravtrykk skulle rekonstrueres. Under registreringen arkiveres nodekartet i et kryptert format som bare kan leses av Secure Enclave, som en mal som sammenlignes ved fremtidige gjenkjenninger, men som ikke inneholder identifiserende informasjon. Disse dataene forlater aldri enheten. De sendes ikke til Apple, og inkluderes heller ikke i sikkerhetskopier av enheten.

Innebygd kanalsikkerhet for Touch ID

Kommunikasjonen mellom Secure Enclave og den innebygde Touch ID-sensoren foregår via en databuss for serie-eksternt grensesnitt. Prosessoren videresender dataene til Secure Enclave, men kan ikke lese dem. Den er kryptert og autentisert med en øktnøkkel som formidles ved hjelp av en delt nøkkel som opprettes for hver Touch ID-sensor og dens tilhørende Secure Enclave ved fabrikken. Den delte nøkkelen er sterk, tilfeldig og forskjellig for hver Touch ID-sensor. Utvekslingen av øktnøkler bruker AES-nøkkelinnpakning der begge sider oppgir en tilfeldig nøkkel som danner øktnøkkelen og bruker transportkryptering som gir både autentisering og konfidensialitet (ved hjelp av AES-CCM).

Face ID-sikkerhet

Med et enkelt blikk låser Face ID opp støttede Apple-enheter på en sikker måte. Denne gir intuitiv og sikker autentisering ved hjelp av TrueDepth-kamerasystemet, som bruker avansert teknologi til å kartlegge geometrien til en brukers ansikt. Face ID bruker nevrale nettverk for å fastslå oppmerksomhet, treff og antiforfalskning, slik at brukeren kan låse opp telefonen med et blikk. Face ID tilpasser seg automatisk til endringer i utseendet og gir en grundig beskyttelse av personvernet og sikkerheten til brukerens biometriske data nøye.

Face ID er laget for å bekrefte oppmerksomhet fra brukeren, tilby robust autentisering med lav feilgjenkjenningsfrekvens og redusere både digital og fysisk forfalskning.

TrueDepth-kameraet ser automatisk etter brukerens ansikt når brukeren vekker en Apple-enhet som har Face ID (ved å løfte enheten opp eller trykke på skjermen), i tillegg til når enhetene forsøker å autentisere deg for å vise en innkommende varsling eller når en app som støtter det, ber om Face ID-autentisering. Når et ansikt gjenkjennes, bekrefter Face ID oppmerksomhet og intensjon om å låse opp ved å gjenkjenne at øynene til brukeren er åpne og oppmerksomheten er rettet mot enheten. Face ID-oppmerksomhetskontrollen er deaktivert når VoiceOver er aktivert for å lette tilgjengeligheten, og kan hvis nødvendig deaktiveres separat.

Etter at TrueDepth-kameraet har bekreftet tilstedeværelsen av et oppmerksomt ansikt, projiserer og leser det mer enn 30 000 infrarøde prikker for å lage et dybdekart av ansiktet, sammen med et infrarødt 2D-bilde. Disse dataene brukes til å skape en sekvens med 2D-bilder og dybdekart, som signeres digitalt og sendes til Secure Enclave. For å motvirke både digitale og fysiske forfalskninger, genererer TrueDepth-kameraet en tilfeldig sekvens av 2D-bilder og dybdekartopptak og utarbeider et enhetsspesifikt tilfeldig mønster. En del av Secure Neural Engine, som ligger beskyttet i Secure Enclave, omdanner disse dataene til en matematisk fremstilling og sammenligner den fremstillingen med de registrerte ansiktsdataene. Disse registrerte ansiktsdataene er selv en matematisk fremstilling av brukerens ansikt fanget opp gjennom en serie poseringer.

Magic Keyboard med Touch ID

Magic Keyboard med Touch ID (og Magic Keyboard med Touch ID og talltastatur) gjør det mulig å bruke Touch ID-sensoren i et eksternt tastatur sammen med Mac med Apple Silicon. Magic Keyboard med Touch ID fungerer som biometrisk sensor. Enheten lagrer ikke biometriske maler, utfører ikke biometrisk identifisering og håndhever ikke sikkerhetsregler (for eksempel å måtte oppgi passordet etter 48 timer uten opplåsing). Touch ID-sensoren i Magic Keyboard med Touch ID må være sikkert sammenkoblet med Secure Enclave i Mac før den kan brukes, og deretter gjennomfører Secure Enclave registreringen, gjenkjenning og håndhevelse av sikkerhetsregler på samme måte som for en innebygd Touch ID-sensor. Apple gjennomfører sammenkoblingsprosessen for Magic Keyboard med Touch ID som leveres med Mac, på fabrikken. Brukeren kan også gjennomføre sammenkoblingen ved behov. Et Magic Keyboard med Touch ID kan være sikkert sammenkoblet med én Mac om gangen, men én Mac kan sammenkobles med opptil fem forskjellige Magic Keyboard med Touch ID-tastatur.

Magic Keyboard med Touch ID og innebygde Touch ID-sensorer er kompatible. Hvis en finger som ble registrert på en innebygd Mac Touch ID-sensor brukes på et Magic Keyboard med Touch ID, behandler Secure Enclave i Macen gjenkjenningen – og vice versa.

For å støtte sikker sammenkobling og kommunikasjon mellom Mac Secure Enclave og Magic Keyboard med Touch ID er tastaturet utstyrt med en Public Key Accelerator (PKA) maskinvareblokk for attestering, og med maskinvarebaserte nøkler for å gjennomføre de nødvendige kryptografiske prosessene.

Sikker sammenkobling

Før et Magic Keyboard med Touch ID kan brukes til Touch ID-operasjoner, må det være sikkert sammenkoblet med Macen. Ved sammenkoblingen utveksler Secure Enclave i Macen og PKA-blokken i Magic Keyboard med Touch ID offentlige nøkler, rotfestet i godkjente Apple-sertifikater, og de bruker maskinvarebaserte attesteringsnøkler og kortvarig ECDH for å attestere identiteten. På Macen beskyttes disse dataene av Secure Enclave; og på Magic Keyboard med Touch ID beskyttes de av PKA-blokken. Etter sikker sammenkobling vil all kommunikasjon mellom Macen og Magic Keyboard med Touch ID krypteres av AES-GCM, med kortvarige ECDH-nøkler basert på de lagrede identitetene.

Sikker intensjon for sammenkobling

For å gjennomføre enkelte Touch ID-operasjoner for første gang, som å registrere et nytt fingeravtrykk, må brukeren fysisk bekrefte intensjonen om å bruke et Magic Keyboard med Touch ID med Macen. Fysisk intensjon bekreftes ved å trykke to ganger på av/på-knappen

på Macen når brukergrensesnittet ber om det, eller ved å bruke et fingeravtrykk som allerede er registrert på Macen. Du finner mer informasjon om dette under Sikker intensjon og tilkoblinger til Secure Enclave.

Apple Pay-transaksjoner kan autoriseres med en Touch ID-gjenkjenning eller ved å oppgi passordet for macOS-brukeren og trykke to ganger på Touch ID-knappen på Magic Keyboard med Touch ID. Sistnevnte lar også brukeren bekrefte fysisk intensjon uten Touch ID-gjenkjenning.

Kanalsikkerhet for Magic Keyboard med Touch ID

For å opprette en sikker kommunikasjonskanal mellom Touch ID-sensoren i Magic Keyboard med Touch ID og Secure Enclave på den sammenkoblede Macen, må følgende oppfylles:

- Sikker sammenkobling mellom PKA-blokken for Magic Keyboard med Touch ID og Secure Enclave, som beskrevet tidligere
- En sikker kanal mellom Magic Keyboard med Touch ID-sensor og PKA-blokken

Den sikre kanalen mellom Magic Keyboard med Touch ID-sensor og PKA-blokken opprettes ved fabrikken ved hjelp av en unik nøkkel som deles av de to. (Dette er den samme teknikken som brukes til å opprette den sikre kanalen mellom Secure Enclave på Macen og den innebygde sensoren for Mac-maskiner med innebygd Touch ID.)

Touch ID, Face ID, koder og passord

For å kunne bruke Touch ID eller Face ID må brukeren konfigurere enheten slik at det kreves en kode eller et passord for å låse den opp. Når Touch ID eller Face ID gjenkjenner et registrert fingeravtrykk eller et ansikt, låses enheten opp uten forespørsel om enhetskode eller passord. Dette gjør det enklere å bruke lengre og mer komplekse koder eller passord, fordi brukeren ikke trenger å oppgi dem like ofte. Touch ID og Face ID erstatter ikke brukerens kode eller passord, men de gir i stedet enkel tilgang til enheten innenfor gjennomtenkte grenser og tidsbegrensinger. Dette er viktig fordi en sterk kode eller et sterkt passord er grunnlaget for hvordan en brukers iPhone-, iPad-, Mac- eller Apple Watch-enhet beskytter brukerens data kryptografisk.

Når en enhetskode eller et passord kreves

Brukerne kan bruke koden eller passordet når som helst i stedet for Touch ID eller Face ID, men det finnes situasjoner der biometri ikke tillates. Følgende sikkerhetssensitive handlinger krever alltid at det oppgis en kode eller et passord:

- oppdatering av programvaren
- sletting av enheten
- visning eller endring av kodeinnstillinger
- installering av konfigurasjonsprofiler
- opplåsing av Sikkerhet og personvern-panelet i Systemvalg på Mac
- opplåsing av Brukere og grupper-panelet i Systemvalg på Mac (hvis FileVault er slått på)

En kode eller et passord kreves også hvis enheten befinner seg i en av følgende tilstander:

- enheten har nettopp blitt slått på eller startet på nytt
- brukeren har logget av Mac-kontoen (eller har ikke logget på ennå)

- brukeren har ikke låst opp enheten i løpet av en periode på mer enn 48 timer
- brukeren har ikke brukt koden eller passordet til å låse opp enheten i løpet av en periode på 156 timer (seks og en halv dag), og brukeren har ikke brukt biometri til å låse opp enheten i løpet av en periode på 4 timer
- enheten har mottatt en fjernlåsingskommando
- brukeren gikk ut av slå av / Nødanrop (SOS) ved å trykke og holde på en av volumknappene og dvale/vekke-knappen samtidig i 2 sekunder og trykket deretter på Avbryt
- det var fem mislykkede biometriforsøk (men for brukervennlighetens skyld kan enheten tilby kode eller passord i stedet for biometri etter et mindre antall mislykkede forsøk)

Når Touch ID eller Face ID er aktivert på en iPhone eller iPad, låses enheten umiddelbart når dvale/vekke-knappen trykkes inn, og enheten låses hver gang den går i dvale. Touch ID og Face ID krever gjenkjenning, eller alternativt koden, ved hver enkelt vekking.

Sannsynligheten for at en tilfeldig person i befolkningen kan låse opp en brukers iPhone, iPad eller Mac er 1 til 50 000 med Touch ID eller 1 til 1 000 000 med Face ID. Denne sannsynligheten øker med flere registrerte fingeravtrykk (opptil 1 til 10 000 med fem fingeravtrykk) eller utseender (opptil 1 til 500 000 med to utseender). For ytterligere beskyttelse tillater både Touch ID og Face ID kun fem mislykkede gjenkjenningsforsøk før koden eller passordet kreves for å få tilgang til brukerens enhet eller konto. Med Face ID er sannsynligheten for en falsk gjenkjenning forskjellig for tvillinger og søsken som ligner på brukeren, og blant barn under 13 år (ettersom ansiktstrekkene deres kanskje ikke er fullt utviklet). Hvis en bruker er bekymret for en falsk gjenkjenning, anbefaler Apple at det brukes en kode til autentisering.

Ansiktsgjenkjenning-sikkerhet

Ansiktsgjenkjenningen utføres inne i Secure Enclave ved hjelp av nevrale nettverk som er trent spesifikt for dette formålet. Apple utviklet de nevrale nettverkene for ansiktsgjenkjenning ved hjelp av mer enn én milliard bilder, inkludert infrarøde bilder (IR) og dybdebilder som ble samlet inn i studier med deltakernes informerte samtykke. Apple jobbet deretter med deltakere over hele verden for å inkludere en representativ gruppe med personer, og tok hensyn til kjønn, alder, etnisitet og andre faktorer. Studiene ble utvidet etter behov for å gi en stor grad av nøyaktighet for et bredt utvalg brukere. Face ID er laget for å fungere med hatter, skjerf, briller, kontaktlinser og mange typer solbriller. Det er videre laget for å kunne fungere innendørs, utendørs og selv i stummende mørke. Et ytterligere nevralt nettverk som er opptrent i å gjenkjenne og motstå forfalskning, beskytter mot forsøk på å låse opp enheten med bilder eller masker. Face ID-data, inkludert matematiske fremstillinger av en brukers ansikt, krypteres og er kun tilgjengelig for Secure Enclave. Disse dataene forlater aldri enheten. De sendes ikke til Apple, og inkluderes heller ikke i sikkerhetskopier av enheten. Følgende Face ID-data arkiveres og krypteres kun for bruk av Secure Enclave ved normal drift:

- De matematiske fremstillingene av en brukers ansikt under registreringen.
- De matematiske fremstillingene av en brukers ansikt som beregnes under enkelte opplåsingsforsøk hvis Face ID vurderer dem som nyttige for å forbedre framtidig gjenkjenning.

Ansiktsbilder som registreres under normal bruk lagres ikke, men forkastes i stedet umiddelbart etter at den matematiske fremstillingen beregnes for enten registrering eller sammenligning med de registrerte Face ID-dataene.

Forbedre Face ID-treff

For å forbedre treffprosenten og holde tritt med de naturlige endringene i et ansikt og utseende, utvider Face ID den lagrede matematiske fremstillingen over tid. Ved vellykket treff kan Face ID bruke den nylig beregnede matematiske framstillingen, forutsatt at kvaliteten er tilfredsstillende, for et begrenset antall ytterligere treff før de dataene forkastes. I motsatt fall, hvis Face ID ikke lykkes i å gjenkjenne et ansikt, men gjenkjenningskvaliteten er over en bestemt terskel og en bruker umiddelbart følger opp med å angi koden, tar Face ID et nytt bilde og utvider de registrerte Face ID-dataene med den nyberegnede matematiske framstillingen. Disse nye Face ID-dataene forkastes hvis en bruker slutter å gjenkjennes mot dem eller etter et begrenset antall treff. Disse utvidelsesprosessene gjør det mulig for Face ID å holde tritt med vesentlige endringer i en brukers ansiktshår eller sminkebruk samtidig som falske positiver minimeres.

Bruksområder for Touch ID og Face ID

Låse opp en enhet eller en brukerkonto

Når en enhet eller konto låses, og Touch ID eller Face ID er deaktivert, forkastes nøklene for den høyeste klassen av databeskyttelse, som oppbevares i Secure Enclave. Filene og nøkkelringobjektene i denne klassen er ikke tilgjengelige før brukeren låser opp enheten eller kontoen ved å oppgi koden eller passordet.

Hvis Touch ID eller Face ID er slått på, kastes ikke nøklene når enheten eller kontoen låses. De blir i stedet pakket med en nøkkel som gis til Touch ID- eller Face ID-undersystemet i Secure Enclave. Når en bruker forsøker å låse opp enheten eller kontoen og enheten gjenkjenner brukeren, oppgir den nøkkelen for å pakke ut databeskyttelsesnøklene, og enheten eller kontoen låses opp. Denne prosessen sørger for ytterligere beskyttelse ved å kreve samarbeid mellom undersystemene for databeskyttelse og Touch ID eller Face ID for at enheten skal låses opp.

Når enheten starter på nytt, går nøklene som kreves for at Touch ID eller Face ID skal låse opp enheten, tapt. De forkastes av Secure Enclave når en hvilken som helst betingelse som krever at koden eller passordet oppgis, blir oppfylt.

Sikring av kjøp med Apple Pay

Brukeren kan også bruke Touch ID og Face ID med Apple Pay for å gjennomføre enkle og sikre kjøp i butikker, apper og på nettet:

- *Med Touch ID:* For Touch ID bekreftes intensjonen om å betale ved å bruke handlingen av å aktivere Touch ID-sensoren kombinert med vellykket gjenkjenning av brukerens fingeravtrykk.
- *Med Face ID i butikker:* Hvis du vil godkjenne en betaling i butikk med Face ID, må brukeren først bekrefte intensjon om å betale ved å dobbelttrykke på sideknappen. Dette dobbelttrykket bekrefter brukerintensjonen ved å bruke en fysisk handling som er koblet direkte til Secure Enclave og er motstandsdyktig mot forfalskning som følge av en ondsinnet prosess. Deretter autentiserer brukeren med Face ID før enheten holdes opp til den kontaktløse leseren. En annen Apple Pay-betalingsmetode kan velges etter Face ID-autentisering. Dette krever ny autentisering, men brukeren trenger ikke å dobbelttrykke på sideknappen igjen.
- *Med Face ID i apper og på internett:* Hvis brukeren vil utføre betalinger i apper og på nettet, bekrefter brukeren intensjonen om å betale ved å dobbelttrykke på sideknappen og deretter autentisere med Face ID for å godkjenne betalingen. Hvis Apple Pay-transaksjonen ikke fullføres innen 60 sekunder etter at du har dobbelttrykket på sideknappen, må brukeren bekrefte intensjonen om å betale ved å dobbelttrykke på nytt.

Bruke API-er levert av systemet

Tredjepartsapper kan bruke systemets API-er til å be om at brukeren autentiserer med Touch ID eller Face ID eller en kode, og apper som støtter Touch ID, støtter automatisk Face ID uten endringer. Når Touch ID eller Face ID brukes, får appen kun beskjed om autentiseringen lyktes eller ikke. Den får ikke tilgang til Touch ID eller Face ID eller dataene som er tilknyttet den registrerte brukeren.

Beskytte nøkkelringobjekter

Nøkkelringobjekter kan også beskyttes med Touch ID eller Face ID. Da kreves det et gjenkjent fingeravtrykk eller ansikt eller at koden på enheten eller kontopassordet oppgis for at de skal bli frigitt av Secure Enclave. Apputviklere har API-er for å verifisere at en kode eller et passord har blitt angitt av brukeren, før Touch ID eller Face ID eller en kode eller et passord kreves for å låse opp nøkkelringobjekter. Apputviklere kan gjøre følgende:

- Kreve at API-autentiseringsoperasjoner ikke faller tilbake til et app-passord eller koden på enheten. De kan spørre om en bruker er registrert, noe som gjør det mulig å bruke Touch ID og Face ID som en sekundær faktor i sikkerhetssensitive apper.
- Generere og bruke ECC-nøkler (elliptisk kurve-kryptografi) inne i Secure Enclave som kan beskyttes av Touch ID eller Face ID. Operasjoner med disse nøklene utføres alltid inne i Secure Enclave etter at Secure Enclave har godkjent bruken.

Foreta og godkjenne kjøp

Brukere kan også konfigurere Touch ID og Face ID til å godkjenne kjøp fra iTunes Store, App Store, Apple Books og annet, slik at brukere slipper å oppgi Apple-ID-passordet. Ved kjøp verifiserer Secure Enclave at en biometrisk autentisering ble gjennomført, og deretter frigjør den ECC-nøkler som brukes til å signere forespørselen fra butikken.

Sikker intensjon og tilkoblinger til Secure Enclave

Sikker intensjon er en måte å bekrefte brukerens intensjon på uten interaksjon med operativsystemet eller applikasjonsprosessoren. Tilkoblingen er en fysisk kobling – fra en fysisk knapp til Secure Enclave – som er tilgjengelig i følgende:

- iPhone X eller nyere
- Apple Watch Series 1 eller nyere
- iPad Pro (alle modeller)
- iPad Air (2020)
- Mac-maskiner med Apple-chiper

Med denne koblingen kan brukere bekrefte intensjonen om å gjennomføre en operasjon på en måte som ikke kan forfalskes selv av programvare som kjøres med rotrettigheter eller i kjernen.

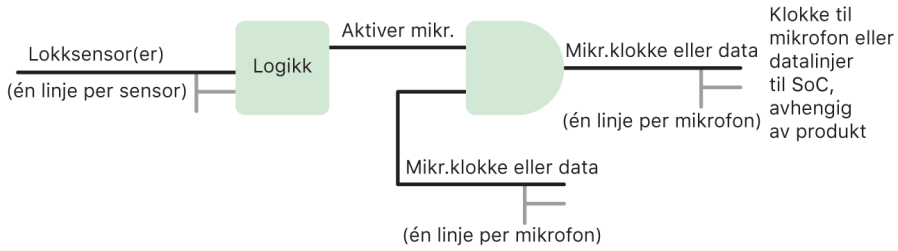
Denne funksjonen brukes til å bekrefte brukerintensjon under Apple Pay-transaksjoner og ved fullføring av sammenkoblingen av Magic Keyboard med Touch ID og en Mac med Apple Silicon. Et dobbelttrykk på riktig tast når brukergrensesnittet ber om det, er en bekreftelse av brukerens intensjon. Du finner mer informasjon om dette under Sikring av kjøp med Apple Pay. En lignende mekanisme – basert på Secure Enclave og T2-firmware – støttes på MacBook-modeller med Apple T2-sikkerhetsbrikke, uten Touch Bar.

Maskinvarefrakobling av mikrofoner

Alle Apple Silicon-baserte bærbare Macer og Intel-baserte bærbare Macer med Apple T2-sikkerhetsbrikken har en maskinvarefrakobling som deaktiverer mikrofonen når maskinen lukkes. På alle 13-tommers MacBook Pro- og MacBook Air-maskiner med T2-brikken, alle bærbare MacBook-maskiner med en T2-brikke fra 2019 eller nyere og alle bærbare MacBook-maskiner med Apple Silicon er denne frakoblingen kun implementert i maskinvaren. Frakoblingen er utformet for å hindre at all programvare, selv med rot- eller kjernerettigheter i macOS og selv programvaren på T2-brikken eller annen firmware, bruker mikrofonen når maskinen er lukket. (Kameraet er ikke frakoblet i maskinvare fordi synsfeltet er fullstendig blokkert når maskinen er lukket.)

iPad-modeller fra og med 2020 har også maskinvarefrakobling av mikrofonen. Når et MFi-kompatibelt deksel (inkludert de som selges av Apple) kobles til iPaden og lukkes, kobles mikrofonen fra i maskinvaren. Dette er utviklet for å hindre at mikrofonlyddata gjøres tilgjengelig for hvilken som helst programvare – selv programvare med rot- eller kjernerettigheter i iPadOS, eller hvilken som helst enhetsfirmware.

Beskyttelsene i denne delen implementeres direkte med maskinvarelogikk, i henhold til følgende kretsdiagram:



Kretsdiagram.

I hvert produkt med frakobling av maskinvaremikrofon registrerer én eller flere lokksensorer at lokket eller dekselet fysisk lukkes ved bruk av en fysisk egenskap (for eksempel en Hall-effektsensor eller en hengselvinkelsensor) for interaksjonen. For sensorer som trenger kalibrering, angis det parametere under produksjon av enheten, og kalibreringsprosessen inkluderer en ikke-reversibel låsing av maskinvaren som forhindrer ytterligere endringer av sensitive parametere på sensoren. Disse sensorene avgir et direkte maskinvaresignal som går gjennom et enkelt sett med maskinvarelogikk som ikke kan omprogrammeres. Denne logikken avpreller, håndterer hysterese og/eller forsinker signalet med opptil 500 ms før mikrofonen deaktiveres. Avhengig av produktet, kan dette signalet implementeres ved enten å deaktivere linjene som overfører data mellom mikrofonen og System on Chip (SoC) eller ved å deaktivere en av inndatalinjene til mikrofonmodulen som tillater at den er aktiv, for eksempel klokkelinjen eller en lignende effektiv kontroll.

Ekspresskort med reservestrøm

Hvis iOS ikke kjører fordi iPhone må lades, kan det fortsatt være nok strøm i batteriet til å støtte transaksjoner med ekspressreisekort. Støttede iPhone-enheter støtter denne funksjonen automatisk med:

- Et betalingskort eller reisekort som er angitt som ekspressreisekort
- Studentbevis med Ekspressmodus slått på
- Adgangskort til resorter med Ekspressmodus slått på
- Bilnøkler med Ekspressmodus slått på

Når du trykker på sideknappen (eller på Hjem-knappen på iPhone SE (andre generasjon)), vises symbolet for lite batteri, i tillegg til tekst som viser at Ekspressreisekort er tilgjengelige for bruk. NFC-kontrolleren utfører ekspressreisekorttransaksjoner under de samme forholdene som når iOS kjører, med unntak av at transaksjoner kun indikeres med et følbart varsel (det vises ikke en synlig varsling). På iPhone SE (andre generasjon) kan det ta noen sekunder før fullførte transaksjoner vises på skjermen. Denne funksjonen er ikke tilgjengelig når en standard brukerutløst nedstengning utføres.

Systemsikkerhet

Oversikt over systemsikkerhet

Systemsikkerheten er ansvarlig for å kontrollere tilgangen til operativsystemene på Apple-enheter uten at det går på bekostning av brukervennlighet og baserer seg på de unike egenskapene som finnes i Apple-maskinvaren. Systemsikkerhet omfatter oppstartsprosessen, programvareoppdateringer og beskyttelse av datamaskinsystemressurser som prosessoren, minnet, disken, programvareprogrammer og lagrede data.

De nyeste versjonene av Apple-operativsystemene er de sikreste. En viktig del av Apple-sikkerheten er *sikker oppstart*, som beskytter systemet fra infisering av skadelig programvare ved oppstart. Sikker oppstart starter i maskinvare og bygger en godkjenningskjede gjennom programvare, der hvert trinn er utviklet for å sikre at det neste fungerer riktig før kontrollen overlates. Denne sikkerhetsmodellen støtter ikke kun standardoppstarten til Apple-enheter, men også de forskjellige modusene for gjenoppretting og rettidige oppdateringer på Apple-enheter. Underkomponenter som T2-brikken og Secure Enclave utfører også sin egen sikre oppstart for å sørge for at de kun starter godkjent kode fra Apple. Oppdateringssystemet kan til og med bidra til å forhindre nedgraderingsangrep, slik at enheter ikke kan rulles tilbake til en eldre versjon av operativsystemet (som en angriper vet hvordan skal angripes) som en metode for å stjele brukerdata.

Apple-enheter har også oppstarts- og kjøringsbeskyttelse, slik at de opprettholder integriteten under kontinuerlig drift. Apple-designede komponenter i iPhone, iPad, Apple Watch, Apple TV, HomePod og Mac med Apple Silicon har en felles arkitektur for beskyttelse av integriteten til operativsystemet. macOS har også et utvidet og konfigurerbart sett med beskyttelsesfunksjoner for å støtte den differensierte databehandlingsmodellen sin, samt funksjoner som støttes på alle Mac-maskinvareplattformer.

Sikker oppstart

Oppstartsprosess for iOS- og iPadOS-enheter

De enkelte trinnene i oppstartsprosessen inneholder komponenter som Apple har signert kryptografisk for å sikre integriteten, slik at oppstarten bare fortsetter etter at godkjenningskjeden er bekreftet. Disse komponentene består blant annet av bootloaderne, kjernen, kjerneutvidelser og basebåndfirmware for mobilnett. Den sikre oppstartssekvensen er utviklet for å bekrefte at det ikke er mulig å manipulere programvare på de laveste nivåene.

Når en iOS- eller iPadOS-enhet slås på, kjører applikasjonsprosessoren umiddelbart kode fra skrivebeskyttet minne, som kalles oppstart-ROM. Denne koden kan ikke endres og kalles for *maskinvarens «root of trust»*. Den implementeres når brikken produseres, og godkjenning er underforstått. Oppstart-ROM-koden inneholder den offentlige nøkkelen for Apple-rotsertifikatautoriteter (CA), som brukes til å bekrefte at iBoot-bootloaderen er signert av Apple før lasting. Dette er første trinn i godkjenningskjeden der hvert trinn sørger for at det neste er signert av Apple. Når iBoot har utført oppgavene sine, verifiserer og kjører den iOS- eller iPadOS-kjernen. For enheter som har en A9-prosessor eller en eldre prosessor i A-serien, blir en ekstra LLB-fase (Low-Level Bootloader) lastet inn og bekreftet av oppstart-ROM-en, som så laster inn og bekrefter iBoot.

Feil i innlasting eller verifisering av følgende trinn håndteres ulikt avhengig av maskinvaren:

- *Oppstart-ROM kan ikke laste inn LLB (eldre enheter):* DFU-modus (Device Firmware Upgrade)
- *LLB eller iBoot:* Gjenopprettingsmodus

I begge tilfeller må enheten være koblet til Finder (macOS 10.15 eller nyere) eller iTunes (i macOS 10.14 eller eldre) via USB og gjenopprettet med fabrikkinnstillingene.

Boot Progress Register (BPR) brukes av Secure Enclave til å begrense tilgang til brukerdata i forskjellige moduser og oppdateres før følgende moduser:

- *DFU-modus:* Angis av oppstart-ROM på enheter med A12 eller nyere SoC-er
- *Gjenopprettingsmodus:* Angis av iBoot på enheter med Apple A10, S2 og nyere SoC-er

På enheter med tilgang til mobilnettverk utfører et basebåndundersystem for mobilnett ekstra sikker oppstart ved hjelp av signert programvare og nøkler som er bekreftet av basebåndprosessoren.

Secure Enclave utfører også en sikker oppstart som sørger for at programvaren (sepOS) er bekreftet og signert av Apple.

Minnesikker iBoot-implementering

I iOS 14 og iPadOS 14 endret Apple C-kompilatorverktøykjeden som brukes til å bygge iBoot-bootloaderen for å forbedre sikkerheten. Den endrede verktøykjeden implementerer kode som er utviklet for å forhindre minne- og typesikkerhetsproblemer som vanligvis forekommer i C-programmer. Den bidrar for eksempel til å forhindre de fleste sårbarheter i følgende klasser:

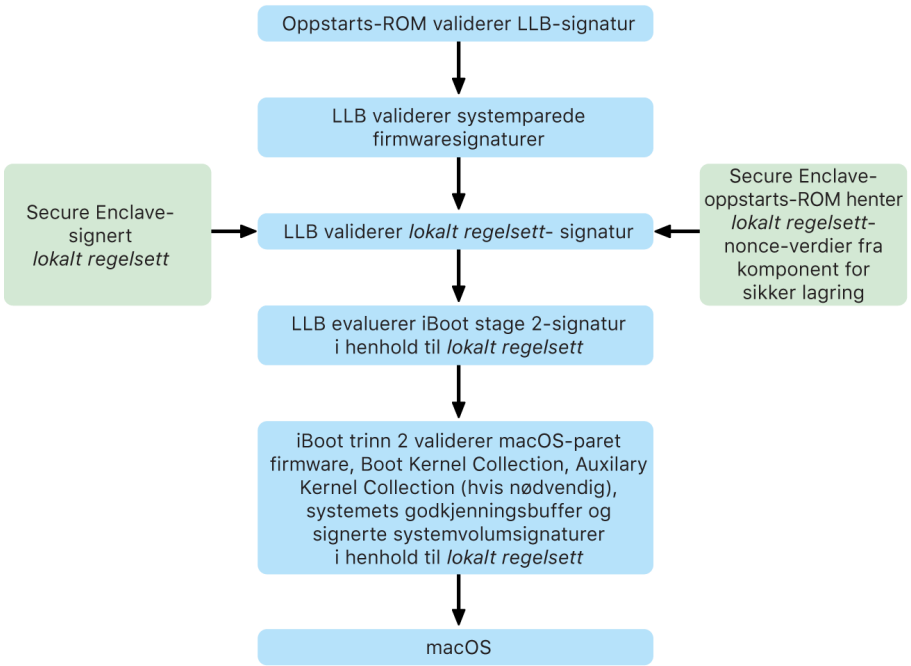
- Bufferoverskridelse, ved å sørge for at alle pekere inneholder områdeinformasjon som verifiseres ved tilgang til minnet
- Utnyttelse av svakheter i håndtering av «heap»-minne, ved å separere «heap»-data fra tilhørende metadata og nøyaktig oppdage feiltilstander som «double-free»
- Typeforvirring, ved å sørge for at alle pekere inneholder typeinformasjon om kjøring som verifiseres under pekerkonverteringsoperasjoner
- Typeforvirring forårsaket av «use-after-free»-feil, ved å skille alle dynamiske minnetilordninger etter statisk type

Denne teknologien er tilgjengelig på iPhone med Apple A13 Bionic eller nyere, og på iPad med A14 Bionic-brikken.

Mac-maskiner med Apple-chiper

Oppstartsprosess for Macer med Apple Silicon

Når en Mac med Apple Silicon slås på, utfører den en oppstartsprosess som er svært lik oppstartsprosessen på iPhone og iPad.



Trinnene i oppstartsprosessen når en Mac med Apple Silicon startes.

Brikken kjører kode fra oppstart-ROM i det første trinnet i godkjenningskjeden. Sikker macOS-oppstart på Macer med Apple Silicon verifiserer ikke bare selve operativsystemkoden, men også sikkerhetsregelsettene og til og med kjerneutvidelser (støttet, men ikke anbefalt) konfigurert av autoriserte brukere.

Når LLB startes, verifiserer den signaturene og laster systemparet firmware for intra-SoC-kjerner som lagrings-, skjerm- og systemadministreringskontrollerne samt Thunderbolt-kontrollere. LLB er også ansvarlig for å laste LocalPolicy, som er en fil signert av Secure Enclave-prosessoren. LocalPolicy-filen beskriver konfigurasjonen som brukeren har valgt for sikkerhetsregelsettene for systemoppstart og kjøring. LocalPolicy har samme datastrukturformat som alle andre oppstartsobjekter, men det signeres lokalt av en privat nøkkel som kun er tilgjengelig i en bestemt datamaskins Secure Enclave, i stedet for å bli signert av en sentral Apple-tjener (som programvareoppdateringer).

For å bidra til å hindre repetisjon av tidligere LocalPolicy må LLB slå opp en nonce-verdi fra den Secure Enclave-tilknyttede komponenten for sikker lagring. For å gjøre dette bruker den oppstart-ROM for Secure Enclave og sørger for at nonce-verdien i LocalPolicy samsvarer med nonce-verdien i komponenten for sikker lagring. Dette bidrar til å forhindre at en gammel LocalPolicy, som kan ha vært konfigurert for lavere sikkerhet, brukes på nytt i systemet etter at sikkerheten er oppgradert. Resultatet er at sikker oppstart på Macer med Apple Silicon hjelper til med å beskytte mot ikke bare tilbakerulling av operativsystemversjoner, men også mot nedgraderinger av sikkerhetsregelsett.

LocalPolicy fanger opp om operativsystemet er konfigurert for Full, Redusert eller Middels sikkerhet.

- *Full sikkerhet:* Systemet oppfører seg som iOS og iPadOS og tillater kun oppstartsprogramvare som var kjent for å være den nyeste som var tilgjengelig på installasjonstidspunktet.
- *Redusert sikkerhet:* LLB instrueres om å godkjenne «globale» signaturer som er pakket med operativsystemet. Dette tillater systemet å kjøre eldre versjoner av macOS. Siden eldre versjoner av macOS naturlig nok har ikke-oppdaterte sårbarheter, beskrives denne sikkerhetsmodusen som *Redusert*. Dette er også regelsettnivået som kreves for å støtte oppstart av kjerneutvidelser.
- *Middels sikkerhet:* Systemet oppfører seg som Redusert sikkerhet siden det bruker global signaturverifisering for iBoot og videre, men det forteller også iBoot at det skal akseptere noen oppstartsobjekter som signeres av Secure Enclave, med samme nøkkel som brukes til å signere LocalPolicy. Dette regelsettnivået støtter brukere som bygger, signerer og starter sine egne, tilpassede XNU-kjerner.

Hvis LocalPolicy indikerer til LLB at det valgte operativsystemet kjører i Full sikkerhet, evaluerer LLB den tilpassede signaturen for iBoot. Hvis det kjører i Redusert sikkerhet eller Middels sikkerhet, evaluerer det den globale signaturen. Eventuelle signaturverifiseringsfeil fører til at systemet starter til recoveryOS for å tilby reparasjonsvalg.

Etter at LLB overleverer til iBoot, laster den macOS-paret firmware som de for Secure Neural Engine, Alltid på-prosessor og annen firmware. iBoot ser også på informasjon om LocalPolicy overlevert fra LLB. Hvis LocalPolicy indikerer at det skal finnes en AuxKC (Auxiliary Kernel Collection), ser iBoot etter den på filsystemet, verifiserer at den er signert av Secure Enclave med samme nøkkel som LocalPolicy, og verifiserer at hashen samsvarer med en hash lagret i LocalPolicy. Hvis AuxKC verifiseres, plasserer iBoot den i minnet med Boot Kernel Collection, før den låser hele minneområdet og dekker Boot Kernel Collection og AuxKC med SCIP (System Coprocessor Integrity Protection). Hvis

regelsettet indikerer at det skal finnes en AuxKC, men den ikke blir funnet, fortsetter systemet å starte opp i macOS uten den. iBoot er også ansvarlig for å verifisere rothashen av det signerte systemvolumet (SSV) for å sikre at filsystemet som kjernen skal aktivere, er fullstendig integritetsverifisert.

Oppstartsmoduser for Macer med Apple Silicon

Macer med Apple Silicon har oppstartsmodusene som beskrives nedenfor.

Modus	Tastaturkombinasjon	Beskrivelse
macOS	Trykk på og **slipp** av/på-knappen når maskinen er avslått.	1. Oppstart-ROM overleverer til LLB. 2. LLB laster systemparet firmware og LocalPolicy for valgt macOS. 3. LLB låser inn en indikasjon i Boot Progress Register (BPR) om at den starter opp i vanlig macOS, og overleverer deretter til iBoot. 4. iBoot laster macOS-paret firmware, den statiske godkjenningsbufferen, enhetstreet og Boot Kernel Collection. 5. Hvis LocalPolicy tillater det, laster iBoot Auxiliary Kernel Collection (AuxKC) til kjerneutvidelser fra tredjeparter. 6. Hvis LocalPolicy ikke deaktiverte det, verifiserer iBoot rotsignaturhashen for det signerte systemvolumet (SSV).
recoveryOS	**Hold nede** av/på-knappen når maskinen er avslått.	1. Oppstart-ROM overleverer til LLB. 2. LLB laster systemparet firmware og LocalPolicy for recoveryOS. 3. LLB låser inn en indikasjon i Boot Progress Register om at den starter opp i recoveryOS, og overleverer deretter til i Boot for recoveryOS. 4. iBoot laster inn macOS-paret firmware, godkjennings-bufferen, enhetstreet og Boot Kernel Collection. *Merk:* Sikkerhetsnedgraderinger er ikke tillatt i LocalPolicy for recoveryOS.
Tilbakefalls-recoveryOS	**Trykk to ganger på og hold nede** av/på-knappen når maskinen er avslått.	Samme prosess som recoveryOS-oppstart, bortsett fra at den starter opp en ekstra kopi av recoveryOS som beholdes for robusthet. LLB låser imidlertid ikke en indikasjon i Boot Progress Register som sier at den går inn i recoveryOS, og derfor har ikke tilbakefalls-recoveryOS muligheten til å endre systemets sikkerhetstilstand.
Sikker modus	Start opp til recoveryOS som beskrevet over, og hold nede **Skift**-tasten mens du velger oppstartsvolumet.	1. Starter opp til recoveryOS som beskrevet over. 2. Hvis du holder nede Skift-tasten mens du velger et volum, vil BootPicker-applikasjonen godkjenne dette macOS-et for oppstart, som normalt, men det vil også angi en `nvram`-variabel som ber iBoot om ikke å laste AuxKC ved neste oppstart. 3. Systemet starter på nytt og starter opp til det målrettede volumet, men iBoot laster ikke AuxKC.

Kontroll av sikkerhetsregelsett for Startdisk for Macer med Apple Silicon

Oversikt

I motsetning til sikkerhetsregelsett på Intel-baserte Macer, gjelder sikkerhetsregelsettet på Macer med Apple Silicon for hvert installerte operativsystem. Dette betyr at det støttes flere installerte macOS-forekomster med forskjellige versjoner og sikkerhetsregelsett på samme Mac. Dette er grunnen til at det er lagt til en operativsystemvelger i Oppstartssikkerhetsverktøy.

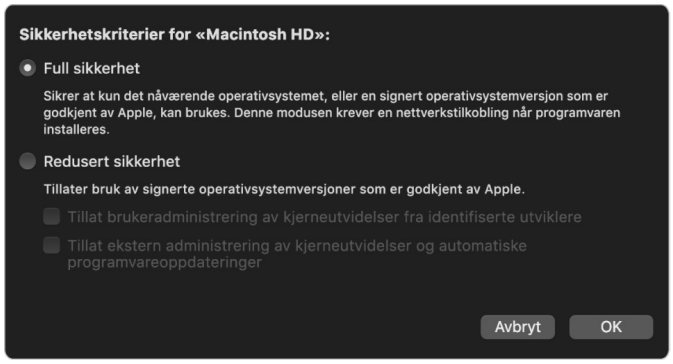
Valg av macOS-lagringsplass for å endre sikkerhetsregelsettet.

På Macer med Apple Silicon indikerer System Security Utility den totale brukerkonfigurerte sikkerhetstilstanden til macOS, for eksempel oppstart av en kjerneutvidelse eller konfigurering av System Integrity Protection (SIP). Hvis endring av en sikkerhetsinnstilling ville svekke sikkerheten betydelig eller gjøre systemet lettere å kompromittere, må brukerne gå inn i recoveryOS ved å holde nede av/på-knappen (slik at skadelig programvare ikke kan utløse signalet, kun et menneske med fysisk tilgang kan gjøre det), for å utføre endringen. På grunn av dette vil heller ikke Apple Silicon-baserte Macer kreve (eller støtte) et firmwarepassord. Alle kritiske endringer styres allerede av brukergodkjenning. Du finner mer informasjon om SIP under System Integrity Protection.

Full sikkerhet og Redusert sikkerhet kan angis ved bruk av Oppstartssikkerhetsverktøy fra recoveryOS. Middels sikkerhet er imidlertid kun tilgjengelig fra kommandolinjeverktøy for brukere som aksepterer risikoen med å gjøre Macen mye mindre sikker.

Full sikkerhet-regelsett

Valget for full sikkerhet er standard, og det fungerer som på iOS og iPadOS. Når programvaren lastes ned og forberedes for installering, vil macOS, i stedet for å bruke den globale signaturen som følger med programvaren, kontakte den samme Apple-signeringstjeneren som brukes for iOS og iPadOS og be om en ny «tilpasset» signatur. En signatur er tilpasset når den inkluderer Exclusive Chip Identification (ECID), som er en unik ID som er spesifikk for Apple-prosessoren i dette tilfellet – som en del av signeringsforespørselen. Signaturen som returneres fra signeringstjeneren, er da unik og kan kun brukes av den spesifikke Apple-prosessoren. Når Full sikkerhet-regelsettet er i bruk, bidrar oppstart-ROM og LLB til å sikre at en gitt signatur ikke bare er signert av Apple, men er signert for denne spesifikke Macen, noe som i praksis binder den aktuelle versjonen av macOS til Macen.



Valg av Full sikkerhet-regelsett i macOS.

Bruk av en nettbasert signeringstjener gir også bedre beskyttelse mot tilbakerullingsangrep enn typiske globale signaturtilnærminger. I et globalt sikkerhetssystem kunne sikkerhetsversjonen ha rullert flere ganger, men et system som aldri har sett den nyeste firmwaren, vil ikke være klar over det. For eksempel vil en datamaskin som tror at den er i sikkerhetsversjon 1 godta programvare fra sikkerhetsversjon 2, selv om den aktuelle sikkerhetsversjonen er 5. Med et nettbasert Apple Silicon-signeringssystem, kan signeringstjeneren avvise oppretting av signaturer for programvare som befinner seg i annet enn den nyeste sikkerhetsversjonen.

Hvis en angriper oppdager en sårbarhet etter et sikkerhetsversjonskifte, kan den heller ikke bare hente den sårbare programvaren fra en tidligere sikkerhetsversjon fra system A og bruke den på system B for å angripe den. Det faktum at den sårbare programvaren fra en eldre sikkerhetsversjon har blitt tilpasset for system A, bidrar til å forhindre at den kan overføres og dermed brukes til å angripe system B. Alle disse mekanismene fungerer sammen og utgjør sterkere garantier for at angripere ikke kan plassere sårbar programvare på en Mac for å omgå beskyttelsen til den nyeste programvaren. En person som har et administratorbrukernavn og passord til Macen, kan imidlertid alltid velge det sikkerhetsregelsettet som passer best for deres bruk.

Redusert sikkerhet-regelsett

Redusert sikkerhet ligner på Middels sikkerhet-adferden på Intel-baserte Macer med T2-brikke, der en leverandør (i dette tilfellet Apple) genererer en digital signatur for koden for å bekrefte at den kommer fra leverandøren. Denne designen bidrar til å hindre angripere i å sette inn usignert kode. Apple kaller denne signaturen en «global» signatur fordi den kan brukes på en hvilken som helst Mac, uten tidsbegrensning, for en Mac som for øyeblikket har et Redusert sikkerhet-regelsett. Redusert sikkerhet gir ikke selv beskyttelse mot tilbakerullingsangrep (selv om uautoriserte endringer i operativsystemet kan gjøre brukerdata utilgjengelig). Du finner mer informasjon om dette under Kjerneutvidelser på Macer med Apple Silicon.

Valg av Redusert sikkerhet-regelsett i macOS.

I tillegg til å gjøre det mulig for brukere å kjøre eldre versjoner av macOS, er Redusert sikkerhet påkrevd for andre handlinger som kan utsette brukerens systemsikkerhet for risiko, for eksempel å introdusere kjerneutvidelser fra tredjeparter. Kjerneutvidelser har de samme rettighetene som kjernen, og derfor kan sårbarheter i kjerneutvidelser fra tredjeparter føre til et fullstendig kompromittert operativsystem. Dette er grunnen til at utviklere oppfordres sterkt til å ta i bruk systemutvidelser før støtte av kjerneutvidelser fjernes fra macOS for fremtidige Macer med Apple Silicon. Selv når kjerneutvidelser fra tredjeparter aktiveres, kan de ikke lastes inn i kjernen ved behov. I stedet slås kjerneutvidelsene sammen i en Auxiliary Kernel Collection (AuxKC), der hashen er lagret i LocalPolicy, noe som krever en omstart. Du finner mer informasjon om generering av AuxKC under Kjerneutvidelser i macOS.

Middels sikkerhet-regelsett

Middels sikkerhet er for brukere som godtar risikoen med å sette Macen inn i en mye mer usikker tilstand. Denne modusen er forskjellig fra Ingen sikkerhet-modus på Intel-baserte Macer med T2-brikke. Med Midddels sikkerhet utføres signaturverifisering fortsatt langs hele den sikre oppstartssekvensen, men hvis regelsettet settes til Valgfri, signaliserer det at iBoot skal akseptere lokale Secure Enclave-signerte oppstartsobjekter, for eksempel en brukergenerert Boot Kernel Collection bygd fra tilpasset XNU-kjerne. På denne måten har Middels sikkerhet også en arkitektegenskap for å kjøre en tilfeldig «fullstendig ikke-godkjent operativsystem»-kjerne. Når en tilpasset oppstartskjernesamling eller et fullstendig ikke-godkjent operativsystem lastes i systemet, blir noen dekrypteringsnøkler utilgjengelige Dette er utviklet for å forhindre at et fullstendig ikke-godkjent operativsystem får tilgang til data fra godkjente operativsystemer.

Viktig: Apple leverer ikke og støtter ikke tilpassede XNU-kjerneutvidelser.

Valg av Middels sikkerhet-regelsett i macOS.

Middels sikkerhet skiller seg også fra Ingen sikkerhet på Intel-baserte Macer med T2-brikke på en annen måte. Det er en forutsetning for noen sikkerhetsnedgraderinger som tidligere har vært uavhengig kontrollerbare. For å deaktivere System Integrity Protection (SIP) på Macer med Apple Silicon må brukeren anerkjenne at de setter systemet til Middels sikkerhet. Dette er påkrevd fordi deaktivering av SIP alltid har satt systemet i en tilstand som gjør kjernen mye enklere å kompromittere. Hvis du deaktiverer SIP på en Mac med Apple Silicon, deaktiveres håndheving av kjerneutvidelsessignatur under AuxKC-genereringstid, noe som tillater at tilfeldige kjerneutvidelser kan lastes inn i kjerneminnet. En annen SIP-forbedring som er gjort på Macer med Apple Silicon, er at regelsettlageret er flyttet ut av NVRAM og inn i LocalPolicy. Dermed krever deaktivering av SIP autentisering av en bruker som har tilgang til signeringsnøkkelen til LocalPolicy fra recoveryOS (ved å holde nede Av/på-knappen). Dette gjør det betydelig mer vanskelig for en som kun angriper programvare, eller til og med en angriper som er fysisk til stede, å deaktivere SIP.

Det er ikke mulig å nedgradere til Middels sikkerhet fra Oppstartssikkerhetsverktøy-appen. Brukerne kan kun nedgradere ved å kjøre kommandolinjeverktøy fra Terminal i recoveryOS, for eksempel `csrutil` (for å deaktivere SIP). Etter at brukeren har nedgradert, gjenspeiles dette i Oppstartssikkerhetsverktøy, slik at en bruker enkelt kan angi sikkerheten til en sikrere modus.

Merk: Macer med Apple Silicon verken krever eller støtter et spesifikt regelsett for medieoppstart siden alle oppstarter teknisk sett utføres lokalt. Hvis en bruker velger å starte opp fra eksterne medier, må operativsystemversjonen først tilpasses ved hjelp av en

autentisert omstart fra recoveryOS. Denne omstarten oppretter en LocalPolicy-fil på den interne stasjonen som brukes til å utføre en godkjent oppstart fra operativsystemet lagret på det eksterne mediet. Dette betyr at konfigurasjonen av oppstarten fra eksterne medier, alltid er eksplisitt aktivert per operativsystem og krever allerede brukergodkjenning, slik at det ikke er nødvendig med ytterligere sikker konfigurering.

Oppretting og administrering av LocalPolicy-signeringsnøkkel

Oppretting

Når macOS installeres for første gang i fabrikken, eller når en sletting og installering utføres når enheten er tilkoblet, kjører Macen kode fra en midlertidig gjenopprettings-RAM-disk for å initialisere standardtilstanden. I løpet av denne prosessen oppretter gjenopprettingsmiljøet et nytt par med offentlige og private nøkler som oppbevares i Secure Enclave. Den private nøkkelen kalles *Owner Identity Key (OIK)*. Hvis det allerede eksisterer en OIK, ødelegges denne som en del av prosessen. Gjenopprettingsmiljøet initialiserer også nøkkelen som brukes for Aktiveringslås: *User Identity Key (UIK)*. En del av prosessen er unik for Macer med Apple Silicon: Når UIK-sertifiseringen forespørres for Aktiveringslås, inkluderes et sett med forespurte begrensninger som skal håndheves ved valideringstidspunktet på LocalPolicy. Det er ikke mulig å få opprettet en LocalPolicy hvis enheten ikke kan få en UIK-sertifisert for Aktiveringslås (for eksempel fordi enheten er knyttet til en Finn Mac-konto og er rapportert som mistet). Hvis en enhet tildeles en *User identity Certificate (ucrt)*, inneholder ucrt tjenerpålagte regelsettbegrensninger og brukerforespurte regelsettbegrensninger i en X.509 v3-utvidelse.

Når en Aktiveringslås/ucrt er hentet, oppbevares den i en database på tjenersiden og sendes også tilbake til enheten. Når enheten har fått en ucrt, sendes en sertifiseringsforespørsel for den offentlige nøkkelen som tilsvarer OIK-en, til *Basic Attestation Authority (BAA)*-tjeneren. BAA verifiserer OIK-sertifiseringsforespørselen ved hjelp av den offentlige nøkkelen fra ucrt som oppbevares i databasen som er tilgjengelig for BAA. Hvis BAA kan verifisere sertifiseringen, sertifiseres den offentlige nøkkelen, og *Owner Identity Certificate (OIC)*, som er signert av BAA og inneholder begrensningene som oppbevares i ucrt, returneres. OIC sendes deretter tilbake til Secure Enclave. Fra nå av, når Secure Enclave signerer en ny LocalPolicy, festes den til OIC i Image4. LLB har innebygd godkjenning i BAA-rotsertifikatet, noe som fører til at den godkjenner OIC, som så fører til at den godkjenner hele LocalPolicy-signaturen.

RemotePolicy-begrensninger

Alle Image4-filer, ikke bare LocalPolicies, inneholder begrensninger for Image4-listeevaluering. Disse begrensningene er kodet med spesielle objektidentifikatorer (OID-er) i bladsertifikatet. Image4-verifiseringsbiblioteket slår opp den spesielle OID-en for sertifikatbegrensning fra et sertifikat under signaturevaluering og evaluerer deretter mekanisk begrensningene som er spesifisert i det. Begrensningene har formen:

- X må eksistere
- X må ikke eksistere
- X må ha en spesifikk verdi

For «tilpassede» signaturer vil sertifikatbegrensningene for eksempel inneholde «ECID må eksistere», og for «globale» signaturer vil den inneholde «ECID må ikke eksistere». Disse begrensningene er utviklet for å sørge for at alle Image4-filer signert av en gitt nøkkel, må oppfylle bestemte krav for å unngå feil signert Image4-listegenerering.

I forbindelse med hver LocalPolicy kalles disse Image4-sertifikatbegrensningene for *RemotePolicy*. En annerledes RemotePolicy kan eksistere for forskjellige oppstartsmiljøers LocalPolicies. RemotePolicy brukes til å begrense LocalPolicy for recoveryOS, så når recoveryOS starter, kan det kun oppføre seg som om det starter med Full sikkerhet. Dette øker tilliten til integriteten i recoveryOS-oppstartsmiljøet som et sted der regelsett kan endres. RemotePolicy begrenser LocalPolicy til å inneholde ECID til Mac der LocalPolicy ble generert, og den spesifikke Remote Policy Nonce Hash (`rpnh`) lagret i komponenten for sikker lagring på den maskinen. `rpnh`, og dermed RemotePolicy, endres kun når det utføres handlinger for Finn Mac og Aktiveringslås, for eksempel registrering, avregistrering, fjernlåsing og fjernsletting. Remote Policy-begrensninger bestemmes og spesifiseres på sertifiseringstidspunktet for User Identity Key (UIK) og signeres inn i det utstedte User identity Certificate (ucrt). Enkelte Remote Policy-begrensninger bestemmes av tjeneren, som ECID, ChipID og BoardID. Dette er utviklet for å hindre at enheter kan signere LocalPolicy-filer for andre enheter. Andre Remote Policy-begrensninger kan spesifiseres av enheten for å bidra til å hindre sikkerhetsnedgradering av Local Policy uten å oppgi begge de lokale autentiseringene som kreves for å få tilgang til den gjeldende OIK-en og fjernautentiseringen for kontoen som enheten er låst til med Aktiveringslås.

Innhold i en LocalPolicy-fil for en Mac med Apple Silicon

LocalPolicy er en Image4-fil signert av Secure Enclave. Image4 er et ASN.1 (Abstract Syntax Notation One) DER-kodet datastrukturformat som brukes til å beskrive informasjon om objekter i den sikre oppstartssekvensen på Apple-plattformer. I en Image4-basert sikker oppstartsmodell blir sikkerhetsregelsett forespurt ved installasjon av programvare, igangsatt av en signeringsforespørsel til en sentral Apple-signeringstjener. Hvis regelsettet var akseptabelt, returnerer signeringstjeneren en signert Image4-fil som inneholder en rekke 4CC-sekvenser (four-character-code). Disse signerte Image4-filene og 4CC-ene evalueres ved oppstart av programvare som oppstart-ROM eller LLB.

Overføring av eierskap mellom operativsystemer

Tilgang til Owner Identity Key (OIK) omtales som «Eierskap» Eierskap kreves for å gi brukerne muligheten til å si opp LocalPolicy etter å ha gjort endringer i regelsett eller programvare. OIK-en beskyttes av det samme nøkkelhierarkiet som beskrives i Sealed Key Protection (SKP), med OIK-en som beskyttes av den samme nøkkelkrypteringsnøkkelen (KEK) som volumkrypteringsnøkkelen (VEK). Dette betyr at den vanligvis beskyttes av både brukerpassord og målinger av operativsystemet og regelsett. Det er kun én OIK for alle operativsystemene på Macen. Derfor kreves eksplisitt samtykke fra brukerne i det første operativsystemet for å overføre eierskap til brukerne i det andre operativsystemet når det andre operativsystemet installeres. Brukere eksisterer imidlertid ikke ennå for det andre operativsystemet når installereren kjøres fra det første operativsystemet. Brukere i operativsystemer genereres vanligvis ikke før operativsystemet startes opp og Oppsettassistent kjører. To nye handlinger kreves dermed ved installering av nytt operativsystem på Macer med Apple Silicon.

- Opprette en LocalPolicy for det andre operativsystemet
- Forberede en «Install User» for overføring av eierskap

Når du kjører en installeringsassistent og retter installeringen mot et sekundært tomt volum, vil du få en forespørsel om brukeren vil kopiere en bruker fra det gjeldende volumet til å være den første brukeren på det sekundære volumet. Hvis brukeren sier ja, er «Install User» som opprettes, faktisk en KEK som avledes fra den valgte brukerens passord og maskinvarenøkler, som deretter brukes til å kryptere OIK-en mens den overføres til det

andre operativsystemet. I installeringsassistenten for det andre operativsystemet kommer en forespørsel om brukerens passord for å gi tilgang til OIK-en i Secure Enclave for det nye operativsystemet. Hvis brukerne velger å ikke kopiere en bruker, opprettes fremdeles Install User på samme måte, men et tomt passord brukes i stedet for en brukers passord. Denne andre flyten eksisterer for enkelte scenarioer for systemadministrering. Brukere som vil installere på flere volumer og ønsker å utføre overføring av eierskap på en sikker måte, bør imidlertid alltid velge å kopiere en bruker fra det første operativsystemet til det andre operativsystemet.

LocalPolicy på Macer med Apple Silicon

For Macer med Apple Silicon er kontroll av lokale sikkerhetsregelsett delegert til en applikasjon som kjører i Secure Enclave. Denne programvaren kan benytte brukerens akkreditiver og oppstartsmodusen til hovedprosessoren for å fastslå hvem som kan endre sikkerhetsregelsettet og fra hvilket oppstartsmiljø. Dette hindrer ondsinnet programvare fra å bruke kontrollene av sikkerhetsregelsettet mot brukeren ved å nedgradere dem for å få flere rettigheter.

Listeegenskaper for LocalPolicy

LocalPolicy-filen inneholder noen arkitektoniske 4CC-er som finnes i de aller fleste Image4-filene, for eksempel en kort-ID eller modell-ID (BORD), som indikerer en bestemt Apple-brikke (CHIP), eller Exclusive Chip Identification (ECID). Men 4CC-er under fokuserer kun på sikkerhetsregelsettene som brukerne kan konfigurere.

Merk: Apple bruker begrepet *One True recoveryOS (1TR)* til å indikere en oppstart i det primære recoveryOS som oppnås ved å trykke fysisk på av/på-knappen. Dette er forskjellig fra en normal recoveryOS-oppstart, som kan oppnås ved bruk av NVRAM, eller som kan skje når det oppstår feil ved oppstart. Det fysiske knappetrykket øker tilliten til at oppstartsmiljøet ikke kan nås av noen som kun angriper programvare, som har brutt seg inn i macOS.

LocalPolicy Nonce Hash (lpnh)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* `lpnh` brukes for anti-repetisjon av LocalPolicy. Dette er en SHA384-hash av LocalPolicy Nonce (LPN), som er lagret i komponenten for sikker lagring og kan nås via oppstart-ROM for Secure Enclave eller Secure Enclave. Den rå nonce-verdien er aldri synlig for applikasjonsprosessoren, kun for sepOS. En angriper som ønsker å overtale LLB om at et tidligere LocalPolicy de hadde fanget opp, var gyldig, vil måtte plassere en verdi i komponenten for sikker lagring, som hasher til den samme `lpnh`-verdien som finnes i LocalPolicy-en de vil repetere. Vanligvis er det kun én gyldig LPN på systemet. Unntaket er under programvareoppdateringer, hvor to er gyldige samtidig for at det skal være mulig å falle tilbake til oppstart av den gamle programvaren ved en eventuell oppdateringsfeil. Når en LocalPolicy for et operativsystem endres, signeres alle regler på nytt med den nye *lpnh*-verdien, som tilsvarer den nye LPN-en i komponenten for sikker lagring. Denne endringen skjer når brukeren endrer sikkerhetsinnstillingene eller oppretter nye operativsystemer med en ny LocalPolicy for hver.

Remote Policy Nonce Hash (rpnh)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* `rpnh` oppfører seg på samme måte som `lpnh`, men oppdateres kun når det eksterne regelsettet oppdateres, for eksempel ved endring av statusen på «Hvor er?»-registrering. Denne endringen skjer når brukeren endrer statusen på «Hvor er?» på Macen.

recoveryOS Nonce Hash (ronh)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* `ronh` oppfører seg på samme måte som `lpnh`, men finnes kun i LocalPolicy for recoveryOS. Den oppdateres når recoveryOS oppdateres, for eksempel ved programvareoppdateringer. Det brukes en egen nonce-verdi fra `lpnh` og `rpnh`, så når en enhet settes i en deaktivert tilstand av «Hvor er?», kan eksisterende operativsystemer deaktiveres (ved å fjerne LPN og RPN fra komponenten for sikker lagring), samtidig som recoveryOS fortsatt kan startes opp. På denne måten kan operativsystemene reaktiveres når systemeieren beviser kontrollen sin over systemet ved å legge inn iCloud-passordet som gjelder for «Hvor er?»-kontoen. Denne endringen skjer når en bruker oppdaterer recoveryOS eller oppretter nye operativsystemer.

Next Stage Image4 Manifest Hash (nsih)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse: nsih*-feltet representerer en SHA384-hash av Image4-listedatastrukturen som beskriver macOS-et som er aktivert. Image4-listen til macOS inneholder målinger for alle oppstartsobjektene, for eksempel iBoot, den statiske godkjenningsbufferen, enhetstreet, Boot Kernel Collection og volumrothashen av det signerte systemvolumet (SSV). Når LLB instrueres om å starte et gitt macOS, er den utviklet for å sørge for at hashen av Image4-listen til macOS tilknyttet iBoot samsvarer med det som er fanget opp i `nsih`-feltet i LocalPolicy. På denne måten fanger `nsih` opp brukerens hensikt om hvilket operativsystem brukeren har opprettet LocalPolicy for. Brukerne kan endre `nsih`-verdien implisitt når de utfører en programvareoppdatering.

Auxiliary Kernel Collection (AuxKC) Policy Hash (auxp)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* macOS
- *Beskrivelse:* `auxp` er en SHA384-hash av regelsettet for brukerautorisert kjerneutvidelsesliste (UAKL). Dette brukes ved generering av AuxKC for å bidra til å sikre at kun brukerautoriserte kjerneutvidelser inkluderes i AuxKC. `smb2` er en forutsetning for å angi dette feltet. Brukere endrer `auxp`-verdien implisitt når de endrer UAKL ved å godkjenne en kjerneutvidelse fra Sikkerhet og personvern-panelet i Systemvalg.

Auxiliary Kernel Collection (AuxKC) Image4 Manifest Hash (auxi)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* macOS
- *Beskrivelse:* Etter at systemet verifiserer at UAKL-hashen samsvarer med det som er i auxp-feltet i LocalPolicy, ber det om at AuxKC signeres av Secure Enclave-prosessorapplikasjonen som er ansvarlig for signering av LocalPolicy. Deretter plasseres en SHA384-hash av AuxKC Image4-listesignaturen i LocalPolicy for å unngå muligheten for å velge og kombinere tidligere signerte AuxKC-er med et operativsystem ved oppstart. Hvis iBoot finner `auxi`-feltet i LocalPolicy, forsøker det å laste inn AuxKC fra lagringsplassen og validere signaturen. Den verifiserer også at hashen av Image4-listen tilknyttet AuxKC samsvarer med verdien i `auxi`-feltet. Hvis AuxKC av en eller annen grunn mislykkes i å laste, fortsetter systemet å starte opp uten dette oppstartsobjektet og dermed uten å laste noen kjerneutvidelser fra tredjeparter. auxp-feltet er en forutsetning for å angi auxi-feltet i LocalPolicy. Brukere endrer `auxi`-verdien implisitt når de endrer UAKL ved å godkjenne en kjerneutvidelse fra Sikkerhet og personvern-panelet i Systemvalg.

Auxiliary Kernel Collection (AuxKC) Receipt Hash (auxr)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* macOS
- *Beskrivelse:* `auxr` er en SHA384-hash av AuxKC-kvitteringen, som indikerer det nøyaktige settet med kjerneutvidelser som ble inkludert i AuxKC. AuxKC-kvitteringen kan være et delsett av UAKL, fordi kjerneutvidelser kan ekskluderes fra AuxKC selv om de er brukerautorisert, hvis det er kjent at de brukes til angrep. I tillegg kan noen kjerneutvidelser som brukes til å bryte brukerkjernegrensen, føre til nedsatt funksjonalitet, for eksempel en manglende evne til å bruke Apple Pay eller spille 4K- og HDR-innhold. Brukere som vil ha disse funksjonene, velger en mer restriktiv AuxKC-inkludering. auxp-feltet er en forutsetning for å angi `auxr`-feltet i LocalPolicy. Brukerne endrer `auxr`-verdien implisitt når de bygger en ny AuxKC fra Sikkerhet og personvern-panelet i Systemvalg.

CustomOS Image4-listesignatur (coih).

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* `coih` er en SHA384-hash av CustomOS Image4-listen. Nyttelasten for den listen brukes av iBoot (i stedet for XNU-kjernen) for å overføre kontroll. Brukere endrer `coih`-verdien implisitt når de bruker kommandolinjeverktøyet `kmutil configure-boot` i 1TR.

APFS volume group UUID (vuid)

- *Type:* OctetString (16)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* `vuid` indikerer volumgruppen som kjernen skal bruke som rot. Dette feltet er hovedsakelig informativt og brukes ikke for sikkerhetsbegrensninger. Denne `vuid` angis implisitt av brukeren ved å opprette en ny operativsysteminstallasjon.

Key encryption key (KEK) Group UUID (kuid)

- *Type:* OctetString (16)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* `kuid` indikerer volumet som ble brukt som startvolum. Nøkkelkrypteringsnøkkelen har vanligvis blitt brukt til databeskyttelse. Det brukes til å beskytte LocalPolicy-signeringsnøkkelen for hvert LocalPolicy. `kuid` angis implisitt av brukeren ved å opprette en ny operativsysteminstallasjon.

Paired recoveryOS Trusted Boot Policy Measurement (prot)

- *Type:* Oktettstreng (48)
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* En paret recoveryOS Trusted Boot Policy Measurement (TBPM) er en spesiell iterativ SHA384-hashberegning over Image4-listen til en LocalPolicy, unntatt nonce-verdier, for å kunne gi en konsekvent måling over tid (fordi nonce-verdier som `lpnh` oppdateres ofte). `prot`-feltet, som kun finnes i hver LocalPolicy for macOS, gir en paring for å indikere LocalPolicy for recoveryOS som tilsvarer LocalPolicy for macOS.

Has Secure Enclave Signed recoveryOS Local Policy (hrlp)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR, recoveryOS, macOS
- *Beskrivelse:* `hrlp` indikerer om `prot`-verdien (over) er målingen av en Secure Enclave-signert LocalPolicy for recoveryOS eller ikke. Hvis ikke signeres LocalPolicy for recoveryOS av Apples nettbaserte signeringstjener, som blant annet signerer macOS Image4-filer.

Secure Multi-Boot (smb0)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR, recoveryOS
- *Beskrivelse:* Hvis `smb0` har verdi og er sann, tillater LLB at neste trinns Image4-liste signeres globalt, i stedet for å kreve en tilpasset signatur. Brukerne kan endre dette feltet ved hjelp av Oppstartssikkerhetsverktøy eller `bputil` for å nedgradere til Redusert sikkerhet.

Secure Multi-Boot (smb1)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* Hvis `smb1` har verdi og er sann, tillater iBoot at objekter som en tilpasset kjernesamling blir Secure Enclave-signert med samme nøkkel som LocalPolicy. Forekomst av `smb0` er en forutsetning for forekomst av `smb1`. Brukerne kan endre dette feltet ved hjelp av kommandolinjeverktøy som `csrutil` eller `bputil` for å nedgradere til Middels sikkerhet.

Secure Multi-Boot (smb2)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* Hvis `smb2` har verdi og er sann, tillater iBoot at Auxiliary Kernel Collection blir Secure Enclave-signert med samme nøkkel som LocalPolicy. Forekomst av `smb0` er en forutsetning for forekomst av `smb2`. Brukerne kan endre dette feltet ved hjelp av Oppstartssikkerhetsverktøy eller `bputil` for å nedgradere til Redusert sikkerhet og aktivere kjerneutvidelser fra tredjeparter.

Secure Multi-Boot (smb3)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* Hvis `smb3` har verdi og er sann, har en bruker på enheten valgt MDM-kontroll over systemet. Forekomst av dette feltet fører til at Secure Enclave-prosessorapplikasjonen som kontrollerer LocalPolicy, aksepterer MDM-autentisering i stedet for å be om lokal brukerautentisering. Brukerne kan endre dette feltet ved hjelp av Oppstartssikkerhetsverktøy eller `bputil` for å aktivere administrert kontroll over kjerneutvidelser fra tredjeparter og programvareoppdateringer. (I macOS 11.2 eller nyere kan MDM også igangsette en oppdatering til nyeste macOS-versjon hvis gjeldende sikkerhetsmodus er Full sikkerhet.)

Secure Multi-Boot (smb4)

- *Type:* Boolsk
- *Foranderlige miljøer:* recoveryOS, macOS
- *Beskrivelse:* Hvis `smb4`har verdi og er sann, har enheten valgt MDM-kontroll over operativsystemet ved hjelp av Apple School Manager eller Apple Business Manager. Forekomst av dette feltet fører til at Secure Enclave-applikasjonen som kontrollerer LocalPolicy, aksepterer MDM-autentisering i stedet for å be om lokal brukerautentisering. Dette feltet endres av MDM-løsningen når den oppdager at en enhets serienummer vises i Apple School Manager eller Apple Business Manager.

System Integrity Protection (sip0)

- *Type:* 64-bit-heltall uten fortegn
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* `sip0` holder de eksisterende SIP-regelsettbitene (System Integrity Protection) som tidligere ble lagret i NVRAM. Nye SIP-regelsettbiter legges til her (i stedet for å bruke LocalPolicy-felter som under), hvis de kun brukes i macOS, og ikke brukes av LLB. Brukerne kan endre dette feltet ved hjelp av `csrutil` fra 1TR for å deaktivere SIP og nedgradere til Middels sikkerhet.

System Integrity Protection (sip1)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* Hvis `sip1` har verdi og er sann, vil iBoot tillate feil for å verifisere SSV-volumrothashen. Brukerne kan endre dette feltet ved hjelp av `csrutil` eller `bputil` fra 1TR.

System Integrity Protection (sip2)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* Hvis sip2 har verdi og er sann, vil ikke iBoot låse maskinvareregisteret *Configurable Text Read-only Region (CTRR)* som merker kjerneminnet som skrivebeskyttet. Brukerne kan endre dette feltet ved hjelp av `csrutil` eller `bputil` fra 1TR.

System Integrity Protection (sip3)

- *Type:* Boolsk
- *Foranderlige miljøer:* 1TR
- *Beskrivelse:* Hvis `sip3` har verdi og er sann, håndhever ikke iBoot den innebygde tillatelseslisten for NVRAM-variabelen boot-args, som ellers ville filtrert valgene sendt til kjernen. Brukerne kan endre dette feltet ved hjelp av `csrutil` eller `bputil` fra 1TR.

Sertifikater og RemotePolicy

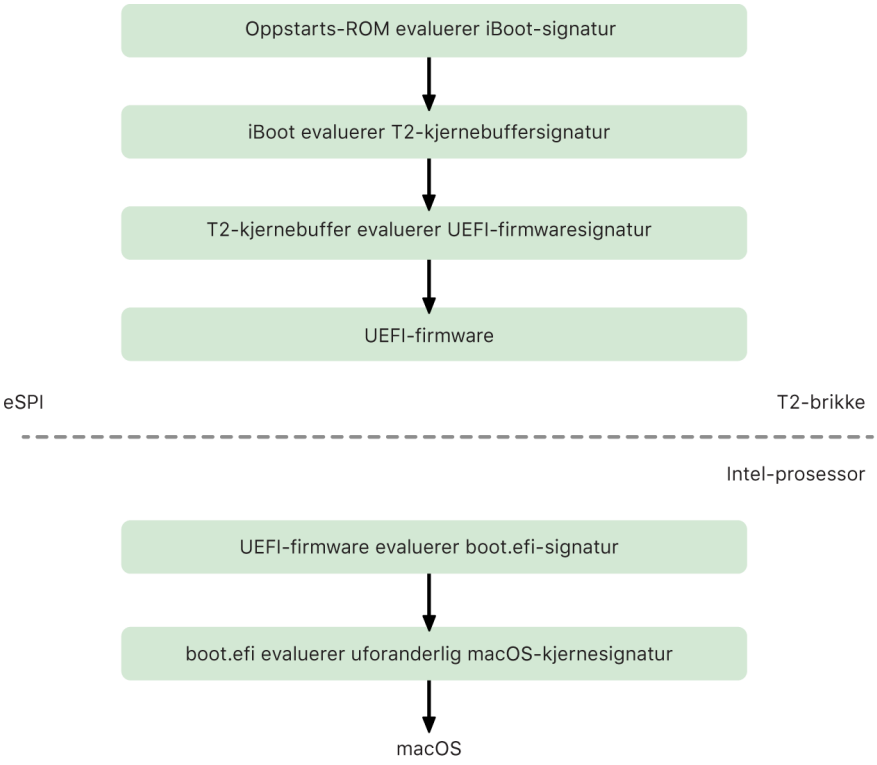
Som beskrevet i Oppretting og administrering av LocalPolicy-signeringsnøkkel, inneholder LocalPolicy Image4 også Owner Identity Certificate (OIC) og den innebygde RemotePolicy.

Intel-baserte Macer

Oppstartsprosess for Intel-baserte Macer

Intel-baserte Macer med Apple T2-sikkerhetsbrikke

Når en Intel-basert Mac med Apple T2-sikkerhetsbrikke slås på, utfører brikken en sikker oppstart fra oppstart-ROM på samme måte som iPhone, iPad og en Mac med Apple Silicon. Dette verifiserer iBoot-bootloaderen og er det første trinnet i godkjenningskjeden. iBoot sjekker kjernen og kjerneutvidelseskoden på T2-brikken, som deretter sjekker Intel UEFI-firmwaren. UEFI-firmwaren og den tilknyttede signaturen er i utgangspunktet kun tilgjengelig for T2-brikken.



Sikker oppstartssekvens i macOS T2.

Etter verifisering lagres UEFI-firmwarefilen i en del av T2-brikkeminnet. Dette minnet gjøres tilgjengelig for Intel-prosessoren via enhanced Serial Peripheral Interface (eSPI). Når Intel-prosessoren først starter, henter den UEFI-firmwaren via eSPI fra den integritetskontrollerte, minnetilordnede kopien av firmwaren som befinner seg på T2-brikken.

Vurderingen av godkjenningskjedet fortsetter på Intel-prosessoren, der UEFI-firmwaren vurderer signaturen for boot.efi, som er macOS-bootloaderen. De Intel-residente signaturene for sikker macOS-oppstart er lagret i samme Image4-format som brukes for iOS, iPadOS og sikker oppstart for T2-brikken, og koden som analyserer Image4-filene, er den samme herdede koden som brukes i nåværende implementering av sikker oppstart i iOS og iPadOS. Boot.efi verifiserer så signaturen til en ny fil kalt immutablekernel. Dette er en kjerne med uforanderlig innhold. Når sikker oppstart er aktivert, representerer immutablekernel-filen hele settet med Apple-kjerneutvidelser som kreves for å starte

macOS. Regelsettet for sikker oppstart avsluttes ved overlevering til immutablekernel, og etter det trer macOS-sikkerhetsregelsett (for eksempel System Integrity Protection og signerte kjerneutvidelser) i kraft.

Hvis det er noen feil eller svikt i denne prosessen, går Macen inn i gjenopprettingsmodus, Apple T2-sikkerhetsbrikke-gjenopprettingsmodus eller Apple T2-sikkerhetsbrikke-DFU-modus (Device Firmware Upgrade).

Microsoft Windows på Intel-baserte Macer med T2-brikke

Standardinnstillingen er at Intel-baserte Macer som støtter sikker oppstart, kun godkjenner innhold som er signert av Apple. For å forbedre sikkerheten til Boot Camp-installeringer tilbys imidlertid også støtte for sikker oppstart av Windows. UEFI-firmwaren inkluderer en kopi av Microsoft Windows Production CA 2011-sertifikatet som brukes til å autentisere Microsoft-bootloadere.

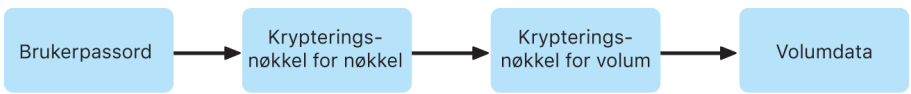
Merk: Det er for øyeblikket ingen godkjenning for Microsoft Corporation UEFI CA 2011, som ville tillatt verifisering av kode signert av Microsoft-partnere. UEFI CA-en brukes vanligvis til å verifisere ektheten av bootloadere for andre operativsystemer, for eksempel varianter av Linux.

Støtte for sikker oppstart av Windows aktiveres ikke som standard. I stedet aktiveres det ved hjelp av Boot Camp-assistent (BCA). Når en bruker kjører BCA, rekonfigureres macOS til å godkjenne Microsoft-førstepartssignert kode under oppstart. Når BCA er ferdig, hvis macOS ikke har lyktes med å bestå Apple-førstepartsgodkjenningsevalueringen under sikker oppstart, forsøker UEFI-firmwaren å evaluere godkjenningen av objektet i henhold til UEFI-formatering for sikker oppstart. Hvis godkjenningsevalueringen lykkes, fortsetter Macen og starter Windows. Hvis ikke, går Macen inn i recoveryOS og informerer brukeren om at godkjenningsevalueringen mislyktes.

Intel-baserte Macer uten en T2-brikke

Intel-baserte Macer uten T2-brikke støtter ikke sikker oppstart. Derfor laster UEFI-firmwaren macOS-booteren (boot.efi) fra filsystemet uten verifisering, og booteren laster inn kjernen (prelinkedkernel) fra filsystemet uten verifisering. For å beskytte integriteten til oppstartskjeden bør brukere aktivere alle av følgende sikkerhetsmekanismer:

- *System Integrity Protection (SIP):* Aktivert som standard. Dette beskytter booteren og kjernen mot ondsinnede skriveoperasjoner fra inne i et kjørende macOS.
- *FileVault:* Dette kan aktiveres på to måter: enten av brukeren eller av en MDM (mobile device management)-administrator. Dette beskytter mot en fysisk tilstedeværende angriper som bruker måldiskmodus til å overskrive booteren.
- *Firmwarepassord:* Dette kan aktiveres på to måter: enten av brukeren eller av en MDM-administrator. Dette bidrar til å hindre en fysisk tilstedeværende angriper fra å starte alternative oppstartsmoduser som recoveryOS, enkeltbrukermodus eller måldiskmodus som booteren kan overskrives fra. Dette bidrar også til å hindre oppstart fra alternative medier, som en angriper kunne kjøre kode fra for å overskrive booteren.



Opplåsingsprosessen på Intel-baserte Macer uten T2-brikke.

Oppstartsmoduser til Intel-baserte Macer med Apple T2-sikkerhetsbrikke

Intel-baserte Macer med Apple T2-sikkerhetsbrikke har en rekke oppstartsmoduser som kan startes ved å holde nede tastekombinasjoner som gjenkjennes av UEFI-firmwaren eller booteren. Enkelte oppstartsmoduser, for eksempel enkeltbrukermodus, vil ikke fungere med mindre sikkerhetsregelsettet endres til Ingen sikkerhet i Oppstartssikkerhetsverktøy.

Modus	Tastaturkombinasjon	Beskrivelse
macOS-oppstart	Ingen	UEFI-firmwaren overleverer til macOS-oppstart (en UEFI-applikasjon), som så overleverer til macOS-kjernen. Ved standard oppstart av Macer der FileVault er aktivert, presenterer macOS-booteren Påloggingsvindu-grensesnittet der passordet oppgis, slik at lagringen kan dekrypteres.
Startup Manager	Tilvalg (⌥)	UEFI-firmwaren starter den innebygde UEFI-applikasjonen som presenterer brukeren med et grensesnitt for å velge oppstartsenhet.
Måldiskmodus (MDM)	T	UEFI-firmwaren starter den innebygde UEFI-applikasjonen som eksponerer den interne lagringsenheten som en rå, blokkbasert lagringsenhet over Fire Wire, Thunderbolt, USB eller en hvilken som helst kombinasjon av de tre (avhengig av Mac-modellen).
Enkeltbrukermodus	Kommando (⌘)-S	macOS-kjernen sender –s-flagget i launchds argumentvektor, og launchd oppretter deretter et enkeltbrukershell i Konsoll-programmets tty. *Merk:* Hvis brukeren avslutter shell-et, fortsetter macOS oppstarten fram til påloggingsvinduet.
recoveryOS	Kommando (⌘)-R	UEFI-firmwaren laster inn en redusert utgave av macOS fra en signert diskfil (.dmg) på den interne lagringsenheten.
Internet recoveryOS	Tilvalg (⌥)-Kommando (⌘)-R	Den signerte diskfilen lastes ned fra internett ved hjelp av HTTP.
Diagnostikk	D	UEFI-firmwaren laster inn et minimalt diagnostisk UEFI-miljø fra en signert diskfil (.dmg) på den interne lagringsenheten.
Internettdiagnostikk	Tilvalg (⌥)-D	Den signerte diskfilen lastes ned fra internett ved hjelp av HTTP.
Windows-oppstart	Ingen	Hvis Windows er installert ved hjelp Boot Camp, overleverer UEFI-firmwaren til Windows-booteren, som overleverer til Windows-kjernen.

Oppstartssikkerhetsverktøy på Macer med Apple T2-sikkerhetsbrikke

Oversikt

Oppstartssikkerhetsverktøy håndterer en rekke innstillinger for sikkerhetsregelsett på Intel-baserte Macer med Apple T2-sikkerhetsbrikke. Verktøyet er tilgjengelig når du starter i recoveryOS og velger Oppstartssikkerhetsverktøy fra Verktøy-menyen, og det beskytter støttede sikkerhetsinnstillinger fra enkel manipulering av en angriper.

Et skjermbilde av Oppstartssikkerhetsverktøy.

Kritiske regelsettendringer krever autentisering, selv i gjenopprettingsmodus. Når Oppstartssikkerhetsverktøy åpnes for første gang, ber den brukeren om å oppgi et administratorpassord fra den primære macOS-installeringen som er knyttet til den startede recoveryOS. Hvis ingen administrator eksisterer, må det opprettes en før regelsettet kan endres. T2-brikken krever at Macen er startet opp i recoveryOS og at autentisering med et Secure Enclave-støttet akkreditiv har funnet sted før en slik regelsettendring kan utføres. Endringer av sikkerhetsregelsett har to underforståtte krav. recoveryOS må:

- Være startet fra en lagringsenhet som er koblet direkte til T2-brikken, ettersom partisjoner på andre enheter ikke har Secure Enclave-støttede akkreditiver knyttet til den interne lagringsenheten.
- Være på et APFS-basert volum, ettersom det kun er støtte for lagring av Authentication in Recovery-akkreditiver som er sendt til Secure Enclave på «Preboot» APFS-volumet til en stasjon. HFS-plus-formaterte volumer kan ikke bruke sikker oppstart.

Dette regelsettet vises kun i Oppstartssikkerhetsverktøy på Intel-baserte Macer med T2-brikke. Selv om de fleste brukstilfeller ikke skulle kreve endringer i regelsettet for sikker oppstart, har brukere i siste instans kontroll over enhetens innstillinger og kan velge, avhengig av behov, å deaktivere eller nedgradere sikker oppstart-funksjonaliteten på Macen.

Endringer i regelsettet for sikker oppstart som gjøres med dette programmet, gjelder kun evaluering av godkjenningskjedet som verifiseres på Intel-prosessoren. Valget for sikker oppstart av T2-brikken er alltid på.

Regelsettet for sikker oppstart kan konfigureres med én av tre innstillinger: Full sikkerhet, Middels sikkerhet og Ingen sikkerhet. Ingen sikkerhet deaktiverer sikker oppstart-evaluering helt på Intel-prosessoren og tillater at brukere kan starte hva de vil.

Full sikkerhet-oppstartsregelsett

Full sikkerhet er standardregelsettet for oppstart, og det fungerer i stor grad som iOS og iPadOS og Full sikkerhet på Macer med Apple Silicon. Når programvaren lastes ned og klargjøres for installering, tilpasses den, som en del av signeringsforespørselen, med en signatur som inkluderer Exclusive Chip Identification (ECID). Dette er en unik ID som i dette tilfellet er spesifikk for T2-brikken. Signaturen som returneres fra signeringstjeneren, er da unik og kan kun brukes av den spesifikke T2-brikken. UEFI-firmwaren er utviklet for å sikre at når Full sikkerhet-regelsettet er i bruk, vil ikke en gitt signatur bare signeres av Apple, men også signeres for denne spesifikke Macen, noe som i praksis binder den aktuelle versjonen av macOS til Macen. Dette bidrar til å forhindre tilbakerullingsangrep, som beskrevet for Full sikkerhet på Macer med Apple Silicon.

Middels sikkerhet-oppstartsregelsett

Middels sikkerhet-oppstartsregelsett har likhetstrekk med en sikker oppstart med UEFI, der en leverandør (Apple i dette tilfellet) genererer en digital signatur for koden for å bekrefte at den kommer fra den forhandleren. På denne måten hindres angripere i å sette inn usignert kode. Vi kaller denne signaturen en «global» signatur fordi den kan brukes på en hvilken som helst Mac, uten tidsbegrensning, for en Mac som for øyeblikket har et Middels sikkerhet-regelsett. Verken iOS, iPadOS eller selve T2-brikken støtter globale signaturer. Denne innstillingen forsøker ikke å forhindre tilbakerullingsangrep.

Regelsett for medieoppstart

Regelsettet for medieoppstart eksisterer kun på Intel-baserte Macer med T2-brikke og er uavhengig av regelsettet for sikker oppstart. Selv om en bruker deaktiverer sikker oppstart, endrer det ikke standardadferden med å forhindre at noe annet enn lagringsenheten som er direkte koblet til T2-brikken, starter Macen. (Regelsett for medieoppstart er ikke påkrevd på Macer med Apple Silicon.) Du finner mer informasjon i Kontroll av sikkerhetsregelsett for Startdisk.

Firmwarepassordbeskyttelse på Intel-baserte Macer

macOS på Intel-baserte Macer med en Apple T2-sikkerhetsbrikke støtter bruk av et firmwarepassord for å bidra til å hindre utilsiktede endringer av firmwareinnstillinger på en bestemt Mac. Firmwarepassordet er utviklet for å hindre valg av alternative oppstartsmoduser som oppstart i recoveryOS eller enkeltbrukermodus, oppstart fra et uautorisert volum eller oppstart i måldiskmodus.

Merk: Firmwarepassordet er ikke påkrevd på Macer med Apple Silicon fordi den kritiske firmwarefunksjonaliteten det begrenset, er flyttet inn i recoveryOS, og (når FileVault er aktivert) recoveryOS krever brukerautentisering før den kritiske funksjonaliteten kan nås.

Den enkleste modusen for firmwarepassord nås fra Firmwarepassordverktøy i recoveryOS på Intel-baserte Macer *uten* T2-brikker og fra Oppstartssikkerhetsverktøy på Intel-baserte Macer *med* T2-brikke. Avanserte valg (for eksempel muligheten til å be om passordet ved hver oppstart) er tilgjengelige fra `firmwarepasswd`-kommandolinjeverktøyet i macOS.

Det er spesielt viktig å angi et firmwarepassord for å redusere risikoen for angrep på Intel-baserte Macer uten T2-brikke fra angripere som er fysisk til stede. Firmwarepassordet kan

hindre en angriper i å starte opp til recoveryOS, der System Integrity Protection (SIP) ellers kan deaktiveres. Og ved å begrense oppstart av alternative medier, kan en angriper ikke kjøre privilegert kode fra et annet operativsystem for å angripe perifer firmware.

Det finnes en mekanisme for å nullstille firmwarepassord i tilfelle brukere glemmer passordet. Brukere holder nede en tastekombinasjon ved oppstart og vises en modellspesifikk streng de oppgir til AppleCare. AppleCare signerer en ressurs digitalt som signaturkontrolleres av Uniform Resource Identifier (URI). Hvis signaturen valideres og innholdet er for den spesifikke Macen, fjerner UEFI-firmwaren firmwarepassordet.

For brukere som ikke ønsker at andre enn dem selv skal kunne fjerne firmwarepassordet ved hjelp av programvare, ble `–disable–reset–capability`-valget lagt til i `firmwarepasswd`-kommandolinjeverktøyet i macOS 10.15. Før dette valget angis, må brukere godta at hvis passordet glemmes og må fjernes, er de selv ansvarlige for kostnaden knyttet til bytte av hovedkort som kreves for å oppnå dette. Organisasjoner som vil beskytte Macer fra eksterne angripere og fra ansatte, må angi et firmwarepassord på organisasjonseide systemer. Dette kan gjennomføres på enheten på en av følgende måter:

- Ved klargjøring, ved å manuelt bruke `firmwarepasswd`-kommandolinjeverktøyet
- Med tredjepartsadministrasjonsverktøy som bruker `firmwarepasswd`-kommandolinjeverktøyet
- Med MDM (Mobile Device Management)

recoveryOS og diagnostikkmiljøer for Intel-baserte Macer

recoveryOS

recoveryOS er helt separat fra hoveddelen av macOS, og alt innholdet lagres i en diskfil med navnet BaseSystem.dmg. Det finnes også en tilknyttet BaseSystem.chunklist som brukes til å verifisere integriteten til BaseSystem.dmg. Chunklisten er en serie med hasher for 10 MB-deler av BaseSystem.dmg. UEFI-firmwaren evaluerer signaturen til chunklistfilen, og evaluerer deretter hashen én del om gangen fra BaseSystem.dmg. Dette bidrar til å sikre at den tilsvarer det signerte innholdet i chunklisten. Hvis en av disse hashene ikke stemmer overens, avbrytes oppstart fra det lokale recoveryOS, og UEFI-firmwaren forsøker å starte fra Internet recoveryOS i stedet.

Hvis verifiseringen fullføres, aktiverer UEFI-firmwaren BaseSystem.dmg som en RAM-disk og starter boot.efi-filen som er i den. Det er ikke nødvendig for UEFI-firmwaren å kontrollere boot.efi spesifikt, eller at boot.efi kontrollerer kjernen, ettersom det ferdige innholdet i operativsystemet (som disse elementene kun er et delsett av) allerede er integritetskontrollert.

Apple-diagnostikk

Prosedyren for å starte det lokale diagnostiske miljøet er stort sett det samme som å starte recoveryOS. Separate filer for AppleDiagnostics.dmg og AppleDiagnostics.chunklist brukes, men de verifiseres på samme måte som BaseSystem-filene. I stedet for å starte boot.efi, starter UEFI-firmwaren en fil inni diskfilen (.dmg-filen) med navnet diags.efi, som så er ansvarlig for å starte en rekke andre UEFI-drivere som kan samhandle med og se etter feil i maskinvaren.

Internet recoveryOS og diagnostikkmiljø

Hvis en feil oppstår i forbindelse med start av lokale gjenopprettings- eller diagnostikkmiljøer, forsøker UEFI-firmwaren å laste ned diskfilene fra internett i stedet. (En bruker kan også spesifikt be om at diskfilene hentes fra internett ved hjelp av spesielle tastaturkombinasjoner ved oppstart.) Integritetsverifiseringen av diskfilene og chunklistene som lastes ned fra OS Recovery Server utføres på samme måte som for diskfiler som hentes fra en lagringsenhet.

Selv om forbindelsen til OS Recovery Server gjennomføres ved hjelp av HTTP, er alt nedlastet innhold underlagt integritetskontroller som tidligere beskrevet, og er dermed ikke beskyttet mot manipulering av et angrep med kontroll over nettverket. Hvis en enkeltdel ikke består integritetsverifiseringen, sendes det forespørsel fra OS Recovery Server opptil 11 ganger før det vises en feilmelding.

Da internettgjenopprettings- og diagnostikkmodusene ble lagt til Macer i 2011, ble det bestemt at det ville være bedre å bruke den enklere HTTP-transporten og håndtere innholdsautentisering ved bruk av chunklistmekanismen, i stedet for å implementere den mer kompliserte HTTPS-funksjonaliteten i UEFI-firmwaren, og dermed øke firmwarens angrepsflate.

Sikre programvareoppdateringer

Oversikt

Sikkerhet er en prosess. Det er ikke nok å starte operativsystemversjonen installert på fabrikken, det må også finnes en mekanisme for å hente de nyeste sikkerhetsoppdateringene på en rask og sikker måte. Med jevne mellomrom lanserer Apple programvareoppdateringer som retter opp sikkerhetsproblemer som har oppstått. Brukere av iOS- og iPadOS-enheter mottar oppdateringsvarsler på enheten. Mac-brukere finner tilgjengelige oppdateringer i Systemvalg. Oppdateringer leveres trådløst, for rask utrulling av nye sikkerhetsoppdateringer.

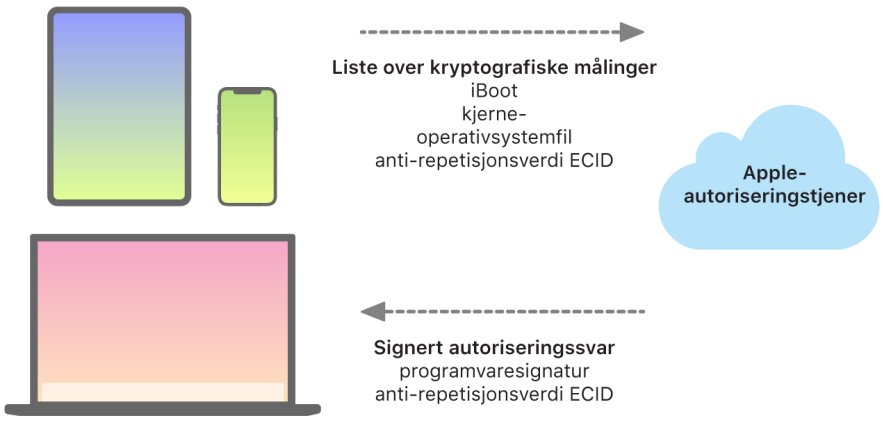
Oppdateringsprosessen bruker samme maskinvarebaserte «root of trust» som sikker oppstart som er utviklet for å installere Apple-signert kode. Oppdateringsprosessen bruker også systemprogramvareautorisering til å kontrollere at kun operativsystemversjoner som aktivt signeres av Apple, kan installeres på iOS- og iPadOS-enheter, eller på Macer med Full sikkerhet-innstillingen konfigurert som regelsettet for sikker oppstart i Oppstartssikkerhetsverktøy. Med disse sikre prosessene på plass kan Apple slutte å signere eldre operativsystemversjoner med kjente sårbarheter, noe som bidrar til å forhindre nedgraderingsangrep.

Hvis enheten som skal oppgraderes, er fysisk koblet til en Mac, vil det av sikkerhetsmessige årsaker lastes ned og installeres en komplett kopi av iOS eller iPadOS. For trådløse programvareoppdateringer er det *kun komponentene som kreves for å fullføre en oppdatering, som lastes ned*. Dette reduserer belastningen på nettverket. I tillegg kan programvareoppdateringer bufres på Macer som kjører macOS 10.13 eller nyere med innholdsbufring slått på, slik at iOS- og iPadOS-enheter ikke trenger å laste ned den nødvendige programvareoppdateringen på nytt over internett. (De må fortsatt kontakte Apple-tjenere for å fullføre oppdateringsprosessen.)

Tilpasset oppdateringsprosess

Ved oppgraderinger og oppdateringer opprettes en forbindelse til Apples tjener for installeringsgodkjenning som inkluderer en liste med kryptografiske målinger for hver del av pakken som skal installeres (for eksempel iBoot, kjernen og operativsystemfilen), en tilfeldig anti-repetisjonsverdi (nonce) og enhetens unike Exclusive Chip Identification (ECID).

Godkjenningstjeneren kontrollerer listen med målinger mot versjoner som det er tillatt å installere. Hvis den finner et samsvarende treff, legger den ECID-en til i målingen og signerer resultatet. Tjeneren sender et komplett sett med signerte data til enheten som en del av oppgraderingsprosessen. Når ECID-en blir lagt til, tilpasses godkjenningen for den aktuelle enheten. Når det bare er kjente målinger som godkjennes og signeres, bidrar tjeneren til å sikre at oppdateringen skjer akkurat slik den ble levert av Apple.



Hvordan Apple-enheter samhandler med Apple-autoriseringstjeneren.

Sikkerhetskjeden verifiserer at signaturen kommer fra Apple, og at målingen av objektet som er lastet fra lagringsenheten, kombinert med enhetens ECID, samsvarer med det som ble omfattet av signaturen. Disse trinnene er utviklet for å sikre at på enheter som støtter tilpassing, gjelder autoriseringen for en spesifikk enhet og at et eldre operativsystem med en firmwareversjon fra én enhet ikke kan kopieres til en annen. Nonce-verdien bidrar til å hindre uvedkommende i å lagre tjenerens svar og bruke det til å manipulere en enhet eller endre systemprogramvaren på andre måter.

Tilpasningsprosessen er grunnen til at en nettverksforbindelse til Apple alltid er påkrevd ved oppdatering av enheter med Apple-designede brikker og Intel-baserte Macer med Apple T2-sikkerhetsbrikken.

Til slutt: Brukerens datavolum aktiveres aldri under en programvareoppdatering, noe som bidrar til å forhindre at noe leses fra eller skrives til dette volumet under oppdateringer.

På enheter med Secure Enclave tar denne maskinvaren også i bruk systemprogramvareautorisering for sjekke integriteten til programvaren, og den er utviklet for å hindre at det installeres nedgraderinger.

Operativsystemintegritet

Apples operativsystem er utviklet med fokus på sikkerhet. Det inkluderer en maskinvarebasert «root of trust» – som brukes for å sørge for sikker oppstart – og en sikker prosess for programvareoppdateringer som er både rask og trygg. Apples operativsystemer bruker også spesialbygde, silisiumbaserte maskinvarefunksjoner for å bidra til å forhindre utnyttelse når systemet kjører. Disse funksjonene ved kjøring beskytter integriteten til godkjent kode når den kjøres. Kort fortalt bidrar Apples operativsystem til å redusere angreps- og utnyttelsesteknikker – uansett om de stammer fra en ondsinnet app, fra internett eller gjennom en annen kanal. Beskyttelsene som er oppført her, er tilgjengelige på enheter med støttede Apple-designede SoC-er, inkludert iOS, iPadOS, tvOS og watchOS, og nå også macOS på Macer med Apple Silicon.

Funksjon	A10	A11, S3	A12, S4	A13, S5	A14, S6	M1
Kernel Integrity Protection	✓	✓	✓	✓	✓	✓
Raske rettighetsrestriksjoner		✓	✓	✓	✓	✓
System Coprocessor Integrity Protection			✓	✓	✓	✓
Pointer Authentication Codes			✓	✓	✓	✓
Sidebeskyttelseslag		✓	✓	✓	✓	Se notat nedenfor.

Merk: PPL (Page Protection Layer) krever at plattformen *kun* kjører signert og godkjent kode. Dette er en sikkerhetsmodell som ikke er gjeldende for macOS.

Kernel Integrity Protection

Etter at operativsystemkjernen fullfører initialisering, aktiveres Kernel Integrity Protection (KIP) for å bidra til å hindre modifisering av kjerne- og driverkode. Minnekontrolleren har et beskyttet område i fysisk minne der iBoot laster inn kjernen og kjerneutvidelser. Når oppstart er fullført, hindrer minnekontrolleren skriving til det beskyttede minneområdet. Applikasjonsprosessorens minneadministreringsenhet (MMU) er konfigurert for å bidra til å forhindre tilordning av privilegert kode fra fysisk minne utenfor det beskyttede minneområdet, og for å bidra til å forhindre skrivbare tilordninger av fysisk minne innen kjerneminneområdet.

For å hindre rekonfigurering, låses maskinvaren som brukes til å aktivere KIP etter at oppstartsprosessen fullføres.

Raske rettighetsrestriksjoner

Fra og med Apple A11 Bionic og S3-SoC-er ble nye maskinvaregrunnelementer introdusert. Disse grunnelementene (raske rettighetsrestriksjoner) inkluderer et CPU-register som raskt begrenser rettigheter etter tråd. Med raske rettighetsrestriksjoner (også kjent som APRR), kan støttede operativsystemer fjerne kjørerettigheter fra minnet uten å ta belastningen som oppstår ved et systemkall og en sidetabellgjennomgang/rydding. Disse registrene reduserer risikoen for angrep fra internett ytterligere, spesielt kode som er kompilert ved kjøring (JIT-kompilert – Just-in-time), siden minnet faktisk ikke er kjørbart samtidig som det kan leses og skrives.

System Coprocessor Integrity Protection

Koprosessorfirmware håndterer mange kritiske systemoppgaver, for eksempel Secure Enclave, bildesensorprosessoren og koprosessoren for bevegelse. Dennes sikkerhet er derfor en viktig del av sikkerheten til hele systemet. For å hindre modifisering av koprosessorfirmwaren bruker Apple en mekanisme som kalles *System Coprocessor Integrity Protection (SCIP)*.

SCIP fungerer på mange måter som Kernel Integrity Protection (KIP). Ved oppstart laster iBoot hver koprosessors firmware inn i et beskyttet minneområde, et som er reservert og separat fra KIP-området. iBoot konfigurerer hver koprosessors minneenhet for å bidra til å forhindre:

- utførbare tilordninger utenfor sin del av det beskyttede minneområdet
- skrivbare tilordninger innenfor sin del av det beskyttede minneområdet

Også ved oppstart, for å konfigurere SCIP for Secure Enclave, brukes Secure Enclave-operativsystemet. Etter at oppstartsprosessen er fullført, låses maskinvaren som brukes til aktivere SCIP. Dette er utviklet for å hindre rekonfigurering.

Pointer Authentication Codes

Pointer Authentication Codes (PAC-er) brukes til å beskytte mot utnyttelse av feil i minnet. Systemprogramvare og innebygde apper bruker PAC til å bidra til å forhindre modifisering av funksjonspekere og returadresser (kodepekere). PAC bruker fem hemmelige 128-bit-verdier til å signere kjerneinstruksjoner og data, og hver brukerområdeprosess har sine egne B-nøkler. Objekter saltes og signeres som vist under.

Objekt	Nøkkel	Salt
Funksjonsreturadresse	IB	Lagringsadresse
Funksjonspekere	IA	0
Blokkoppkallsfunksjon	IA	Lagringsadresse
Objective-C-metodebuffer	IB	Lagringsadresse + Klasse + Velger
C++ V-tabelloppføringer	IA	Lagringsadresse + hash (mangled method name)
Beregnet Goto-etikett	IA	Hash (funksjonsnavn)
Kjernetrådtilstand	GA	•

Objekt	Nøkkel	Salt
Brukers trådtilstandsregistre	IA	Lagringsadresse
C++ V-tabellpekere	DA	0

Signaturverdien lagres i de ubrukte utfyllings-bit-ene øverst i 64-bitspekeren. Signaturen verifiseres før bruk, og utfyllingen gjenopprettes for å bidra til å sikre en fungerende pekeradresse. Hvis verifiseringen mislykkes, avbrytes det. Denne verifiseringen vanskeliggjør mange angrep, for eksempel Return-Oriented Programming-angrep (ROP), som forsøker å lure enheten til å kjøre eksisterende kode på en ondsinnet måte ved å manipulere funksjonsreturadresser lagret på stabelen.

Sidebeskyttelseslag

Page Protection Layer (PPL) i iOS, iPadOS og watchOS er utviklet for å hindre brukerområdekode i å bli modifisert etter at kodesignaturverifisering er fullført. PPL bygger på KIP og Raske rettighetsrestriksjoner og styrer sidetabellrettighetsoverstyringene for å sikre at kun PPL kan endre beskyttede sider som inneholder brukerkode og sidetabeller. Systemet tilbyr en enorm reduksjon i angrepsflate ved å støtte systemomfattende kodeintegritetshåndheving, selv i møte med en kompromittert kjerne. Denne beskyttelsen tilbys ikke på macOS fordi PPL kun gjelder på systemer der all kjørt kode må signeres.

Flere systemsikkerhetsfunksjoner i macOS

Flere systemsikkerhetsfunksjoner i macOS

macOS kjøres på et bredere sett med maskinvare (for eksempel Intel-baserte prosessorer, Intel-baserte prosessorer i kombinasjon med Apple T2-sikkerhetsbrikken og Apple Silicon-baserte SoC-er) og støtter en rekke generelle brukstilfeller innen databehandling. Der noen brukere kun bruker de grunnleggende, forhåndsinstallerte programmene eller programmer som er tilgjengelig fra App Store, er andre kjernehackere som trenger å deaktivere så godt som alle plattformbeskyttelser, slik at de kan kjøre og teste koden mens den kjøres, med de høyeste godkjenningsnivåene. De fleste ligger på et sted i midten, og mange av dem har eksterne enheter og programvare som krever ulike tilgangsnivåer. Apple utviklet macOS-plattformen med en integrert tilnærming til maskinvare, programvare og tjenester. Plattformen tilbyr sikkerhet fra bunnen av og gjør den enkel å konfigurere, rulle ut og administrere, men den beholder konfigurasjonsmulighetene som brukerne forventer. macOS har også de viktigste sikkerhetsteknologiene IT-avdelingen trenger for å beskytte bedriftsdata og integrere løsningen i sikre bedriftsnettverksmiljøer.

Følgende egenskaper støtter og hjelper til med å sikre de ulike behovene til macOS-brukere. De omfatter:

- Sikkerhet for signert systemvolum
- System Integrity Protection
- Godkjenningsbuffere
- Beskyttelse for eksterne enheter
- Støtte og sikkerhet for Rosetta 2 (automatisk oversettelse) for Macer med Apple Silicon

- Støtte og beskyttelse for DMA
- Støtte og sikkerhet for kjerneutvidelser
- Støtte og sikkerhet for Option ROM
- UEFI-firmwaresikkerhet for Intel-baserte Macer

Sikkerhet for signert systemvolum i macOS

I macOS 10.15 introduserte Apple det skrivebeskyttede systemvolumet, et reservert, isolert volum for systeminnhold. macOS 11 legger til sterk, kryptografisk beskyttelse for systeminnhold på et *signert systemvolum (SSV)*. SSV har en kjernemekanisme som verifiserer integriteten til systeminnholdet ved kjøring og avviser alle data, både kode og ikke-kode, som ikke har en gyldig kryptografisk signatur fra Apple.

SSV hjelper ikke bare til med å forhindre manipulering av Apple-programvare som er en del av operativsystemet, men det gjør også macOS-programvareoppdateringer mer pålitelige og mye tryggere. Og siden SSV benytter APFS-øyeblikksbilder (Apple File System) kan den gamle systemversjonen gjenopprettes uten ny installering hvis en oppdatering ikke kan utføres.

APFS har siden introduksjonen dannet integritet for filsystemets metadata ved å bruke ikke-kryptografiske kontrollsummer på den interne lagringsenheten. SSV styrker integritetsmekanismen ved å legge til kryptografiske hasher, noe som utvider den for å omfatte alle byte av fildata. Data fra den interne lagringsenheten (inkludert filsystemets metadata) blir kryptografisk hashet i lesebanen, og hashen sammenlignes deretter med en forventet verdi i filsystemets metadata. I tilfeller med manglende samsvar antar systemet at dataene har blitt tuklet med, og det vil ikke returnere dem til programvaren som sender forespørselen.

Hver SSV SHA256-hash lagres i hovedfilsystemets metadatatre, som selv er hashet. Og fordi hver node av treet rekursivt verifiserer integriteten av hashene til de underordnede elementene, tilsvarende et binært hashtre (Merkle), omfatter rotnodens hashverdi, kalt et *segl*, alle byte av data i SSV, noe som betyr at den kryptografiske signaturen dekker hele systemvolumet.

Under installasjon og oppdatering av macOS beregnes det seglet på nytt fra filsystemet på enheten, og denne målingen verifiseres mot målingen som Apple signerte. På Macer med Apple Silicon verifiserer bootloaderen seglet før kontrollen overføres til kjernen. På Intel-baserte Macer med Apple T2-sikkerhetsbrikke videresender bootloaderen målingen og signaturen til kjernen, som deretter verifiserer seglet direkte før den aktiverer rotfilsystemet. I begge tilfeller, hvis verifiseringen mislykkes, stopper oppstartsprosessen, og brukeren blir bedt om å installere macOS på nytt.

Denne prosedyren gjentas ved hver oppstart med mindre brukeren har valgt å gå inn i en lavere sikkerhetsmodus og har selv valgt å deaktivere det signerte systemvolumet.

SSV og kodesignering

Kodesignering er fortsatt i bruk og håndheves av kjernen. Det signerte systemvolumet gir beskyttelse for når hvilke som helst byte leses fra den interne lagringsenheten. Kodesignering gir derimot beskyttelse når Mach-objekter minnetilordnes som kjørbare. Både SSV og kodesignering beskytter kjørbar kode på alle lese- og kjøringsbaner.

SSV og FileVault

I macOS 11 utføres tilsvarende beskyttelse av systeminnhold på disk av SSV, og systemvolumet trenger derfor ikke lenger å krypteres. Alle endringer utført i filsystemet mens det er på disk, vil oppdages av filsystemet når de leses. Hvis brukeren har aktivert FileVault, er brukerens innhold på datavolumet fortsatt kryptert av en brukeroppgitt hemmelighet.

Hvis brukeren velger å deaktivere SSV, blir systemet sårbart for manipulering, og denne manipuleringen kan gjøre det mulig for en angriper å trekke ut krypterte brukerdata neste gang systemet starter opp. Derfor vil ikke systemet tillate at brukeren deaktiverer SSV hvis FileVault er aktivert. Beskyttelse av diskinnhold må aktiveres eller deaktiveres for begge volumer på en konsekvent måte.

I macOS 10.15 og eldre beskytter FileVault operativsystemprogramvare på disk ved å kryptere bruker- og systeminnhold med en nøkkel beskyttet av en brukeroppgitt hemmelighet. Dette hindrer at en angriper med fysisk tilgang til enheten får tilgang til eller gjør endringer i filsystemet som inneholder systemprogramvaren.

SSV og Macer med Apple T2-sikkerhetsbrikke

På Macer med Apple T2-sikkerhetsbrikke er kun selve macOS beskyttet av SSV. Programvaren som kjører på T2-brikken og verifiserer macOS, er beskyttet av sikker oppstart.

System Integrity Protection

macOS bruker kjernetillatelser til å begrense skrivbarheten til kritiske systemfiler med en funksjon som kalles *System Integrity Protection (SIP)*. Denne funksjonen er atskilt fra og er i tillegg til maskinvarebasert Kernel Integrity Protection (KIP) som er tilgjengelig på Macer med Apple Silicon. Kernel Integrity Protection beskytter modifisering av kjernen i minnet. Det benyttes obligatorisk tilgangskontrollteknologi for å tilby dette og en rekke andre kjernenivåbeskyttelser, inkludert sandkasseteknologi og datahvelv.

Obligatoriske tilgangskontroller

macOS bruker obligatoriske tilgangskontroller. Dette er regelsett som angir sikkerhetsrestriksjoner som opprettes av utvikleren og som ikke kan overstyres. Denne tilnærmingen er forskjellig fra valgfrie tilgangskontroller, som gjør det mulig for brukere å overstyre sikkerhetsregelsett etter eget ønske.

Obligatoriske tilgangskontroller er ikke synlige for brukere, men de er den underliggende teknologien som bidrar til å muliggjøre flere viktige funksjoner, inkludert sandkasseteknologi, foreldrekontroller, administrerte valg, tillegg og System Integrity Protection.

System Integrity Protection

System Integrity Protection skrivebeskytter komponenter i spesifikke, kritiske filsystemplasseringer for å bidra til å forhindre ondsinnet kode i å endre dem. System Integrity Protection er en maskinspesifikk innstilling som er på som standard når en bruker oppgraderer til OS X 10.11 eller nyere. Deaktivering på Intel-baserte Macer fjerner

beskyttelse for alle partisjoner på den fysiske lagringsenheten. macOS bruker dette sikkerhetsregelsettet på alle prosesser som kjører på systemet, uavhengig av om de kjører med sandkasseteknologi eller med administratorrettigheter.

Godkjenningsbuffere

Ett av objektene som er inkludert i den sikre oppstartssekvensen, er den statiske godkjenningsbufferen, en godkjent oppføring av alle Mach-O-binærfilene som er integrert i det signerte systemvolumet. Hver Mach-O representeres av en kodekataloghash. Disse hashene sorteres før innsetting i godkjenningsbufferen for å legge til rette for effektiv søking. Kodekatalogen er resultatet av signeringsoperasjonen utført av `codesign(1)`. Håndheving av godkjenningsbufferen krever at SIP forblir aktivert. Hvis håndheving av godkjenningsbuffer på en Mac med Apple Silicon skal deaktiveres, må sikker oppstart konfigureres til Middels sikkerhet.

Når en binærfil kjøres (enten som en del av oppretting av en ny prosess eller tilordning av kjørbar kode i en eksisterende prosess), blir kodekatalogen trukket ut og hashet. Hvis den resulterende hashen finnes i godkjenningsbufferen, vil de kjørbare tilordningene opprettet for binærfilen få plattformrettigheter. Det vil si at de kan ha en hvilken som helst rettighet og kjøre uten ytterligere verifisering med hensyn til signaturens autentisitet. Dette står i kontrast til Intel-baserte Macer, der plattformrettigheter overføres til operativsysteminnhold av Apple-sertifikatet som signerer binærfilene. (Dette sertifikatet begrenser ikke hvilke rettigheter binærfilen kan inneholde.)

Binærfiler som ikke er plattformrelaterte (for eksempel attestert kode fra tredjeparter), må ha gyldige sertifikatkjeder for å kunne kjøre, og rettighetene de kan inneholde, begrenses av signeringsprofilen utstedt av Apple Developer-programmet til utvikleren.

Alle binærfiler som leveres med macOS, signeres med en *plattformidentifikator*. På Macer med Apple Silicon brukes denne identifikatoren til å indikere at kodekataloghashen må være til stede i godkjenningsbufferen for å kunne kjøre, selv om binærfilen signeres av Apple. På Intel-baserte Macer brukes plattformidentifikatoren til å utføre målrettet tilbakekalling av binærfiler fra eldre versjoner av macOS. Denne målrettede tilbakekallingen bidrar til å forhindre at binærfilene kjører på nyere versjoner.

Den statiske godkjenningsbufferen låser et sett med binærfiler fullstendig til en gitt versjon av macOS. Dette bidrar til å forhindre at legitime Apple-signerte binærfiler fra eldre operativsystemer introduseres i nyere operativsystemer for å gi en angriper fordeler.

Plattformkode levert utenfor operativsystemet

Apple leverer noen binærfiler, deriblant Xcode og samlingen med utviklingsverktøy, som ikke er signert med en plattformidentifikator. De har fortsatt lov til å kjøre med plattformrettigheter på Macer med Apple Silicon og Macer med T2-brikke. Siden denne plattformprogramvaren leveres uavhengig av macOS, er den ikke underlagt tilbakekallingsatferden pålagt av den statiske godkjenningsbufferen.

Kjøreklare godkjenningsbuffere

Apple leverer bestemte programvarepakker med *kjøreklare godkjenningsbuffere*. Disse bufferne har den samme datastrukturen som den statiske godkjenningsbufferen. Men selv

om det bare finnes én statisk godkjenningsbuffer, og det er garantert at innholdet alltid er låst inn i skrivebeskyttede områder etter at kjernens tidlige initialisering er fullført, legges kjøreklare godkjenningsbuffere til i systemet ved kjøring.

Disse godkjenningsbufferne autentiseres enten gjennom den samme mekanismen som autentiserer oppstartsfirmware (tilpasning ved hjelp av den Apple-godkjente signeringstjenesten), eller som globalt signerte objekter (der signaturene ikke binder dem til en bestemt enhet).

Et eksempel på en tilpasset godkjenningsbuffer er en godkjenningsbuffer som leveres med diskfilen som brukes til å utføre feltdiagnostikk på Macer med Apple Silicon. Denne godkjenningsbufferen tilpasses, sammen med diskfilen, og lastes inn i Macens kjerne mens den startes opp i diagnostikkmodus. Godkjenningsbufferen tillater at programvaren i diskfilen kan kjøre med plattformrettighet.

Et eksempel på en globalt signert godkjenningsbuffer leveres med macOS-programvareoppdateringer. Denne godkjenningsbufferen tillater at en del av koden i programvareoppdateringen (*oppdateringshjernen*), kjører med plattformrettighet. Oppdateringshjernen utfører alle oppgaver for å klargjøre programvareoppdateringen som vertssystemet ikke har kapasitet til å utføre konsekvent på tvers av versjoner.

Sikkerhet for perifer prosessor i Macer

Alle moderne datasystemer har mange innebygde perifere prosessorer som utfører oppgaver knyttet til nettverk, grafikk, strømstyring og annet. Disse perifere prosessorene har ofte kun ett formål og er mye mindre kraftige enn hovedprosessoren. Innebygde perifere prosessorer som ikke implementerer tilstrekkelig sikkerhet, blir dermed et mål for angripere som søker enda enklere mål å utnytte for varig infisering av operativsystemet. Ved å infisere firmwaren til en perifer prosessor kan en angriper angripe programvare på hovedprosessoren eller fange opp sensitive data direkte (for eksempel kan en Ethernet-enhet se innholdet i pakker som ikke er kryptert).

Der det er mulig, jobber Apple for å redusere antallet perifere prosessorer som trengs, og for å unngå designer som krever firmware. Når separate prosessorer med sin egen firmware er nødvendig, iverksettes tiltak for å bidra til å sikre at en angriper ikke kan lykkes på den prosessoren. Det kan være å verifisere prosessoren på én av to måter:

- kjøre prosessoren slik at den laster ned verifisert firmware fra hovedprosessoren ved oppstart
- sikre at den perifere prosessoren implementerer sin egen sikre oppstartssekvens der den verifiserer sin egen firmware hver gang Macen starter

Apple samarbeider med leverandører for å revidere implementeringer og forbedre design slik at følgende egenskaper kan inkluderes:

- krav om minimum krypteringsstyrke
- krav om sterk tilbakekalling av firmware med kjente feil
- deaktivering av feilsøkingsgrensesnitt
- signering av firmwaren med kryptografiske nøkler som er lagret i Apple-kontrollerte maskinvaresikkerhetsmoduler (HSM-er)

I de senere år har Apple samarbeidet med enkelte eksterne leverandører om implementering av «Image4»-datastrukturene, verifiseringskoden og signeringsinfrastrukturen som brukes av Apple Silicon.

Når verken lagringsfri drift eller lagring pluss sikker oppstart er en mulighet, tilsier designen at firmwareoppdateringer signeres og verifiseres kryptografisk før den varige lagringen kan oppdateres.

Rosetta 2 på Macer med Apple Silicon

Macer med Apple Silicon kan kjøre kode kompilert for x86_64-instruksjonssettet ved hjelp av en oversettingsmekanisme som kalles *Rosetta 2*. Det tilbys to typer oversetting: Just-in-time og Ahead-of-time.

Just-in-time-oversetting

I oversettingsprosessen Just-in-time (JIT) identifiseres et x86_64-Mach-objekt tidlig i diskfilens kjøringsbane. Når disse bildene forekommer, overfører kjernen kontroll til en spesiell Rosetta-oversettelsesstubb i stedet for den dynamiske lenkeredigereren, `dyld(1)`. Oversettelsesstubben oversetter deretter x86_64-sider når diskfilen kjører. Denne oversettelsen finner kun sted i prosessen. Kjernen verifiserer fortsatt kodehashene av hver x86_64-side mot kodesignaturen tilknyttet binærfilen. I tilfeller med manglende hashsamsvar håndhever kjernen repareringsregelsettet som er egnet for den prosessen.

Ahead-of-time-oversetting

I oversettingsprosessen Ahead-of-time (AOT) leses x86_64-binærfiler fra lagring på tidspunkter som systemet vurderer som optimale for responsivitet av den koden. De oversatte artefaktene skrives til lagring som en spesiell type Mach-objektfil. Den filen ligner på en kjørbar diskfil, men markeres for å indikere at det er det oversatte produktet til en annen diskfil.

I denne modellen avleder AOT-artefakten all identitetsinformasjon fra den originale kjørbare x86_64-diskfilen. For å håndheve denne bindingen signerer en privilegert brukerområdeenhet oversettelsesartefakten ved hjelp av en enhetsspesifikk nøkkel som administreres av Secure Enclave. Denne nøkkelen frigis kun til den privilegerte brukerområdeenheten, som identifiseres som dette ved hjelp av en begrenset rettighet. Kodekatalogen opprettet for oversettelsesartefakten inkluderer kodekataloghashen av den originale kjørbare x86_64-diskfilen. Signaturen på selve oversettelsesartefakten kalles *tilleggssignaturen*.

AOT-prosessen begynner likt som JIT-prosessen, der kjernen overfører kontroll til Rosetta-kjøringen i stedet for til den dynamiske lenkeredigereren `dyld(1)`. Rosetta-kjøringen sender imidlertid deretter en IPC-spørring til Rosetta-systemtjenesten, som spør om det finnes en AOT-oversettelse tilgjengelig for den gjeldende kjørbare diskfilen. Hvis det finnes, gir Rosetta-tjenesten en referanse til denne oversettelsen og så tilordnes den i prosessen og kjøres. Under kjøring håndhever kjernen kodekataloghashene av oversettelsesartefakten som autentiseres av signaturen rotfestet i den enhetsspesifikke signeringsnøkkelen. Kodekataloghashene av den originale x86_64-diskfilen er ikke involvert i denne prosessen.

Oversatte artefakter lagres i et datahvelv som ikke er tilgjengelig under kjøring av noen enheter, unntatt Rosetta-tjenesten. Rosetta-tjenesten administrerer tilgang til bufferen

ved å distribuere skrivebeskyttede fildeskriptorer til individuelle oversettelsesartefakter, noe som begrenser tilgang til AOT-artefaktbufferen. Denne tjenestens IPC og tilhørende fotavtrykk er med vilje svært innskrenket for å begrense angrepsflaten.

Hvis kodekataloghashene av den originale x86_64-diskfilen ikke samsvarer med den som er kodet inn i signaturen til AOT-oversettelsesartefakten, tilsvarer dette en ugyldig kodesignatur og det blir tatt egnede håndhevingstiltak.

Hvis en ekstern prosess spør kjernen om rettighetene eller andre kodeidentitetsegenskaper til en AOT-oversatt kjørbar fil, returneres identitetsegenskapene til den originale x86_64-diskfilen til den.

Innhold i statiske godkjenningsbuffere

macOS 11 eller nyere leveres med «fete» Mach-binærfiler som inneholder biter av x86_64- og arm64-datamaskinkode. På Macer med Apple Silicon kan brukeren velge å kjøre x86_64-biten av en systembinærfil via Rosetta-prosessen, for eksempel for å laste et programtillegg som ikke har en opprinnelig arm64-variant. For å støtte denne tilnærmingen inneholder den statiske godkjenningsbufferen som leveres med macOS, generelt sett, tre kodekataloghasher per Mach-objektfil:

- kodekataloghash av arm64-biten
- kodekataloghash av x86_64-biten
- kodekataloghash av AOT-oversettelsen av x86_64-biten

Rosetta AOT-oversettelsesprosedyren er deterministisk og reproduserer identisk utdata for hvilken som helst gitt inndata, uten hensyn til når oversettelsen ble utført eller på hvilken enhet den ble utført.

Under bygging av macOS kjøres hver Mach-objektfil gjennom Rosetta AOT-oversettelsesprosessen som er forbundet med versjonen av macOS som bygges, og den resulterende kodekataloghashen registreres i godkjenningsbufferen. De faktisk oversatte produktene leveres ikke med operativsystemet av effektivitetsårsaker og kan rekonstitueres på etterspørsel når brukeren ber om dem.

Når en x86_64-diskfil kjøres på en Mac med Apple Silicon, hvis den diskfilens kodekataloghash er i den statiske godkjenningsbufferen, forventes det at den resulterende kodekataloghashen av AOT-artefaktet *også* er i den statiske godkjenningsbufferen. Slike produkter signeres ikke av den enhetsspesifikke nøkkelen siden signeringsautoriteten er rotfestet i Apples sikre oppstartssekvens.

Usignert x86_64-kode

Macer med Apple Silicon tillater ikke at standard arm64-kode kjøres uten en tilknyttet kodesignatur. Denne signaturen kan være så enkel som en «ad hoc»-kodesignatur (jf. `codesign(1)`), som ikke innehar en faktisk identitet fra den hemmelige halvdelen av et asymmetrisk nøkkelpar (det er rett og slett en ikke-godkjent måling av binærfilen).

Oversatt x86_64-kode får lov til å kjøre gjennom Rosetta uten noen som helst signaturinformasjon på grunn av binær kompatibilitet. Det overføres ingen spesifikk identitet til denne koden gjennom den enhetsspesifikke Secure Enclave-signeringsprosedyren, og den kjøres med akkurat samme begrensninger som standard usignert kode som kjøres på Intel-baserte Macer.

Beskyttelser for direkte minnetilgang for Macer

For å oppnå høye overføringshastigheter på høyhastighetsgrensesnitt som PCIe, FireWire, Thunderbolt og USB, må datamaskiner ha støtte for direkte minnetilgang (DMA) fra eksterne enheter. Det vil si at de må kunne lese og skrive til RAM uten å kontinuerlig involvere prosessoren. Siden 2012 har Macer implementert en lang rekke teknologier for å beskytte mot angrep via DMA, noe som har ført til det beste og mest omfattende settet med DMA-beskyttelser på noen PC.

Beskyttelser for direkte minnetilgang for Macer med Apple Silicon

Apples System on Chip (SoC) inneholder en IOMMU (Input/Output Memory Management Unit) for hver DMA-agent i systemet, inkludert PCIe og Thunderbolt-porter. Siden hver IOMMU har sitt eget sett med adresseoversettingstabeller for å oversette DMA-forespørsler, har eksterne enheter som er koblet til via PCIe eller Thunderbolt, kun tilgang til deler av minnet som eksplisitt har blitt tilordnet for deres bruk. Eksterne enheter har ikke tilgang til minne som tilhører andre deler av systemet, som kjernen eller firmware, eller minne som er tilordnet andre eksterne enheter. Hvis en IOMMU oppdager at en ekstern enhet forsøker å få tilgang til minne som ikke er tilordnet for bruk av den eksterne enheten, utløser den en kjernepanikk.

Beskyttelser for direkte minnetilgang for Intel-basert Macer

Intel-baserte Macer med Intel Virtualization Technology for Directed I/O (VT-d) initialiserer IOMMU, noe som aktiverer DMA-omadressering og omadressering av avbrudd svært tidlig i oppstartsprosessen, for å redusere diverse klasser med sikkerhetssårbarheter. Apples IOMMU-maskinvare begynner operasjonen med et default-deny-regelsett. Straks systemet slås på, begynner den automatisk å blokkere DMA-forespørsler fra eksterne enheter. Etter at IOMMU-ene initialiseres av programvare begynner de å tillate DMA-forespørsler fra eksterne enheter til minneområder som har blitt eksplisitt tilordnet for deres bruk.

Merk: Omadressering av avbrudd for PCIe er ikke nødvendig på Macer med Apple Silicon siden hver IOMMU håndterer MSI-er for sine egne eksterne enheter.

Fra og med macOS 11 kjører også alle Macer med Apple T2-sikkerhetsbrikke UEFI-drivere som støtter DMA i et begrenset ring 3-miljø når disse driverne brukes sammen med eksterne enheter. Denne egenskapen bidrar til å redusere sikkerhetssårbarheter som kan oppstå når en ondsinnet enhet samhandler med en UEFI-driver på en uventet måte ved oppstart. Den reduserer særlig innvirkningen av sårbarheter i en drivers håndtering av DMA-buffere.

Kjerneutvidelser i macOS

Fra og med macOS 11, hvis kjerneutvidelser fra tredjeparter aktiveres, kan de ikke lastes inn i kjernen ved behov. I stedet slås de sammen i en *Auxiliary Kernel Collection (AuxKC)* som lastes under oppstartsprosessen. For Macer med Apple Silicon signeres målingen av AuxKC inn i LocalPolicy, mens for tidligere maskinvare befinner AuxKC seg på datavolumet. Ombygging av AuxKC krever brukerens godkjenning og at macOS må startes på nytt for å laste endringene inn i kjernen, og det krever at sikker oppstart konfigureres til Redusert sikkerhet.

Viktig: Kjerneutvidelser anbefales ikke lenger for macOS. Kjerneutvidelser utsetter integriteten og påliteligheten til operativsystemet for risiko, og Apple anbefaler at brukere velger løsninger som ikke krever utvidelse av kjernen.

Kjerneutvidelser på Macer med Apple Silicon

Kjerneutvidelser må eksplisitt aktiveres for Macer med Apple Silicon ved å holde inne av/på-knappen ved oppstart for å gå inn i One True Recovery-modus (1TR) og deretter nedgradere til Redusert sikkerhet og merke av i ruten for å aktivere kjerneutvidelser. Denne handlingen krever også at det skrives inn et administratorpassord for å godkjenne nedgraderingen. Kombinasjonen av 1TR og passordkravet gjør det vanskelig for en som kun angriper programvare fra innsiden av macOS, å legge inn kjerneutvidelser i macOS, som de deretter kan utnytte for å oppnå kjernerettigheter.

Etter at en bruker godkjenner lasting av kjerneutvidelser, brukes den ovennevnte flyten for brukergodkjent lasting av kjerneutvidelse til å godkjenne installasjonen av kjerneutvidelser. Godkjenningen som brukes for den ovennevnte flyten, brukes også til å fange opp en SHA384-hash av den brukerautoriserte kjerneutvidelseslisten (UAKL) i LocalPolicy. Kjernestyringsdaemonen (`kmd`) er deretter ansvarlig for å validere kun de kjerneutvidelsene som finnes i UAKL, for inkludering inn i AuxKC.

- Hvis System Integrity Protection (SIP) aktiveres, verifiseres signaturen til hver kjerneutvidelse før de inkluderes i AuxKC.
- Hvis SIP deaktiveres, håndheves ikke kjerneutvidelsessignaturen.

Denne tilnærmingen muliggjør Middels sikkerhet-flyter der utviklere eller brukere som ikke er en del av Apple Developer-programmet, tester kjerneutvidelser før de signeres.

Når AuxKC er opprettet, sendes den tilhørende målingen til Secure Enclave for å bli signert og inkludert i en Image4-datastruktur som kan evalueres av iBoot ved oppstart. Det genereres også en kjerneutvidelseskvittering som en del av AuxKC-konstruksjonen. Denne kvitteringen inneholder listen over kjerneutvidelser som faktisk ble inkludert i AuxKC, fordi settet kunne være et undersett av UAKL hvis det forekom forbudte kjerneutvidelser. En SHA384-hash av AuxKC Image4-datastrukturen og kjerneutvidelseskvitteringen inkluderes i LocalPolicy. AuxKC Image4-hash brukes for ekstra verifisering av iBoot ved oppstart for å bidra til å sikre at det ikke er mulig å starte opp en eldre Secure Enclave-signert AuxKC Image4-fil med en nyere LocalPolicy. Kjerneutvidelseskvitteringen brukes av undersystemer som Apple Pay for å bestemme om det for øyeblikket er lastet noen kjerneutvidelser som kan påvirke integriteten til macOS. Hvis det er det, kan Apple Pay-funksjoner deaktiveres.

Alternativer til kjerneutvidelser (macOS 10.15 eller nyere)

I macOS 10.15 kan utviklere utvide funksjonaliteten til macOS ved å installere og administrere systemutvidelser som kjører i brukerområdet i stedet for på kjernenivået. Ved å kjøre i brukerområdet, forbedrer systemutvidelser stabiliteten og sikkerheten til macOS. Selv om kjerneutvidelser i utgangspunktet har full tilgang til hele operativsystemet, tilordnes utvidelser som kjører i brukerområdet, kun de rettighetene som er nødvendige for utføring av de angitte funksjonene.

Utviklere kan bruke rammeverk som DriverKit, EndpointSecurity og NetworkExtension til å skrive USB- og grensesnittdrivere, endepunktsikkerhetsverktøy (som forhindring av tap av

data eller andre endepunktagenter) og VPN- og nettverksverktøy helt uten å måtte skrive kjerneutvidelser. Tredjepartssikkerhetsagenter bør kun brukes hvis de bruker disse API-ene eller har et robust veikart for en overgang til dem og bort fra kjerneutvidelser.

Brukergodkjent lasting av kjerneutvidelse

For å forbedre sikkerheten kreves brukergodkjenning for å laste inn kjerneutvidelser som installeres med eller etter installering av macOS 10.13. Denne prosessen kalles *brukergodkjent innlasting av kjerneutvidelser*. Administratorrettigheter kreves for å godkjenne en kjerneutvidelse. Kjerneutvidelser krever ikke godkjenning hvis de:

- ble installert på en Mac når den kjører macOS 10.12 eller eldre
- erstatter tidligere godkjente utvidelser
- kan lastes inn uten brukergodkjenning ved hjelp av `spctl`-kommandolinjeverktøyet som er tilgjengelig når en Mac ble startet fra recoveryOS
- kan lastes inn med en MDM-konfigurasjon

  Fra macOS 10.13.2 kan brukere bruke MDM til å spesifisere en liste med kjerneutvidelser som lastes inn uten brukergodkjenning. Dette valget krever en Mac som kjører macOS 10.13.2 som er registrert i MDM via Apple School Manager, Apple Business Manager eller brukergodkjent MDM-registrering.

Option ROM-sikkerhet i macOS

Merk: Option ROM-er støttes ikke på Macer med Apple Silicon.

Option ROM-sikkerhet på Macer med Apple T2-sikkerhetsbrikke

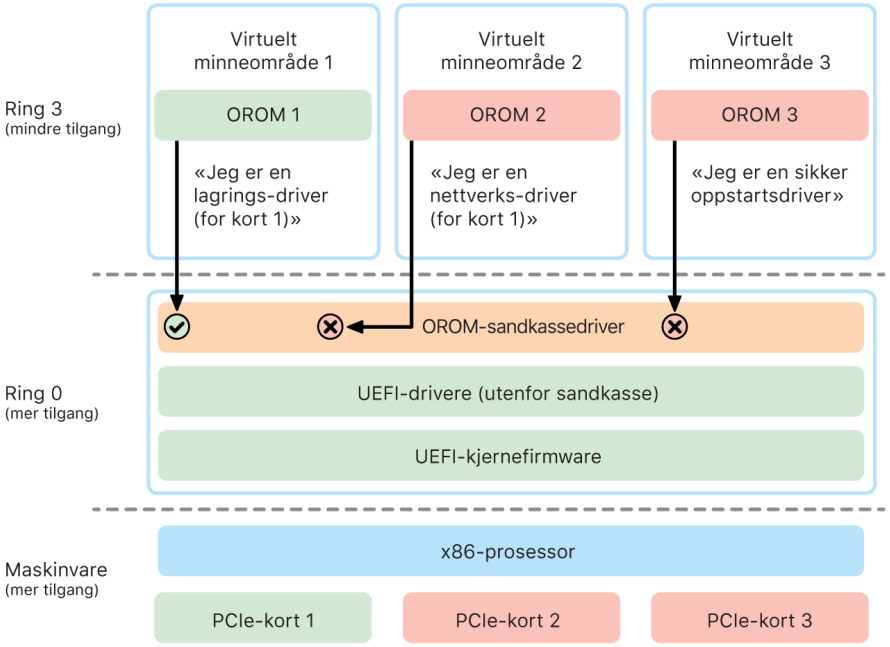
Både Thunderbolt- og PCIe-enheter kan ha en «Option ROM (OROM)» fysisk knyttet til enheten. (Dette er vanligvis ikke en ekte ROM, men er i stedet en skrivbar brikke som lagrer firmware.) På UEFI-baserte systemer er den firmwaren vanligvis en UEFI-driver, som leses inn av UEFI-firmwaren og kjøres. Den kjørte koden skal initialisere og konfigurere maskinvaren den ble hentet fra, slik at maskinvaren kan brukes av resten av firmwaren. Denne funksjonaliteten kreves slik at spesialisert tredjepartsmaskinvare kan laste inn og fungere under de tidligste oppstartsfasene, for eksempel for å starte fra eksterne RAID-oppsett.

OROM-er er imidlertid generelt skrivbare og en eventuell angriper som overskriver OROM-en til en legitim ekstern enhet, kjører kode tidlig i oppstartsprosessen og kan manipulere kjøringsmiljøet og krenke integriteten til programvare som lastes senere. På samme måte, hvis angriperen introduserer sin egen ondsinnede enhet i systemet, kan de også kjøre ondsinnet kode.

I macOS 10.12.3 ble adferden til Macer som ble solgt etter 2011, endret til å ikke utføre OROM-er som standard på det tidspunktet Macen startet, med mindre en spesiell tastaturkombinasjon ble trykket ned. Denne tastaturkombinasjonen beskyttet mot at ondsinnede OROM-er utilsiktet ble introdusert i macOS-oppstartssekvensen. Standardadferden til Firmwarepassordverktøy ble også endret slik at når brukeren angir et firmwarepassord, kan ikke OROM-er utføres selv hvis tastaturkombinasjonen ble brukt. Dette beskyttet mot at en fysisk tilstedeværende angriper med hensikt introduserte en ondsinnet OROM. For brukere som fortsatt må kjøre OROM-er mens de har angitt et firmwarepassord, kan et ikke-standard-valg konfigureres ved hjelp av `firmwarepasswd`-kommandolinjeverktøyet i macOS.

Sikkerhet for OROM-sandkasse

I macOS 10.15 ble UEFI-firmware oppdatert for å inneholde en mekanisme for sandkasseteknologi for OROM-er og fjerne rettigheter fra dem. UEFI-firmware kjører vanligvis all kode, inkludert OROM-er, på det høyeste CPU-rettighetsnivået, som kalles ring 0, og bruker ett delt virtuelt minne-område for all kode og data. Ring 0 er rettighetsnivået der macOS-kjernen kjører, mens det lavere rettighetsnivået, ring 3, er der programmer kjører. OROM-sandkassen fjernet rettigheter for OROM-er ved å benytte separasjon av virtuelt minne på samme måte som for kjernen, og deretter få OROM-ene til å kjøre i ring 3.



Option ROM (OROM)-sandkasseteknologi.

Videre begrenser sandkassen ytterligere både grensesnittene som OROM-ene kan kontakte (omtrent på samme måte som systemoppkallfiltrering i kjerner), og enhetstypen en OROM kan registreres som (omtrent på samme måte som godkjenning av programmer.) Fordelen med denne designen er at ondsinnede OROM-er ikke lenger kan skrive direkte noe sted innen ring 0-minne. I stedet begrenses de til et svært smalt og veldefinert sandkassegrensesnitt. Dette begrensede grensesnittet reduserer angrepsflaten betydelig, og tvinger angripere til å først slippe ut av sandkassen og eskalere rettigheter.

UEFI-firmwaresikkerhet for Intel-baserte Macer

Oversikt

Macer med Intel-basert prosessor har siden 2006 brukt en Intel-firmware basert på Extensible Firmware Interface (EFI) Development Kit (EDK) versjon 1 eller versjon 2. EDK2-basert kode oppfyller kravene i Unified Extensible Firmware Interface (UEFI)-spesifikasjonen. I denne delen refererer vi til Intel-firmwaren som *UEFI-firmware*. UEFI-firmwaren var den første koden som ble kjørt på Intel-brikken.

For Intel-baserte Macer uten Apple T2-sikkerhetsbrikken befinner «root of trust» for UEFI-firmwaren seg på brikken der firmwaren oppbevares. UEFI-firmwareoppdateringer signeres digitalt av Apple og verifiseres av firmwaren før lagringen oppdateres. For å bidra til å hindre tilbakerullingsangrep må oppdateringer alltid ha en versjon som er nyere enn den eksisterende. Imidlertid kunne en angriper med fysisk tilgang til Macen potensielt bruke maskinvare for å koble til firmwarelagringsbrikken og oppdatere brikken med ondsinnet innhold. Likeledes, hvis sårbarheter oppdages i den tidlige oppstartsprosessen til UEFI-firmwaren (før den skrivebegrenser lagringsbrikken), kunne dette også føre til varig infisering av UEFI-firmwaren. Dette er en maskinvarearkitekturbegrensning som er vanlig i de fleste Intel-baserte PC-er, og som er til stede i alle Intel-baserte Macer uten T2-brikken.

For å hjelpe til med å forhindre fysiske angrep som endrer UEFI-firmware, fikk Macer ny arkitektur for å rotfeste godkjenningen av UEFI-firmware i T2-brikken. På disse Macene er «root of trust» for UEFI-firmwaren spesifikt T2-firmwaren, som beskrevet i Oppstartsprosess for Intel-baserte Macer.

Underkomponent i Intel Management Engine (ME)

En underkomponent som lagres inne i UEFI-firmwaren, er *Intel Management Engine* (ME)-firmwaren. ME, som er en separat prosessor og et undersystem i Intel-brikker, kan brukes til lyd- og videoopphavsrettsbeskyttelse på Macer som kun har Intel-baserte grafikkort. For å redusere denne underkomponentens angrepsflate kjører Intel-baserte Macer en tilpasset ME-firmware der flesteparten av komponentene er fjernet. Siden den resulterende Mac ME-firmwaren er mindre enn det vanlige minimumsbygget som Intel tilbyr, er mange komponenter som har vært mål for offentlige angrep fra sikkerhetsforskere, ikke lenger til stede.

System Management Mode (SMM)

Intel-prosessorer har en spesiell utføringsmodus som er separat fra normal drift. Den kalles *System Management Mode (SMM)* og ble opprinnelig introdusert for å håndtere tidssensitive operasjoner som strømstyring. Historisk sett har imidlertid Macer brukt en egen mikrokontroller som kalles *System Management Controller (SMC)* til å utføre slike handlinger. SMC-en er ikke lenger en separat mikrokontroller, men er integrert i T2-brikken.

Systemsikkerhet for watchOS

Apple Watch bruker mange av de samme maskinvarebaserte plattformsikkerhetsfunksjonene som iOS og iPadOS bruker. For eksempel vil Apple Watch:

- gjennomføre sikker oppstart og sikre programvareoppdateringer
- ivareta operativsystemintegritet
- bidra til å beskytte data – både på enheten og ved kommunikasjon med en sammenkoblet iPhone eller internett

Støttede teknologier omfatter de som er oppført i Systemsikkerhet (for eksempel KIP, SKP og SCIP), samt databeskyttelse, nøkkelring og nettverksteknologier.

Oppdatere systemprogramvaren

Apple Watch kan konfigureres for en systemprogramvareoppdatering som gjennomføres samme natt. Hvis du vil vite mer om hvordan Apple Watch-koden lagres og brukes under oppdateringen, kan du lese mer om Nøkkeletuier.

Håndleddsensor

Hvis håndleddsensor er aktivert, vil enheten låses automatisk en stund etter at brukeren tar den av håndleddet. Hvis håndleddssensoren er deaktivert, viser Kontrollsenter et valg om å låse Apple Watch. Når Apple Watch er låst, kan Apple Pay kun brukes ved at brukeren oppgir koden på Apple Watch. Håndleddssensoren kan slås av ved hjelp av Apple Watch-appen på iPhone. Denne innstillingen kan også styres gjennom en MDM-løsning.

Aktiveringslås

Når Hvor er? er aktivert på iPhone, kan sammenkoblet Apple Watch bruke Aktiveringslås. Aktiveringslås gjør det vanskeligere å bruke eller selge en Apple Watch som er blitt borte eller stjålet. Aktiveringslås krever brukerens Apple-ID og passord for å fjerne sammenkobling, slette eller reaktivere en Apple Watch.

Sikker sammenkobling med iPhone

Apple Watch kan sammenkobles med kun én iPhone om gangen. Når sammenkoblingen med Apple Watch fjernes, kommuniserer iPhone instruksjoner om å slette alt innhold og alle data fra klokken.

Sammenkobling av Apple Watch med iPhone sikres med en out-of-band-prosess for å utveksle offentlige nøkler, etterfulgt av den delte hemmeligheten for Bluetooth Low Energy (BLE)-koblingen. Apple Watch viser et animasjonsmønster som fanges opp av kameraet på iPhone. Mønsteret inneholder en kodet hemmelighet som brukes til OOB-sammenkobling med BLE 4.1. Standard BLE-passordtilgang brukes som reservemetode for sammenkobling om nødvendig.

Etter at BLE-økten er etablert og kryptert ved hjelp av den høyeste sikkerhetsprotokollen som er tilgjengelig i Bluetooth Core-spesifikasjonen, utveksler iPhone og Apple Watch nøkler ved hjelp av enten:

- En prosess som er tilpasset fra Apple Identity Service (IDS) som beskrevet i Oversikt over iMessage-sikkerhet.

- En nøkkelutveksling ved bruk av IKEv2/IPsec. Den første nøkkelutvekslingen autentiseres ved hjelp av enten nøkkelen for Bluetooth-økten (for sammenkoblinger) eller IDS-nøklene (for oppdateringer av operativsystemet). Hver enhet genererer et tilfeldig og privat 256-bit-Ed25519-nøkkelpar, og i løpet av den første nøkkelutvekslingsprosessen utveksles de offentlige nøklene.

Merk: Mekanismen som brukes for nøkkelutveksling og kryptering varierer avhengig av operativsystemversjonene på iPhone og Apple Watch. iPhone-enheter som kjører iOS 13 eller nyere ved sammenkobling med en Apple Watch som kjører watchOS 6 eller nyere, bruker kun IKEv2/IPsec for nøkkelutveksling og kryptering.

Etter at nøkler har blitt utvekslet:

- Nøkkelen for Bluetooth-økten kastes, og all kommunikasjon mellom iPhone og Apple Watch krypteres ved hjelp av en av de ovennevnte metodene, der de krypterte Bluetooth-, Wi-Fi- og mobildatalenkene sørger for et sekundært krypteringslag.
- (Kun IKEv2/IPsec) Nøklene lagres i systemnøkkelringen og brukes til å autentisere fremtidige IKEv2/IPsec mellom enhetene. All videre kommunikasjon mellom disse enhetene krypteres og autentiseres ved hjelp av ChaCha20-Poly1305 (256-bit-nøkler).

Bluetooth Low Energy-enhetsadressen roteres med intervaller på 15 minutter for å redusere risikoen for at enheten spores lokalt hvis noen kringkaster en varig identifikator.

For å støtte apper som trenger strømmedata, gjøres kryptering tilgjengelig med metoder som er beskrevet i FaceTime-sikkerhet, og som bruker enten Apple Identity Service (IDS) som leveres av den sammenkoblede iPhonen eller en direkte internettforbindelse.

Apple Watch implementerer maskinvarekryptert lagring og klassebasert beskyttelse av filer og nøkkelringobjekter. Nøkkeletuier med tilgangskontroll for nøkkelringobjekter brukes også. Nøkler som brukes til å kommunisere mellom Apple Watch og iPhone, sikres også ved hjelp av klassebasert beskyttelse. Du finner mer informasjon om dette under Nøkkeletuier for databeskyttelse.

Automatisk opplåsing og Apple Watch

For å gjøre det enklere å bruke flere Apple-enheter kan enkelte enheter automatisk låse opp andre enheter i enkelte situasjoner. Automatisk opplåsing støtter tre scenarier:

- En Apple Watch kan låses opp av en iPhone.
- En Mac kan låses opp av en Apple Watch.
- En iPhone kan låses opp av en Apple Watch når det registreres en bruker med maske over munn og nese.

Alle tre scenarier er basert på det samme: en gjensidig autentisert STS-protokoll (Station-to-Station) med langvarige nøkler som utveksles når funksjonen aktiveres, og unike kortvarige øktnøkler som formidles ved hver forespørsel. Uavhengig av den underliggende kommunikasjonskanalen forhandles STS-tunnelen direkte mellom Secure Enclave i de to enhetene, og alt kryptografisk materiale holdes i det sikre domenet (bortsett fra Macer uten Secure Enclave – da termineres STS-tunnelen i kjernen).

Opplåsing

En fullstendig opplåsingssekvens kan deles inn i to faser. Først genererer enheten som låses opp («målet») en kryptografisk opplåsingshemmelighet og sender den til enheten som gjennomfører opplåsingen («igangsetteren»). Deretter gjennomfører igangsetteren opplåsingen ved hjelp av hemmeligheten som ble generert tidligere.

For å klargjøre automatisk opplåsing kobler enhetene seg til hverandre via en BLE-tilkobling. Deretter sendes en opplåsingsnøkkel på 32 bytes fra målenheten til igangsetteren via STS-tunnelen. Under den neste biometriske eller kodebaserte opplåsingen vil målenheten pakke den kodeavledede nøkkelen (PDK) med opplåsingshemmeligheten og fjerne opplåsingshemmeligheten fra minnet.

For å gjennomføre opplåsingen igangsetter enheten en ny BLE-tilkobling og bruker deretter peer-to-peer Wi-Fi til å anslå avstanden mellom enhetene på en sikker måte. Hvis enhetene er innenfor angitt rekkevidde og sikkerhetsreglene er oppfylt, vil igangsetteren sende opplåsingshemmeligheten til målet via STS-tunnelen. Deretter genererer målet en ny opplåsingshemmelighet på 32 bytes, og sender den tilbake til igangsetteren. Hvis opplåsingshemmeligheten som igangsetteren sender, dekrypterer opplåsingsoppføringen, låses målenheten opp, og PDK pakkes på nytt med en ny opplåsingshemmelighet. Til slutt fjernes den nye opplåsingshemmeligheten og PDK fra målets minne.

Sikkerhetsregler for automatisk opplåsing med Apple Watch

Apple Watch kan låses opp av en iPhone umiddelbart etter første oppstart uten at brukeren må angi koden på Apple Watch. For å gjøre dette brukes den tilfeldige opplåsingshemmeligheten (som genereres etter den første opplåsingssekvensen etter at funksjonen aktiveres) til å opprette en langvarig deponeringsoppføring, som lagres i Apple Watch-nøkkeletuiet. Den deponerte hemmeligheten arkiveres i iPhone-nøkkelringen og brukes som Bootstrap for en ny økt hver gang Apple Watch har blitt startet på nytt.

Sikkerhetsregler for automatisk opplåsing med iPhone

Det brukes ekstra sikkerhetsregler for automatisk opplåsing av iPhone med Apple Watch. Apple Watch kan ikke brukes i stedet for Face ID på iPhone til andre operasjoner, for eksempel Apple Pay eller godkjenninger i app. Når Apple Watch har låst opp en sammenkoblet iPhone, viser Apple Watch en varsling kombinert med et følbart varsel. Hvis brukeren trykker på knappen Lås iPhone i varslingen, sender Apple Watch en låsekommando til iPhone via BLE. Når iPhone mottar låsekommandoen, låser den og deaktiverer både Face ID og opplåsing ved hjelp av Apple Watch. Neste gang iPhone låses opp, må det gjøres med koden til iPhone.

For at en sammenkoblet iPhone skal kunne låses opp fra Apple Watch (når funksjonen er aktivert), må følgende kriterier være oppfylt:

- iPhone må ha blitt låst opp på en annen måte minst én gang etter at tilknyttet Apple Watch ble plassert på håndleddet og låst opp.
- Sensorer må kunne registrere at nese og munn er tildekket.
- Målt avstand må ikke overstige 2–3 meter.
- Apple Watch må ikke være i Leggetid-modus.
- Apple Watch eller iPhone må ha blitt låst opp nylig, eller Apple Watch må ha registrert fysisk bevegelse som indikerer at brukeren er aktiv (for eksempel at brukeren ikke sover).

- iPhone må ha blitt låst opp minst én gang de siste 6,5 timene.
- iPhone må være i en tilstand som lar Face ID låse opp enheten. (Du finner mer informasjon om dette under Touch ID, Face ID, koder og passord.)

Godkjenn i macOS med Apple Watch

Når Automatisk opplåsing med Apple Watch er aktivert, kan Apple Watch brukes i stedet for eller sammen med Touch ID for å godkjenne autoriserings- og autentiseringsforespørsler fra:

- macOS og Apple-programmer som krever autorisering
- tredjepartsprogrammer som krever autentisering
- arkiverte Safari-passord
- sikre notater

Sikker bruk av Wi-Fi, mobilnett, iCloud og Gmail

Når Apple Watch ikke er innenfor Bluetooth-rekkevidde, kan Wi-Fi eller mobildata brukes i stedet. Apple Watch kobler automatisk til Wi-Fi-nettverk som allerede er koblet til på den sammenkoblede iPhonen og der akkreditivene er blitt synkronisert til Apple Watch mens begge enhetene var innenfor rekkevidde. Denne autotilkoblingsadferden kan deretter konfigureres per nettverk i Wi-Fi-delen i Innstillinger-appen for Apple Watch. Wi-Fi-nettverk som aldri tidligere har vært koblet til fra noen av enhetene, kan kobles til manuelt i Wi-Fi-delen i Innstillinger-appen for Apple Watch.

Når Apple Watch og iPhone er utenfor rekkevidde, kobler Apple Watch direkte til iCloud- og Gmail-tjenere for å hente e-post, i motsetning til å synkronisere Mail-data med den sammenkoblede iPhonen over internett. For Gmail-kontoer må brukeren autentisere mot Google i Mail-delen i Apple Watch-appen på iPhone. OAuth-kjennetegnet som mottas av Google, sendes til Apple Watch i kryptert format over Apple Identity Service (IDS), slik at den kan brukes til å hente e-post. Dette OAuth-kjennetegnet brukes aldri for tilkobling med Gmail-tjeneren fra den sammenkoblede iPhonen.

Generering av tilfeldige tall

Kryptografiske pseudotilfeldige tallgeneratorer (CPRNG-er) er viktige byggesteiner for sikker programvare. For dette formålet leverer Apple en godkjent programvare-CPRNG som kjører i iOS-, iPadOS-, macOS-, tvOS- og watchOS-kjernene. Den er ansvarlig for å aggregere rå entropi fra systemet og levere sikre, tilfeldige tall til forbrukere i både kjernen og brukerområdet.

Entropikilder

Kjerne-CPRNG-en avledes av flere entropikilder under oppstart og over enhetens levetid. Disse inkluderer (avhengig av tilgjengelighet):

- Secure Enclaves maskinvare-TRNG
- Tidsbasert jitter innsamlet under oppstart
- Entropi innsamlet fra maskinvareavbrudd

- En seed-fil som brukes til å opprettholde entropi på tvers av oppstarter
- Tilfeldige Intel-instruksjoner – for eksempel RDSEED og RDRAND (kun på Intel-baserte Macer)

Kjerne-CPRNG-en

Kjerne-CPRNG-en er en Fortuna-avledet design som tar sikte på et 256-bit sikkerhetsnivå. Den leverer tilfeldige tall med høy kvalitet til sluttbrukere ved hjelp av følgende API-er:

- systemkallet `getentropy`(2)
- Den tilfeldige enheten (/dev/random)

Kjerne-CPRNG godtar brukerlevert entropi via skriving til den tilfeldige enheten.

Apple SRD-enhet (Security Research Device)

Apple SRD-enheten er en spesielt konfigurert iPhone som tillater sikkerhetsforskere å utføre forskning på iOS uten å måtte overvinne eller deaktivere plattformsikkerhetsfunksjonene på iPhone. Med denne enheten kan en forsker direkte installere innhold som kjører med plattformtilsvarende tillatelser, og dermed utføre forskning på en plattform som er svært lik produksjonsenhetene.

For å bidra til å sikre at brukerenhetene ikke påvirkes av SRD-enhetens regelsett for kjøring, implementeres regelsettendringene i en versjon av iBoot og i Boot Kernel Collection. Disse kan ikke startes på brukermaskinvare. Forsknings-iBoot ser etter en ny konfigurasjonstilstand og går inn i en «panic loop» hvis den kjører på ikke-forskningskonfigurert maskinvare.

Med cryptex-undersystemet kan en forsker laste en tilpasset godkjenningsbuffer og en diskfil som inneholder tilsvarende innhold. Det er implementert en rekke grundige forsvarstiltak som er utviklet for å sikre at dette undersystemet ikke tillater kjøring på brukerenheter:

- `launchd` laster ikke launchd-egenskapslisten til `cryptexd` hvis det ikke klarer å oppdage forskningskonfigurasjonen.
- `cryptexd` avbryter hvis det ikke oppdager forskningskonfigurasjonen.
- Rettigheten som gir `cryptexd` muligheten til å aktivere en diskfil, innfris kun av kjernebufferen på forskningsenheter. Den relevante kodebanen er ikke kompilert inn i kjernebufferen på offisielle enheter.
- Signeringstjeneren nekter å tilpasse en cryptex-diskfil for en enhet som ikke er på en eksplisitt tillatelsesliste.

For å respektere personvernet til sikkerhetsforskeren er det kun målingene (for eksempel hasher) fra de kjørbare filene og SRD-enhetens identifikatorer som sendes til Apple under tilpassing. Apple mottar ikke innholdet i cryptexen som lastes på enheten.

For å unngå at en ondsinnet part forsøker å utgi en forskningsenhet for å være en brukerenhet for å lure målet til å bruke den til hverdagsbruk, har SRD-enheten følgende forskjeller:

- SRD-enheten starter kun opp mens den lader. Dette kan være med en Lightning-kabel eller en Qi-kompatibel lader. Hvis enheten ikke lader under oppstart, går enheten inn i gjenopprettingsmodus. Hvis brukeren begynner å lade og starter enheten på nytt, starter den som normalt. Når XNU har startet, trenger ikke enheten å lade for å fortsatt kunne brukes.
- Ordene *Security Research Device* vises under Apple-logoen under start av iBoot.
- XNU-kjernen starter i detaljert modus.
- Det er gravert en melding på siden av enheten: «Property of Apple. Confidential and Proprietary. Call +1 877 595 1125.»

Følgende er ytterligere tiltak som implementeres i programvare som vises etter oppstart:

- Ordene *Security Research Device* vises under konfigurasjon av enheten.
- Ordene *Security Research Device* vises på låst skjerm og i Innstillinger-appen.

SRD-enheten tilbyr forskere følgende funksjoner som en brukerenhet ikke gjør:

- Direkte installering av kjørbar kode på enheten med vilkårlige rettigheter med samme tillatelsesnivå som komponenter i Apple-operativsystemet.
- Starte tjenester ved oppstart.
- Lagre innhold på tvers av omstarter.

Kryptering og databeskyttelse

Oversikt over kryptering og databeskyttelse

Den sikre oppstartssekvensen, systemsikkerhet og appsikkerhetsfunksjoner sørger alle for å verifisere at kun godkjente apper og kode kjører på en enhet. Apple-enheter har ytterligere krypteringsfunksjoner for å sikre brukerdata, selv når andre deler av sikkerhetsinfrastrukturen har blitt kompromittert (for eksempel hvis en enhet mistes eller kjører kode som ikke er godkjent). Alle disse funksjonene gir både brukere og IT-administratorer fordeler med beskyttelse av personopplysninger og bedriftsinformasjon og metoder for umiddelbar og fullstendig fjernsletting hvis enheten blir borte eller stjålet.

iOS- og iPadOS-enheter bruker en filkrypteringsmetode som kalles *databeskyttelse*, mens dataene på Intel-baserte Macer beskyttes med en volumkrypteringsteknologi som kalles *FileVault*. Macer med Apple Silicon bruker en hybridmodell som støtter databeskyttelse, med to forbehold: Det laveste sikkerhetsnivået (D) støttes ikke, og klasse C (standard) bruker faktisk en volumnøkkel og oppfører seg akkurat som FileVault på Intel-baserte Macer. I alle tilfeller er nøkkeladministreringshierarkiene forankret i den fysiske Secure Enclave-brikken, og en dedikert AES-motor støtter sanntidskryptering og bidrar til å sikre at krypteringsnøkler med lang levetid ikke eksponeres for kjerneoperativsystemet eller prosessoren (der de kan kompromitteres). (Intel-baserte Macer med T1 eller uten Secure Enclave bruker ikke dedikert silisium til å beskytte FileVault-krypteringsnøklene.)

I tillegg til å bruke databeskyttelse og FileVault til å bidra til å hindre uautorisert tilgang til data, bruker Apple *operativsystemkjerner* for beskyttelse og sikkerhet. Kjernen bruker tilgangskontroller for sandkasseteknologi for apper (som begrenser hvilke data en app har tilgang til) og en mekanisme som kalles *datahvelv* (som begrenser tilgangen til data i en app fra alle forespurte apper i stedet for å begrense anropene en app kan utføre).

Koder og passord

Koder i enheter som støtter databeskyttelse

Når brukeren angir en kode eller passord for enheten, aktiveres databeskyttelse automatisk. iOS og iPadOS støtter alfanumeriske koder på seks sifre, fire sifre og sifre med vilkårlig lengde. En kode eller et passord låser opp enheten og sørger for entropi for visse krypteringsnøkler. Det betyr at uvedkommende ikke kan få tilgang til dataene i spesifikke beskyttelsesklasser uten koden.

Koden eller passordet er integrert i enhetens UID, slik at brute-force-forsøk må skje på enheten som er under angrep. Et høyt iterasjonstall brukes til å gjøre hvert forsøk på

å tippe passordet tregere. Iterasjonstallet er kalibrert slik at ett forsøk tar om lag 80 millisekunder. Det betyr at det vil ta over fem og et halvt år å prøve alle kombinasjoner av alfanumeriske koder på seks tegn med små bokstaver og tall.

Jo sterkere koden er, jo sterkere blir krypteringsnøkkelen. Og ved å bruke Touch ID og Face ID kan brukeren opprette en mye sterkere kode enn det som ellers hadde vært praktisk å bruke. Den sterkere koden øker den effektive mengden entropi som beskytter krypteringsnøklene som brukes til databeskyttelse, uten at det går utover brukeropplevelsen når enheten skal låses opp mange ganger i løpet av dagen.

For å gjøre det enda mindre fristende å utføre brute-force-angrep er det en funksjon som øker tidsforsinkelsen hver gang det angis en ugyldig kode på låst skjerm.

Forsinkelser mellom kodeforsøk

Forsøk	Forsinkelse håndhevet
1–4	Ingen
5	1 minutt
6	5 minutter
7–8	15 minutter
9	1 time

Hvis Slett data er slått på (i Innstillinger > Touch ID og kode), slettes alt innhold og alle innstillinger etter at det er gjort 10 mislykkede forsøk etter hverandre. Flere etterfølgende forsøk med samme, uriktige kode teller ikke i forhold til grensen. Denne innstillingen er også tilgjengelig som administrative retningslinjer via en MDM-løsning som støtter denne funksjonen og Microsoft Exchange ActiveSync, og det kan angis en lavere grense.

På enheter med Secure Enclave iverksettes forsinkelsene av Secure Enclave. Hvis enheten startes på nytt når en tidsforsinkelse er iverksatt, gjelder forsinkelsen fortsatt, og tidtakingen begynner på nytt for gjeldende periode.

Spesifisere lengre koder

Hvis det oppgis et langt passord som kun består av tall, vises et numerisk tastatur på låst skjerm i stedet for hele tastaturet. Det kan være enklere å oppgi en lengre numerisk kode enn en kortere alfanumerisk kode, og det gir den samme sikkerheten.

Brukerne kan angi lengre alfanumeriske koder ved å velge Tilpasset alfanumerisk kode i Sikkerhetskodevalg i Innstillinger > Touch ID og kode eller Face ID og kode.

Forsinkelser mellom passordforsøk på macOS

For å bidra til å hindre brute-force-angrep tillates ikke mer enn 10 passordforsøk i påloggingsvinduet eller med måldiskmodus når Macen starter opp. Stigende tidsforsinkelser iverksettes etter et definert antall mislykkede forsøk. Forsinkelsene håndheves av Secure Enclave. Hvis Macen startes på nytt når en tidsforsinkelse er iverksatt, gjelder forsinkelsen fortsatt, og tidtakingen begynner på nytt for gjeldende periode.

For å bidra til å hindre skadelig programvare i å forårsake permanent datatap ved å forsøke å angripe brukerens passord, håndheves ikke disse grensene etter at brukeren har lykkes med å logge på Macen, men iverksettes på nytt etter omstart. Hvis de 10 forsøkene brukes opp, er 10 nye forsøk tilgjengelig etter oppstart i recoveryOS. Hvis de også brukes opp, er ytterligere 10 forsøk tilgjengelig for hver FileVault-gjenopprettingsmekanisme (iCloud-gjenoppretting, FileVault-gjenopprettingsnøkkel og institusjonsnøkkel), for maksimalt ytterligere 30 forsøk. Når disse forsøkene er brukt opp, behandler ikke Secure Enclave lenger noen forespørsler om å dekryptere volumet eller verifisere passordet, og dataene på stasjonen er ikke mulige å gjenopprette.

For å beskytte data i et bedriftsoppsett bør IT-avdelingen definere og håndheve FileVault-konfigurasjonsregelsett ved hjelp av MDM-løsning. Organisasjoner har flere muligheter for å administrere krypterte volumer, inkludert gjenopprettingsnøkler for institusjonen, personlige gjenopprettingsnøkler (som kan lagres med MDM for deponering) eller en kombinasjon av begge. Nøkkelrotering kan også angis som en regel i MDM.

Forsinkelser mellom passordforsøk på Macer med Apple Silicon og Macer med T2-brikken

Forsøk	Forsinkelse håndhevet
5	1 minutt
6	5 minutter
7	15 minutter
8	15 minutter
9	1 time
10	Deaktivert

I Macer med Apple T2-sikkerhetsbrikke har passordet en lignende funksjon med unntak av at nøkkelen som genereres, brukes til FileVault-kryptering i stedet for databeskyttelse. macOS tilbyr også flere alternativer for passordgjenoppretting:

- iCloud-gjenoppretting
- FileVault-gjenoppretting
- FileVault-institusjonsnøkkel

Databeskyttelse

Oversikt over databeskyttelse

Apple bruker en teknologi som kalles databeskyttelse for å beskytte data lagret i flashlagring på enhetene som har en Apple SoC, for eksempel iPhone, iPad, Apple Watch, Apple TV og Macer med Apple Silicon. Med databeskyttelse kan enheten svare på vanlige aktiviteter, som for eksempel innkommende telefonsamtaler, men teknologien gjør det også mulig med kryptering av brukerdata på høyt nivå. Enkelte systemapper (for eksempel Meldinger, Mail, Kalender, Kontakter, Bilder) og Helse-dataverdier bruker databeskyttelse som standard. Tredjepartsapper får denne beskyttelsen automatisk.

Implementering

Databeskyttelse implementeres ved å lage og administrere et hierarki av nøkler og er basert på maskinvarekrypteringen som er innebygd i Apple-enheter. Databeskyttelse styres ved at hver enkelt fil tilordnes en klasse, og tilgjengeligheten bestemmes av om klassenøklene er låst opp. Med APFS (Apple File System) kan filsystemet ytterligere dele inn nøklene på et per-utstrekning-grunnlag (der deler av en fil kan ha forskjellige nøkler).

Hver gang det opprettes en fil på datavolumet, lager databeskyttelsen en ny 256-bit-nøkkel (den «filspesifikke» nøkkelen) og gir den til den maskinvarebaserte AES-motoren, som bruker nøkkelen til å kryptere filen når den skrives til flashlagring. På A14- og M1-enheter bruker krypteringen AES-256 i XTS-modus der 256-bit per-fil-nøkkelen går gjennom en nøkkelavledingsfunksjon (NIST Special Publication 800-108) for å avlede en 256-bit-«tweak» og en 256-bit-kodenøkkel. Maskinvaregenerasjonene A9 til A13, S5 og S6 bruker AES-128 i XTS-modus der 256-bit per-fil-nøkkelen deles for å gi en 128-bit-«tweak» og en 128-bit-kodenøkkel.

På Macer med Apple Silicon bruker databeskyttelse som standard klasse C (se Databeskyttelsesklasser), men benytter en volumnøkkel i stedet for en filspesifikk nøkkel eller per-fil-nøkkel. Dette gjenskaper i realiteten sikkerhetsmodellen til FileVault for brukerdata. Brukerne må fortsatt velge FileVault for å kunne motta den fullstendige beskyttelsen med å integrere krypteringsnøkkelhierarkiet med passordet. Utviklere kan også velge en høyere beskyttelsesklasse som benytter en filspesifikk nøkkel eller per-fil-nøkkel.

Databeskyttelse i Apple-enheter

På Apple-enheter med databeskyttelse er hver fil beskyttet med en unik per-fil (eller filspesifikk) nøkkel. Nøkkelen som er pakket ved hjelp av nøkkelinnpakningsalgoritmen NIST AED, pakkes med én av flere klassenøkler, avhengig av hvordan det skal være mulig å få tilgang til filen. Den innpakkede filspesifikke nøkkelen oppbevares i filens metadata.

Enheter med APFS-formatet kan støtte kloning av filer (nullkostnadskopier som bruker kopier-ved-skriv-teknologi). Hvis en fil klones, får hver halvdel av klonen en ny nøkkel for å godta innkommende skriving slik at de nye dataene skrives til mediet med en ny nøkkel. Over tid kan filen bestå av forskjellige utstrekninger (eller fragmenter) som hver tilordnes forskjellige nøkler. Imidlertid beskyttes alle utstrekningene som utgjør en fil av samme klassenøkkel.

Når en fil åpnes, dekrypteres metadataene med filsystemnøkkelen, og den innpakkede filspesifikke nøkkelen og en notasjon om hvilken beskyttelsesklasse den tilhører, avdekkes. Den filspesifikke (eller utstrekningsspesifikke) nøkkelen pakkes ut med klassenøkkelen og gis til den maskinvarebaserte AES-motoren som dekrypterer filen når den leses fra flashlagring. All håndtering av innpakkede filnøkler skjer i Secure Enclave. Filnøkkelen eksponeres aldri for applikasjonsprosessoren. Ved oppstart blir Secure Enclave enig med AES-motoren om en kortvarig nøkkel. Når Secure Enclave pakker ut nøklene til en fil, pakkes nøklene på nytt med den kortvarige nøkkelen og sendes tilbake til applikasjonsprosessoren.

Metadataene i alle filene i datavolumfilsystemet er kryptert med en tilfeldig volumnøkkel som opprettes når operativsystemet først installeres eller når enheten slettes av en bruker. Denne nøkkelen krypteres og pakkes inn av en nøkkelinnpakningsnøkkel som kun Secure Enclave kjenner til, for langsiktig lagring. Nøkkelinnpakningsnøkkelen endres hver gang en bruker sletter enheten. På A9-SoC-er (og nyere) bruker Secure Enclave

entropi, støttet av anti-repetisjonssystemer, for å oppnå slettbarhet og for å beskytte nøkkelinnpakningsnøkkelen, blant andre ressurser. Du finner mer informasjon om dette under Sikker, ikke-flyktig lagring.

Akkurat som filspesifikke eller utstrekningsspesifikke nøkler, eksponeres aldri metadatanøkkelen til datavolumet direkte for applikasjonsprosessoren. Secure Enclave leverer en kortvarig, oppstartsspesifikk versjon i stedet. Når den lagres, blir den krypterte filsystemnøkkelen også innpakket av en «slettbar nøkkel» lagret i Effaceable Storage eller ved å bruke en medienøkkelinnpakningsnøkkel, beskyttet av Secure Enclaves anti-repetisjonsmekanisme. Denne nøkkelen tilbyr ikke ytterligere datakonfidensialitet. Den er i stedet utformet for å kunne slettes raskt ved behov (av brukeren med innstillingen «Slett alt innhold og alle innstillinger» eller av en bruker eller administrator med en kommando for fjernsletting fra en MDM-løsning, Exchange ActiveSync eller iCloud). Når nøkkelen slettes på denne måten, blir alle filer kryptografisk utilgjengelige.

Innholdet i en fil kan krypteres med en eller flere filspesifikke (eller utstrekningsspesifikke) nøkler, som pakkes med en klassenøkkel og lagres i filens metadata, som så krypteres med filsystemnøkkelen. Klassenøkkelen beskyttes av maskinvarens UID, og noen klasser beskyttes også av brukerens kode. Dette hierarkiet gir både fleksibilitet og ytelse. Det er for eksempel kun nødvendig å pakke den filspesifikke nøkkelen på nytt for å endre klassen til en fil, og ved kodeendringer er det kun klassenøkkelen som pakkes på nytt.

Databeskyttelsesklasser

Når det opprettes en ny fil på enheter som støtter databeskyttelse, tilordnes den en klasse av appen som oppretter den. Hver klasse bruker ulike retningslinjer for å avgjøre når data er tilgjengelig. De grunnleggende klassene og retningslinjene er beskrevet nedenfor. Apple Silicon-baserte Macer støtter ikke klasse D: «Ingen beskyttelse», og det etableres en sikkerhetsgrense ved på- og avlogging (ikke låsing eller opplåsing som på iPhone, iPad og iPod touch).

Klasse	Beskyttelsestype
Klasse A: Komplett beskyttelse	(NSFileProtectionComplete)
Klasse B: Beskyttet hvis de ikke er åpne	(NSFileProtectionCompleteUnlessOpen)
Klasse C: Beskyttet fram til første brukerautentisering *Merk:* macOS bruker en volumnøkkel til å gjenskape File Vault-beskyttelsesegenskaper.	(NSFileProtectionCompleteUntilFirstUserAuthentication)
Klasse D: Ingen beskyttelse *Merk:* Støttes ikke på macOS.	(NSFileProtectionNone)

Komplett beskyttelse

(NSFileProtectionComplete): Klassenøkkelen er beskyttet av en nøkkel avledet fra brukerkoden eller -passordet og enhetens UID. Kort tid etter at brukeren låser en enhet (10 sekunder hvis Krev passord-innstillingen er Umiddelbart), kastes den dekrypterte klassenøkkelen, som gjør alle dataene i denne klassen utilgjengelige før brukeren angir koden på nytt eller låser opp (logger på) enheten ved hjelp av Touch ID eller Face ID.

I macOS, kort tid etter at den siste brukeren er logget av, kastes den dekrypterte klassenøkkelen, som gjør alle dataene i denne klassen utilgjengelige frem til en bruker angir koden på nytt eller logger på enheten ved hjelp av Touch ID.

Beskyttet hvis de ikke er åpne

(NSFileProtectionCompleteUnlessOpen): Det er mulig at noen filer må skrives mens enheten er låst eller når brukeren er logget av. Et godt eksempel på dette er et e-postvedlegg som lastes ned i bakgrunnen. Dette er det mulig å få til ved hjelp av asymmetrisk elliptisk kurve-kryptering (ECDH over Curve25519). Den vanlige filspesifikke nøkkelen er beskyttet av en nøkkel som er avledet med One-Pass Diffie-Hellman Key Agreement som beskrevet i NIST SP 800-56A.

Den kortvarige offentlige nøkkelen for avtalen lagres sammen med den innpakkede filspesifikke nøkkelen. KDF er Concatenation Key Derivation Function (Approved Alternative 1) som beskrevet i 5.8.1 i NIST SP 800-56A. AlgorithmID utelates. PartyUInfo og PartyVInfo er henholdsvis den kortvarige og den statiske offentlige nøkkelen. SHA256 brukes som hashing-funksjon. Med det samme filen lukkes, slettes den filspesifikke nøkkelen fra minnet. For å åpne filen igjen lages den delte hemmeligheten på nytt ved hjelp av den private nøkkelen for klassen «Beskyttet hvis de ikke er åpne» og filens kortvarige offentlige nøkkel, som brukes til å pakke ut den filspesifikke nøkkelen som igjen brukes til å dekryptere filen.

I macOS er den private delen av NSFileProtectionCompleteUnlessOpen tilgjengelig så lenge brukerne på systemet er pålogget eller autentisert.

Beskyttet fram til første brukerautentisering

(NSFileProtectionCompleteUntilFirstUserAuthentication): Denne klassen oppfører seg på samme måte som «Komplett beskyttelse» med unntak av at den dekrypterte klassenøkkelen ikke fjernes fra minnet når enheten er låst eller brukeren er logget av. Beskyttelsen i denne klassen kan minne om volumkryptering for stasjonære enheter, og den beskytter data mot angrep som inkluderer omstart. Dette er standardklassen for alle appdata fra tredjeparter som ikke er tilordnet en annen databeskyttelsesklasse.

I macOS bruker denne klassen en volumnøkkel som er tilgjengelig så lenge volumet er aktivert, og oppfører seg akkurat som FileVault.

Ingen beskyttelse

(NSFileProtectionNone): Denne klassenøkkelen beskyttes kun med UID-en og oppbevares i Effaceable Storage. Fordi alle nøklene som kreves for å dekryptere filer i denne klassen lagres på enheten, er den eneste fordelen med krypteringen rask fjernsletting. Hvis en fil ikke har fått tilordnet en databeskyttelsesklasse, lagres den likevel i kryptert form. (Det samme gjelder alle data på en iOS- og iPadOS-enhet.)

Dette støttes ikke i macOS.

Merk: I macOS, for volumer som ikke tilsvarer et oppstartet operativsystem, er alle databeskyttelsesklasser tilgjengelige så lenge volumet er aktivert. Standard databeskyttelsesklasse er *NSFileProtectionCompleteUntilFirstUserAuthentication*. Funksjonaliteten med filspesifikke nøkler er tilgjengelig for både Rosetta 2 og innebygde apper.

Nøkkeletuier for databeskyttelse

Nøklene for både fil- og nøkkelringdatabeskyttelsesklasser samles inn og administreres i nøkkeletuier på iOS, iPadOS, watchOS og tvOS. Disse operativsystemene bruker følgende nøkkeletuier: bruker, enhet, sikkerhetskopiering, deponering og iCloud-sikkerhetskopiering.

Brukernøkkeletui

Brukernøkkeletuiet er der hvor de innpakkede klassenøklene som brukes i vanlig drift av enheten, er lagret. Når for eksempel en kode oppgis, lastes *NSFileProtectionComplete* inn fra brukernøkkeletuiet og pakkes ut. Den er en binær plist-fil (.plist) som oppbevares i No Protection-klassen.

For enheter med SoC-er eldre enn A9, krypteres .plist-filinnholdet med en nøkkel som oppbevares i Effaceable Storage. For å gi såkalt «forward security» til nøkkeletuiene blir denne nøkkelen slettet og generert på nytt hver gang en bruker endrer koden.

For enheter med A9 eller nyere SoC-er, inneholder .plist-filen en nøkkel som indikerer at nøkkeletuiet lagres i et skap som beskyttes av den Secure Enclave-kontrollerte anti-repetisjonsverdien (nonce).

Secure Enclave administrerer brukernøkkeletuiet og kan spørres om låsestatusen til en enhet. Den rapporterer at enheten kun låses opp hvis alle klassenøklene i brukernøkkeletuiet er tilgjengelige og utpakningen har vært vellykket.

Enhetsnøkkeletui

Enhetsnøkkeletuiet brukes til å lagre de pakkede klassenøklene som brukes for operasjoner som involverer enhetsspesifikke data. iPadOS-enheter som er konfigurert for delt bruk, trenger noen ganger tilgang til akkreditiver før en bruker har logget på. Derfor kreves et nøkkeletui som ikke er beskyttet av brukerens kode.

iOS og iPadOS støtter ikke kryptografisk separasjon av filsysteminnhold per-bruker, som betyr at systemet bruker klassenøkler fra enhetsnøkkeletuiet til å pakke per-fil-nøkler. Nøkkelringen bruker imidlertid klassenøklene fra brukernøkkeletuiet til å beskytte objekter i brukernøkkelringen. I iOS- og iPadOS-enheter som er konfigurert for bruk av en enkelt bruker (standardkonfigurasjonen), er enhetsnøkkeletuiet og brukernøkkeletuiet det samme og er beskyttet av brukerens kode.

Nøkkeletuiet for sikkerhetskopiering

Nøkkeletuiet for sikkerhetskopiering blir opprettet når Finder (macOS 10.15 eller nyere) eller iTunes (i macOS 10.14 eller eldre) oppretter en kryptert sikkerhetskopi og denne lagres på samme datamaskin som sikkerhetskopier av enheten. Det opprettes et nytt nøkkeletui med et nytt nøkkelsett, og de sikkerhetskopierte dataene krypteres på nytt til disse nye nøklene. Som forklart tidligere blir nøkkelringobjekter som er ikke-migrerende,

værende innpakket med den UID-avledede nøkkelen. Dermed kan de gjenopprettes til enheten som de opprinnelig ble sikkerhetskopiert fra, men det er umulig å få tilgang til dem på en annen enhet.

Nøkkeletuiet, som beskyttes av passordsettet, kjøres gjennom 10 millioner iterasjoner av nøkkelavledingsfunksjonen PBKDF2. Til tross for det store antallet iterasjoner, er det ingen tilknytninger til en bestemt enhet, og et brute-force-angrep som skjer parallelt mot mange datamaskiner kan derfor teoretisk forsøkes utført på nøkkeletuiet for sikkerhetskopiering. Denne trusselen kan reduseres med et sterkt nok passord.

Hvis en bruker velger å ikke kryptere sikkerhetskopien, krypteres ikke filene uansett hvilken databeskyttelsesklasse de har, men nøkkelringen er fortsatt beskyttet med en UID-avledet nøkkel. Dette er grunnen til at nøkkelringobjekter migrerer til en ny enhet kun hvis det er angitt et passord for sikkerhetskopiering.

Deponeringsnøkkeletuiet

Deponeringsnøkkeletuiet brukes til å synkronisere med Finder (macOS 10.15 eller nyere) eller iTunes (i macOS 10.14 eller eldre) via USB og MDM. Dette nøkkeletuiet gjør det mulig for Finder eller iTunes å sikkerhetskopiere og synkronisere uten å be brukeren om å oppgi koden, og det gjør det mulig for en MDM-løsning å fjernslette brukerens kode. Det lagres på datamaskinen som brukes til å synkronisere med Finder eller iTunes, eller på MDM-løsningen som fjernadministrerer enheten.

Deponeringsnøkkeletuiet forbedrer brukeropplevelsen under enhetssynkronisering, som kan kreve tilgang til alle dataklasser. Når en enhet som er låst med kode, kobles til Finder eller iTunes for første gang, bes brukeren om å oppgi koden. Enheten oppretter deretter et deponeringsnøkkeletui som inneholder nøklene i samme klasse som de som brukes på enheten, som beskyttes av en nøkkel som nettopp er blitt generert. Deponeringsnøkkeletuiet og nøkkelen som beskytter det, deles mellom enheten og verten eller tjeneren, og dataene lagres på enheten i klassen «Beskyttet til første brukerautentisering». Dette er grunnen til at koden på enheten må oppgis før brukeren tar sikkerhetskopi ved hjelp av Finder eller iTunes for første gang etter en omstart.

Hvis det er en over-the-air (OTA)-programvareoppdatering, blir brukeren bedt om å oppgi koden ved igangsetting av oppdateringen. Dette brukes til å opprette et kjennetegn for engangsopplåsing på en sikker måte. Det låser opp brukernøkkeletuiet etter oppdateringen. Dette kjennetegnet kan ikke genereres uten at brukerens kode oppgis, og eventuelle kjennetegn som er generert tidligere, blir ugyldiggjort hvis brukeren endrer koden på enheten.

Kjennetegn for engangsopplåsing er for installering av en programvareoppdatering som utføres enten i forgrunnen eller i bakgrunnen. De krypteres med en nøkkel som er avledet fra gjeldende verdi til en monoton teller i Secure Enclave, nøkkeletuiets UUID og UID-en til Secure Enclave.

På A9-SoC-er (og nyere) bruker ikke kjennetegnet for engangsopplåsing lenger tellere eller Effaceable Storage. Det er i stedet beskyttet av Secure Enclave-kontrollert anti-repetisjonsverdi (nonce).

Kjennetegnet for engangsopplåsing for programvareoppdateringer i forgrunnen utløper etter 20 minutter. I iOS 13 og iPadOS 13.1 eller nyere oppbevares kjennetegnet separat og beskyttes av Secure Enclave. Før iOS 13 ble dette kjennetegnet eksportert fra Secure

Enclave og skrevet til Effaceable Storage, eller det ble beskyttet av Secure Enclaves anti-repetisjonsmekanisme. En retningslinje-tidtaker økte telleren trinnvis hvis enheten ikke hadde startet på nytt i løpet av 20 minutter.

Programvareoppdateringer i bakgrunnen skjer når systemet oppdager at en oppdatering er tilgjengelig og når ett av følgende er sant:

- automatiske oppdateringer konfigureres i iOS 12 eller nyere
- brukeren velger Installer senere ved varsel om oppdateringen

Etter at brukeren angir koden, genereres et kjennetegn for engangsopplåsing som kan forbli gyldig i Secure Enclave i opptil 8 timer. Hvis oppdateringen ennå ikke har funnet sted, ødelegges dette kjennetegnet for engangsopplåsing på alle låser og gjenopprettes for hver etterfølgende opplåsing. Hver opplåsing starter 8-timersperioden på nytt. Etter 8 timer vil en retningslinje-tidtaker gjøre kjennetegnet for engangsopplåsing ugyldig.

Nøkkeletuiet for iCloud-sikkerhetskopiering

Nøkkeletuiet for iCloud-sikkerhetskopiering likner nøkkeletuiet for sikkerhetskopiering. Alle klassenøkler i dette nøkkeletuiet er asymmetriske (bruker Curve25519, i likhet med databeskyttelsesklassen Beskytt hvis de ikke er åpne). Et asymmetrisk nøkkeletui brukes også til sikkerhetskopien når det gjelder nøkkelringgjenoppretting for iCloud-nøkkelringen.

Beskytte nøkler i alternative oppstartsmoduser

Databeskyttelse er utviklet for å gi tilgang til brukerdata kun etter vellykket autentisering, og kun til den autoriserte brukeren. Databeskyttelsesklasser er utviklet for å støtte en rekke brukstilfeller, for eksempel muligheten til å lese og skrive noe data selv når en enhet er låst (men etter første opplåsing). Det finnes ytterligere tiltak for å beskytte tilgang til brukerdata under alternative oppstartsmoduser, for eksempel de som brukes for DFU-modus (Device Firmware Update), gjenopprettingsmodus, Apple-diagnostikk eller til og med under programvareoppdatering. Disse egenskapene er basert på en kombinasjon av maskinvare- og programvarefunksjoner, og de har blitt utvidet etter hvert som Apple-designet silisium har utviklet seg.

Funksjon	A10	A11, S3	A12, S4	A13, S5	A14, M1, S6
Gjenoppretting: alle databeskyttelsesklasser er beskyttet	✓	✓	✓	✓	✓
Alternative oppstarter av DFU-modus, Gjenoppretting og programvareoppdateringer: databeskyttelse i klasse A, B og C			✓	✓	✓

Secure Enclave AES-motoren er utstyrt med låsbare programvare-seed-biter. Når nøkler opprettes fra UID-en, inkluderes disse seed-bitene i nøkkelderivasjonsfunksjonen for å opprette ytterligere nøkkelhierarkier. Hvordan seed-biten brukes, varierer i henhold til SoC-en:

- Fra og med Apple A10- og S3-SoC-ene dedikeres en seed-bit for å skille nøkler som beskyttes av brukerens kode. Seed-biten settes for nøkler som krever brukerens kode (inkludert databeskyttelsesklasse A-, -klasse B- og -klasse C-nøkler), og fjernes for nøkler som ikke krever brukerens kode (inkludert filsystemmetadatanøkkelen og klasse D-nøkler).
- I iOS 13 eller nyere og iPadOS 13.1 eller nyere på enheter med en A10 eller nyere, gjøres alle data kryptografisk utilgjengelig når enheter startes i diagnostikkmodus. Dette oppnås ved å introdusere en ekstra seed-bit der innstillingen styrer muligheten til å få tilgang til medienøkkelen, som selv er nødvendig for å få tilgang til metadataene (og derfor innholdet i alle filer) på datavolumet som er kryptert med databeskyttelse. Denne beskyttelsen gjelder filer som er beskyttet i alle klasser (A, B, C og D), ikke kun de som krevde brukerens kode.
- På A12-SoC-er låser Secure Enclave oppstart-ROM kode-seed-biten hvis applikasjonsprosessoren er i DFU-modus (Device Firmware Upgrade) eller Gjenopprettingsmodus. Når kode-seed-biten er låst, tillates ingen operasjoner for å endre den. Dette er utviklet for å hindre tilgang til data som er beskyttet av brukerens kode.

Ved gjenoppretting av en enhet fra DFU-modus settes enheten tilbake i en kjent, fungerende tilstand der kun uendret, Apple-signert kode brukes. En enhet kan settes i DFU-modus manuelt.

Se følgende Apple-kundestøtteartikler om hvordan man setter en enhet i DFU-modus:

Enhet	Artikkel
iPhone, iPad, iPod touch	Hvis du har glemt sikkerhetskoden på iPhone eller iPhone er deaktivert
Apple TV	Gjenopprett Apple TV
Mac med Apple Silicon	Gjenoppliv eller gjenopprett en Mac med Apple Silicon

Beskytte brukerdata ved et angrep

Angripere som forsøker å trekke ut brukerdata, prøver ofte en rekke teknikker: trekke ut krypterte data til et annet medium for brute-force-angrep, eller manipulering av operativsystemversjonen eller på annen måte endre eller svekke sikkerhetsregelsettet på enheten for å legge til rette for angrep. Ved angrep på data på en enhet kreves det ofte kommunikasjon med enheten ved hjelp av fysiske grensesnitt, som Lightning eller USB. Apple-enheter har funksjoner som bidrar til å forhindre slike angrep.

Apple-enheter støtter en teknologi som kalles *Sealed Key Protection (SKP)*, som er utviklet for å sørge for at kryptografisk materiale blir utilgjengelig utenfor enheten, eller som brukes hvis operativsystemversjoner eller sikkerhetsinnstillinger manipuleres uten riktig brukergodkjenning. Denne funksjonen gis *ikke* av Secure Enclave, men støttes i stedet av maskinvareregistre som eksisterer på et lavere lag for å kunne gi nøklene et ekstra beskyttelseslag som er nødvendig for å dekryptere brukerdata uavhengig av Secure Enclave.

Merk: SKP er kun tilgjengelig på enheter med en Apple-designet SoC.

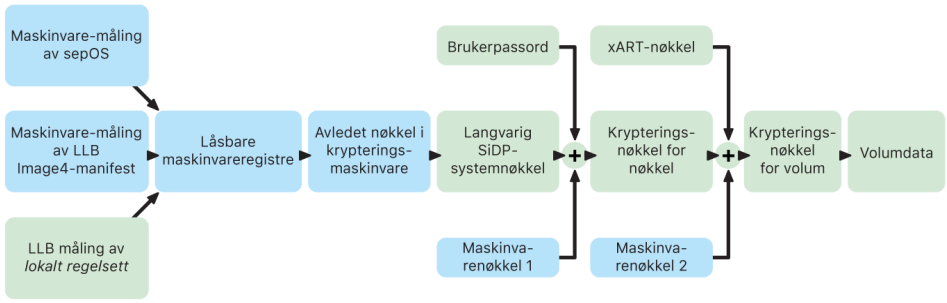
Funksjon	A10	A11, S3	A12, S4	A13, S5	A14, M1, S6
Sealed Key Protection	✓	✓	✓	✓	✓

iPhone og iPad kan også konfigureres til å kun aktivere dataforbindelser under forhold som mer sannsynlig indikerer at enheten fortsatt er under fysisk kontroll av den autoriserte eieren.

Sealed Key Protection (SKP)

På Apple-enheter som støtter databeskyttelse, beskyttes (eller forsegles) nøkkelkrypteringsnøkkelen (KEK) med målinger av programvaren på systemet, i tillegg til at den knyttes til UID-en som kun er tilgjengelig fra Secure Enclave. På Macer med Apple Silicon er beskyttelsen av KEK ytterligere styrket ved å ta med informasjon om sikkerhetsregelsett på systemet, fordi macOS støtter kritiske endringer i sikkerhetsregelsett (som deaktivering av sikker oppstart eller SIP) som ikke støttes på andre plattformer. På Macer med Apple Silicon omfatter denne beskyttelsen FileVault-nøkler, siden FileVault implementeres ved hjelp av databeskyttelse (klasse C).

Nøkkelen som er avledet fra integreringen av brukerens passord med den langvarige SKP-nøkkelen og maskinvarenøkkel 1 (UID-en fra Secure Enclave), kalles *kodeavledet nøkkel*. Denne nøkkelen brukes til å beskytte brukernøkkeletuiet (på alle støttede plattformer) og KEK (kun macOS) og gjør det mulig å bruke biometrisk eller automatisk opplåsing med andre enheter, for eksempel Apple Watch.



Sealed Key Protection-prosessen i en Mac med Apple Silicon.

Oppstartsovervåkingen for Secure Enclave fanger opp målingen av Secure Enclave-operativsystemet som lastes. Når applikasjonsprosessorens oppstart-ROM måler Image4-listen LLB, inneholder listen en måling av all annen systemparet firmware som også er lastet. LocalPolicy inneholder kjernesikkerhetskonfigurasjonene for macOS-et som lastes. LocalPolicy inneholder også `nsih`-feltet, som er en hash av Image4-listen til macOS. Image4-listen til macOS inneholder målinger av all macOS-paret firmware og macOS-kjerneoppstartsobjekter som Boot Kernel Collection eller rothashen av det signerte systemvolumet (SSV).

Hvis en angriper uventet får endret noen av de målte firmware-, programvare- eller sikkerhetskonfigurasjonkomponentene over, endrer den målingene lagret i maskinvareregistrene. Endringen av målingene fører til at den kryptomaskinvareavledede

rotnøkkelen for systemmåling (SMRK) avleder en annen verdi, noe som bryter forseglingen for nøkkelhierarkiet. Dette fører til at *enhetsnøkkelen for systemmåling (SMDK)* blir utilgjengelig, som igjen fører til at KEK, og dermed dataene, blir utilgjengelig.

Men når systemet ikke er under angrep, må det ta hensyn til legitime programvareoppdateringer som endrer firmwaremålingene og `nsih`-feltet i LocalPolicy for å peke på nye macOS-målinger. I andre systemer som forsøker å integrere firmwaremålinger, men som ikke har en kjent, god sannhetskilde, er brukeren nødt til å deaktivere sikkerheten, oppdatere firmware og deretter aktivere på nytt, slik at en ny målingsgrunnlinje kan fanges opp. Dette øker risikoen betydelig for at angriperen kan tukle med firmware under en programvareoppdatering. Systemet får hjelp av det faktum at Image4-listen inneholder alle de nødvendige målingene. Maskinvaren som dekrypterer SMDK med SMRK når målingene samsvarer under en normal oppstart, kan også kryptere SMDK til en foreslått fremtidig SMRK. Ved å spesifisere målingene som forventes etter en programvareoppdatering, kan maskinvaren kryptere en SMDK som er tilgjengelig i et gjeldende operativsystem, slik at det forblir tilgjengelig i et fremtidig operativsystem. På tilsvarende måte, når en kunde legitimt endrer sikkerhetsinnstillingene i LocalPolicy, må SMDK krypteres for fremtidig SMRK basert på målingen for LocalPolicy som LLB beregner ved neste omstart.

Sikker aktivering av dataforbindelser i iOS og iPadOS

Hvis det ikke nylig har blitt opprettet noen dataforbindelser på iOS- eller iPadOS-enheter, må brukerne bruke Touch ID, Face ID eller en kode for å aktivere dataforbindelser via et Lightning-, USB- eller Smart Connector-grensesnitt. Dette begrenser angrepsflaten mot fysisk tilkoblede enheter, for eksempel skadelige ladere, samtidig som det muliggjør bruk av annet tilbehør med rimelige tidsbegrensninger. Hvis det går mer enn én time fra iOS- eller iPadOS-enheten ble låst eller fra datatilkoblingen til et tilbehør ble frakoblet, vil ikke enheten tillate nye datatilkoblinger før enheten låses opp. I løpet av denne timesperioden tillates kun dataforbindelser fra tilbehør som tidligere har vært koblet til enheten mens den har vært ulåst. Dette tilbehøret blir husket i 30 dager etter tilkobling. Forsøk fra en ukjent tilbehørsenhet på å åpne en dataforbindelse i løpet av denne perioden vil føre til at alle forbindelser via Lightning, USB og Smart Connector vil bli deaktivert til enheten låses opp igjen. Denne timesperioden:

- bidrar til å sikre at brukere som ofte kobler til en Mac eller PC eller til tilbehør, eller med kabel til CarPlay, ikke må angi koden hver gang de kobler til enheten
- er nødvendig fordi økosystemet med tilbehør ikke tilbyr en kryptografisk, pålitelig måte å identifisere tilbehør på før en dataforbindelse etableres

I tillegg, hvis det har gått mer enn tre dager siden en datatilkobling ble etablert med et tilbehør, vil enheten avvise nye datatilkoblinger umiddelbart etter at den låses. Dette gjøres for å øke beskyttelsen for brukere som sjelden bruker slikt tilbehør. Datatilkoblinger via Lightning, USB og Smart Connector deaktiveres også hvis enheten er i en tilstand der en kode er nødvendig for å reaktivere biometrisk autentisering.

Brukeren kan velge å reaktivere alltid-på-dataforbindelser i Innstillinger (konfigurering av enkelte hjelpeenheter gjør dette automatisk).

Rollen til Apple File System

Apple File System (APFS) er et internt filsystem som ble designet med tanke på kryptering. APFS fungerer på alle Apples plattformer: iPhone, iPad, iPod touch, Mac, Apple TV og Apple Watch. Det er optimalisert for Flash/SSD-lagring og har sterk kryptering, implisitt deling av metadata, plassdeling, kloning for filer og kataloger, øyeblikksbilder, rask endring av katalogstørrelser, grunnelementer for sikker lagring og forbedrede grunnleggende filsystemfunksjoner, i tillegg til en unik implisitt deling-design som bruker I/O-forening for å levere maksimal ytelse, samtidig som datapåliteligheten sikres.

Deling av lagringsplass

APFS tilordner lagringsplass etter behov. Når én APFS-beholder har flere volumer, deles beholderens ledige plass og kan tilordnes hvilket som helst av enkeltvolumene etter behov. Hvert volum bruker kun en del av den overordnede beholderen, slik at tilgjengelig plass er den totale størrelsen til beholderen, minus plassen som brukes i alle volumer i beholderen.

Flere volumer

I macOS 10.15 eller nyere må en APFS-beholder som brukes til å starte Macen, inneholde minst fem volumer, hvorav de tre første er skjult for brukeren:

- *Føroppstartsvolum:* Inneholder data som er nødvendig for å starte hvert systemvolum i beholderen
- *VM-volum:* Brukes av macOS for oppbevaring av vekslefil
- *Gjenopprettingsvolum:* Inneholder recoveryOS
- *Systemvolum:* Inneholder følgende:
  - Alle filene som er nødvendige for å starte Macen
  - Alle programmene som installeres som standard av macOS (programmer som tidligere lå i /Programmer-mappen, ligger nå i /System/Programmer)

  Merk: Som standard kan ingen prosesser skrive til systemvolumet, ikke engang Apple-systemprosesser.
- *Datavolum:* Inneholder data som kan endres, for eksempel:
  - Alle data i brukerens mappe, inkludert bilder, musikk, videoer og dokumenter
  - Programmer brukeren installerte, inkludert AppleScript- og Automator-programmene
  - Tilpassede rammeverk og daemoner installert av brukeren, organisasjonen eller tredjepartsprogrammer
  - Andre plasseringer som er eid av og skrivbare for brukeren, som /Programmer, /Bibliotek, /Brukere, /Volumer, /usr/local, /private, /var og /tmp

Det opprettes et datavolum for hvert ekstra systemvolum. Føroppstartsvolumet, VM-volumet og gjenopprettingsvolumet er alle delt og ikke duplisert.

I macOS 11 tas det et øyeblikksbilde av systemvolumet. Operativsystemet starter fra et øyeblikksbilde av systemvolumet, ikke bare en aktivering med skrivebeskyttelse av det foranderlige systemvolumet.

I iOS og iPadOS er lagring delt i minst to APFS-volumer:

- Systemvolum
- Datavolum

Beskyttelse av nøkkelringdata

Mange apper må håndtere passord og andre korte, men sensitive data, som nøkler og påloggingskjennetegn. Nøkkelringen lagrer disse objektene på en sikker måte. De ulike Apple-operativsystemene bruker forskjellige mekanismer for å håndheve garantiene tilknyttet de ulike nøkkelringbeskyttelsesklassene. I macOS (inkludert på Macer med Apple Silicon) brukes ikke databeskyttelse direkte til å håndheve disse garantiene.

Oversikt

Nøkkelringobjekter krypteres med to ulike AES-256-GCM-nøkler: en tabellnøkkel (metadata) og en radspesifikk nøkkel (hemmelig nøkkel). Nøkkelringmetadata (alle andre verdier enn kSecValue) krypteres med metadatanøkkelen for hurtigsøk, og den hemmelige verdien (kSecValueData) krypteres med den hemmelige nøkkelen. Metadatanøkkelen beskyttes av Secure Enclave, men bufres i applikasjonsprosessoren for å muliggjøre hurtige nøkkelringforespørsler. Den hemmelige nøkkelen må alltid innom Secure Enclave.

Nøkkelringen implementeres som en SQLite-database som er lagret i filsystemet. Det er bare én database, og `securityd`-daemonen avgjør hvilke nøkkelringobjekter hver prosess eller app skal ha tilgang til. API-er for nøkkelringtilgang sender forespørsler til daemonen, som igjen spør om appens rettigheter når det gjelder «Keychain-access-groups», «application-identifier» og «application-group». I stedet for å begrense tilgangen til én enkelt prosess, gjør tilgangsgrupper det mulig for apper å dele nøkkelringobjekter.

Nøkkelringobjekter kan bare deles av apper fra samme utvikler. For å dele nøkkelringobjekter må tredjepartsapper bruke tilgangsgrupper med et prefiks de har fått tildelt av Apple-utviklerprogrammet i programgruppene deres. Prefikskravet og egenarten til applikasjonsgruppen håndheves gjennom kodesignering, klargjøringsprofiler og Apple-utviklerprogrammet.

Nøkkelringdata beskyttes med en klassestruktur tilsvarende den som brukes til beskyttelse av fildata. Disse klassene fungerer på tilsvarende måte som klassene for beskyttelse av fildata, men de bruker egne nøkler og funksjoner.

Tilgjengelighet	Beskyttelse av fildata	Beskyttelse av nøkkelringdata
Når ulåst	NSFileProtectionComplete	kSecAttrAccessibleWhenUnlocked
Når låst	NSFileProtectionCompleteUnlessOpen	Ikke tilgjengelig
Første gang enheten låses opp	NSFileProtectionCompleteUntilFirstUserAuthentication	kSecAttrAccessibleAfterFirstUnlock
Alltid	NSFileProtectionNone	kSecAttrAccessibleAlways
Kode aktivert	Ikke tilgjengelig	kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly

Apper som bruker tjenester for oppdatering i bakgrunnen, kan bruke *kSecAttrAccessibleAfterFirstUnlock* for nøkkelringobjekter som de trenger tilgang til under bakgrunnsoppdateringer.

Klassen *kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly* oppfører seg på samme måte som *kSecAttrAccessibleWhenUnlocked*, men den er kun tilgjengelig når enheten er konfigurert med at kode må oppgis. Denne klassen eksisterer kun i systemnøkkeletuiet. Den:

- synkroniseres ikke til iCloud-nøkkelring
- sikkerhetskopieres ikke
- inkluderes ikke i deponeringsnøkkeletuier

Hvis koden fjernes eller tilbakestilles, gjøres objektene ubrukelige ved at klassenøklene kastes.

Andre nøkkelringklasser har en «bare denne enheten»-motpart som alltid er beskyttet av UID-en ved kopiering fra enheten under en sikkerhetskopiering. Den er derfor ubrukelig hvis den gjenopprettes på en annen enhet. Apple har sørget for en nøye gjennomtenkt balanse mellom sikkerhet og brukervennlighet ved å velge nøkkelringklasser som avhenger av hvilken type informasjon som sikres og når iOS og iPadOS har bruk for den. Et VPN-sertifikat må for eksempel alltid være tilgjengelig slik at enheten har en kontinuerlig tilkobling. Det er imidlertid klassifisert som «ikke-migrerende» og kan derfor ikke flyttes til en annen enhet.

Beskyttelse av nøkkelringdataklasser

Klassebeskyttelsene som er oppført nedenfor, håndheves for nøkkelringobjekter.

Objekt	Tilgjengelig
Wi-Fi-passord	Første gang enheten låses opp
E-postkontoer	Første gang enheten låses opp
Microsoft Exchange ActiveSync-kontoer	Første gang enheten låses opp
VPN-passord	Første gang enheten låses opp
LDAP, CalDAV, CardDAV	Første gang enheten låses opp
Kjennetegn for kontoer på sosiale nettverk	Første gang enheten låses opp
Krypteringsnøkler for Handoff-annonsering	Første gang enheten låses opp
iCloud-kjennetegn	Første gang enheten låses opp
iMessage-nøkler	Første gang enheten låses opp
Passord for Hjemmedeling	Når ulåst
Safari-passord	Når ulåst
Bokmerker i Safari	Når ulåst
Finder-/iTunes-sikkerhetskopi	Når ulåst, ikke-migrerende
VPN-sertifikater	Alltid, ikke-migrerende
Bluetooth®-nøkler	Alltid, ikke-migrerende
Kjennetegn for Apples pushvarslingstjeneste (APNs)	Alltid, ikke-migrerende
Sertifikater og privat nøkkel for iCloud	Alltid, ikke-migrerende

Objekt	Tilgjengelig
Sertifikater og private nøkler installert av en konfigurasjonsprofil	Alltid, ikke-migrerende
PIN-kode for SIM-kort	Alltid, ikke-migrerende
«Hvor er?»-kjennetegn	Alltid
Talemeldinger	Alltid

Tilgangskontroll for nøkkelring

Nøkkelringer kan bruke tilgangskontrollister (ACL-er) til å angi retningslinjer for tilgang og autentiseringskrav. Objekter kan fastsette betingelser som krever brukerens tilstedeværelse ved å angi at det ikke er mulig å få tilgang til dem med mindre brukeren autentiserer ved hjelp av Touch ID eller Face ID eller ved å oppgi koden eller passordet på enheten. Tilgangen til objekter kan også begrenses ved å angi at Touch ID- eller Face ID-registreringen ikke har blitt endret etter at objektet ble lagt til. Denne begrensningen bidrar til å forhindre at uvedkommende legger til sitt eget fingeravtrykk for å få tilgang til et nøkkelringobjekt. ACL-er vurderes i Secure Enclave og frigis kun til kjernen hvis de spesifikke kravene som gjelder for dem, er oppfylt.

Nøkkelringarkitekturen i macOS

macOS gir også tilgang til nøkkelringen for å lagre brukernavn og passord, digitale identiteter, krypteringsnøkler og sikre notater på en praktisk og sikker måte. Man får tilgang til den ved å åpne programmet Nøkkelringtilgang i /Programmer/Verktøy/. Bruk av en nøkkelring fjerner behovet for å angi, eller til og med huske, akkreditivene for hver ressurs. En første standardnøkkelring opprettes for hver Mac-bruker, selv om brukere kan opprette andre nøkkelringer for spesifikke formål.

I tillegg til å benytte brukernøkkelringer, benytter macOS en rekke systemnivånøkkelringer som vedlikeholder autentiseringsressurser som ikke er brukerspesifikke, for eksempel nettverksakkreditiver og identiteter for infrastruktur for fellesnøkkel (PKI-identiteter). En av disse nøkkelringene, System Roots, er uforanderlig og lagrer sertifikater fra internett-PKI-rotsertifikatautoriteter (CA) for å støtte vanlige oppgaver som nettbank og e-handel. På samme måte kan brukeren rulle ut internt klargjorte CA-sertifikater til administrerte Macer for å bidra til validering av interne nettsider og tjenester.

FileVault

Volumkryptering med FileVault i macOS

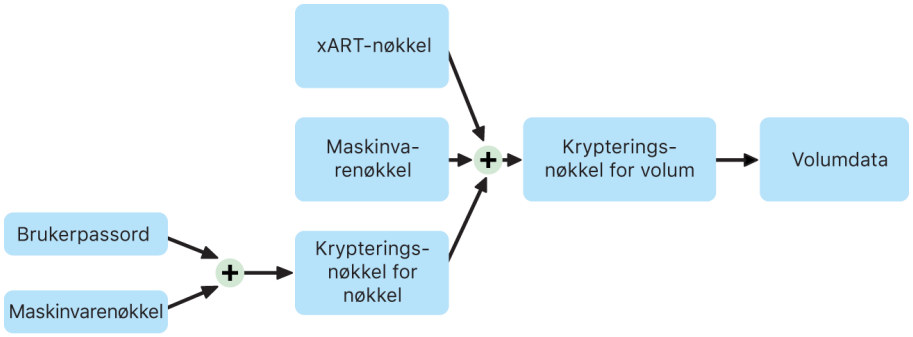
Macer tilbyr FileVault, en innebygd krypteringsfunksjon som sikrer alle lagrede data. FileVault bruker AES-XTS-datakrypteringsalgoritmen til å beskytte hele volumer på interne og uttakbare lagringsenheter.

FileVault på Macer med Apple Silicon implementeres ved å bruke databeskyttelsesklasse C med en volumnøkkel. På Macer med Apple T2-sikkerhetsbrikken og Macer med Apple Silicon benytter krypterte interne lagringsenheter direkte koblet til Secure Enclave seg av de tilhørende maskinvaresikkerhetsfunksjonene samt de til AES-motoren. Når en bruker har slått på FileVault på en Mac, må brukeren oppgi akkreditivene sine under oppstartsprosessen.

Intern lagring med FileVault slått på

Uten gyldige påloggingsakkreditiver eller en kryptografisk gjenopprettingsnøkkel forblir de interne APFS-volumene kryptert og beskyttes mot uautorisert tilgang, selv hvis den fysiske lagringsenheten fjernes og kobles til en annen datamaskin. I macOS 10.15 omfatter dette både systemvolumet og datavolumet. Fra og med macOS 11 beskyttes systemvolumet i kraft av SSV-funksjonen (signert systemvolum), men datavolumet beskyttes fortsatt av kryptering. Kryptering av det interne volumet på Macer med Apple Silicon og Macer med T2-brikken implementeres ved å konstruere og administrere et hierarki med nøkler og er basert på maskinvarekrypteringsteknologien som er innebygd i brikken. Hierarkiet med nøkler er designet for å oppnå fire mål samtidig:

- kreve brukerens passord for dekryptering
- beskytte systemet mot et brute-force-angrep direkte mot lagringsmedier som er fjernet fra Macen
- tilby en rask og sikker metode for å slette innhold via sletting av nødvendig kryptografisk materiale
- gjøre det mulig for brukere å endre passordet (og i sin tur de kryptografiske nøklene som brukes til å beskytte filene deres), uten å kreve ny kryptering av hele volumet



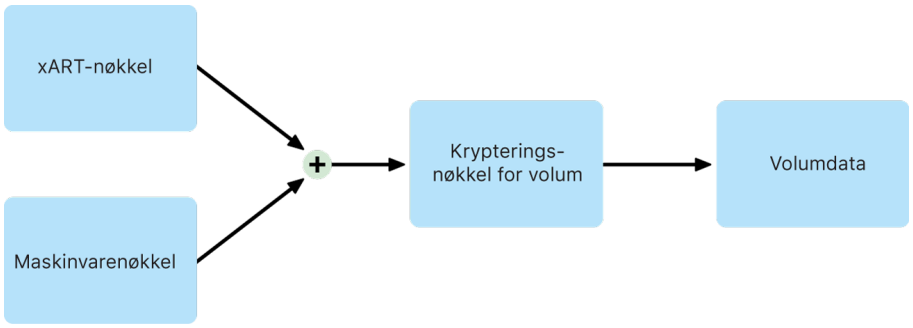
Les om kryptering av det interne volumet når FileVault er på i macOS.

På Macer med Apple Silicon og Macer med T2-brikken skjer all FileVault-nøkkelhåndtering i Secure Enclave. Krypteringsnøkler eksponeres aldri direkte for Intel-prosessoren. Alle APFS-volumer opprettes med en volumkrypteringsnøkkel som standard. Volum- og

metadatainnhold krypteres med denne volumkrypteringsnøkkelen, som er pakket med klassenøkkelen. Klassenøkkelen beskyttes av en kombinasjon av brukerens passord og maskinvare-UID-en når FileVault er på.

Intern lagring med FileVault slått av

Hvis FileVault ikke aktiveres på en Mac med Apple Silicon eller en Mac med T2-brikken under den første Oppsettassistent-prosessen, blir volumet fortsatt kryptert, men volumkrypteringsnøkkelen beskyttes kun av maskinvare-UID-en i Secure Enclave.



Les om kryptering av det interne volumet når FileVault er slått av i macOS.

Hvis FileVault slås på senere, en prosess som er umiddelbar fordi dataene allerede er kryptert, bidrar en anti-repetisjon-mekanisme til å forhindre at den gamle nøkkelen (kun basert på maskinvare-UID-en) brukes til å dekryptere volumet. Volumet beskyttes da av en kombinasjon av brukerpassordet og maskinvare-UID-en som tidligere beskrevet.

Sletting av FileVault-volumer

Når du sletter et volum, slettes volumkrypteringsnøkkelen sikkert av Secure Enclave. Dette bidrar til å hindre senere tilgang med denne nøkkelen, selv fra Secure Enclave. I tillegg pakkes alle volumkrypteringsnøkler med en medienøkkel. Medienøkkelen tilbyr ikke ytterligere datakonfidensialitet, men er i stedet designet for å muliggjøre rask og sikker sletting av data ettersom dekryptering er umulig uten den.

På Macer med Apple Silicon og Macer med T2-brikken er det garantert at medienøkkelen slettes av den støttede teknologien til Secure Enclave, for eksempel ved MDM-fjernkommandoer. Når medienøkkelen slettes på denne måten, blir volumet kryptografisk utilgjengelig.

Uttakbare lagringsenheter

Kryptering av uttakbare lagringsenheter bruker ikke sikkerhetsfunksjonene til Secure Enclave og krypteringen utføres på samme måte som på Intel-baserte Macer uten T2-brikken.

Administrering av FileVault i macOS

Ved bruk av sikkert kjennetegn

Apple File System (APFS) i macOS 10.13 eller nyere endrer hvordan FileVault-krypteringsnøkler genereres. I tidligere versjoner av macOS på CoreStorage-volumer ble nøklene som brukes i FileVault-krypteringsprosessen, opprettet når FileVault ble slått på av en bruker eller en organisasjon. I macOS på APFS-volumer genereres nøklene enten under oppretting av bruker, når den første brukerens passord angis, eller under første pålogging av en bruker på Macen. Denne implementeringen av krypteringsnøklene, når de genereres, og hvordan de lagres, er en del av en funksjon som kalles *sikkert kjennetegn*. Mer konkret er et sikkert kjennetegn en innpakket versjon av en nøkkelkrypteringsnøkkel (KEK) som beskyttes av en brukers passord.

Ved utrulling av FileVault på APFS kan brukeren fortsette å:

- bruke eksisterende verktøy og prosesser, for eksempel en personlig gjenopprettingsnøkkel (PRK) som kan oppbevares med en MDM-løsning for deponering
- opprette og bruke en gjenopprettingsnøkkel for institusjonen (IRK)
- utsette aktivering av FileVault til en bruker logger på eller av Macen

Når det første passordet angis for den aller første brukeren på Macen i macOS 11, får brukeren tildelt et sikkert kjennetegn. I enkelte arbeidsflyter er dette kanskje ikke ønskelig, siden tildeling av det første sikre kjennetegnet tidligere ville ha krevd pålogging av brukerkontoen. For å unngå at dette skjer, må `;DisabledTags;SecureToken` legges til den programmatisk opprettede brukerens `AuthenticationAuthority`-attributt før brukerens passord angis, som vist nedenfor:

```
sudo dscl . append /Users/<user name> AuthenticationAuthority
";DisabledTags;SecureToken"
```

Ved bruk av Bootstrap-kjennetegn

macOS 10.15 introduserte en ny funksjon, Bootstrap-kjennetegn, for å hjelpe til med å tildele et sikkert kjennetegn til både mobile kontoer og den valgfrie administratorkontoen («administrert administrator») opprettet via enhetsregistrering. I macOS 11 kan Bootstrap-kjennetegnet tildele et sikkert kjennetegn til en hvilken som helst bruker som logger på en Mac, inkludert lokale brukerkontoer. Bruk av Bootstrap-kjennetegn-funksjonen i macOS 10.15 eller nyere krever:

- Mac-registrering i MDM ved hjelp av Apple School Manager eller Apple Business Manager, som setter Macen under tilsyn
- MDM-leverandørstøtte

I macOS 10.15.4 eller nyere blir et Bootstrap-kjennetegn generert og deponert i MDM ved første pålogging av en bruker som har aktivert sikkert kjennetegn, hvis MDM-løsningen støtter funksjonen. Et Bootstrap-kjennetegn kan også genereres og deponeres i MDM ved hjelp av `profiles`-kommandolinjeverktøyet, hvis det er nødvendig.

I macOS 11 kan Bootstrap-kjennetegnet også brukes til mer enn kun å tildele sikkert kjennetegn til brukerkontoer. På Macer med Apple Silicon kan Bootstrap-kjennetegnet, hvis det er tilgjengelig, brukes til å autorisere installasjonen av både kjerneutvidelser og programvareoppdateringer når det administreres via MDM.

Slik beskytter Apple brukernes personopplysninger

Beskytte apptilgang til brukerdata

I tillegg til å kryptere lagrede data, hjelper Apple-enheter med å hindre apper fra å få tilgang til brukeres personlige opplysninger uten samtykke ved å bruke ulike teknologier, inkludert datahvelv. I Innstillinger i iOS og iPadOS eller Systemvalg i macOS kan brukere se hvilke apper de har godkjent for tilgang til bestemt informasjon, i tillegg til å godkjenne eller nekte godkjenning for tilgang i fremtiden. Tilgangen håndheves i følgende:

- *iOS, iPadOS og macOS:* Kalender, Kamera, Kontakter, Mikrofon, Bilder, Påminnelse, Talegjenkjenning
- *iOS og iPadOS:* Bluetooth, Hjem, medieinnhold, medieapper og Musikk, bevegelse og trening
- *iOS og watchOS:* Helse
- *macOS:* inndataovervåkning (for eksempel tastaturtrykk), ledetekst, skjermopptak (for eksempel statiske skjermbilder og video, systemvalg

I iOS 13.4 eller nyere og iPadOS 13.4 eller nyere blir dataene i alle tredjepartsapper automatisk beskyttet i et datahvelv. Datahvelv bidrar til å beskytte mot uautorisert tilgang til dataene, selv fra prosesser som ikke selv bruker sandkasseteknologi.

Hvis brukeren logger på iCloud, får apper i iOS og iPadOS tilgang til iCloud Drive som standard. Under iCloud i Innstillinger kan brukerne styre hva den enkelte appen har tilgang til. iOS og iPadOS har også restriksjoner som er utviklet for å forhindre dataflytting mellom apper og kontoer installert av en MDM-løsning og de installert av brukeren.

Beskytte tilgang til brukerens helsedata

HealthKit er et sentralt arkiv for helse- og treningsdata på iPhone og Apple Watch. HealthKit fungerer også direkte med helse- og treningsenheter, for eksempel kompatible Bluetooth Low Energy (BLE)-pulsmålere og koprosessoren for bevegelse som er bygd inn i mange iOS-enheter. HealthKits interaksjon med helse- og treningsapper, helseinstitusjoner og helse- og treningsenheter krever tillatelse fra brukeren. Disse dataene lagres i databeskyttelsesklassen Beskyttet hvis de ikke er åpne. Tilgang til dataene oppgis 10 minutter etter at enheten låses, og data blir tilgjengelig neste gang brukeren angir koden eller bruker Touch ID eller Face ID for å låse opp enheten.

Innsamling og lagring av helse- og treningsdata

HealthKit samler også inn og lagrer administrativ informasjon, for eksempel tilgangsrettigheter for apper, navn på enheter som er koblet til HealthKit, og planleggingsinformasjon som brukes til å starte apper når ny informasjon er tilgjengelig. Disse dataene er lagret i databeskyttelsesklassen «Beskyttet til første brukerautentisering». Midlertidige journalfiler lagrer helseinformasjon som genereres når enheten er låst, for eksempel når brukeren trener. Den lagres i databeskyttelsesklassen Beskyttet hvis de ikke er åpne. Når enheten låses opp, importeres de midlertidige journalfilene til de primære helsedatabasene og slettes deretter når importen er fullført.

Helsedata kan lagres i iCloud. Gjennomgående kryptering for Helsedata krever iOS 12 eller nyere og tofaktorautentisering. Uten dette vil brukerens data fortsatt være krypterte

både når de er lagret og under overføring, men vil ikke være gjennomgående krypterte. Når brukeren har slått på tofaktorautentisering og oppdaterer til iOS 12 eller nyere, vil helsedataene bli migrert til gjennomgående kryptering.

Hvis brukeren sikkerhetskopierer enheten ved hjelp av Finder (macOS 10.15 eller nyere) eller iTunes (i macOS 10.14 eller eldre), lagres helsedata kun hvis sikkerhetskopien er kryptert.

Kliniske helsejournaler

Brukere kan logge på støttede helsesystemer i Helse-appen for å få en kopi av sin kliniske helsejournal. Når en bruker kobles til et helsesystem, autentiseres brukeren ved hjelp av OAuth 2-klientakkreditiver. Etter tilkobling, lastes kliniske helsejournaldata direkte fra helseinstitusjonen ved hjelp av en TLS 1.3-beskyttet forbindelse. Når de er lastet ned, lagres kliniske helsejournaler sikkert sammen med andre helsedata.

Helsedataintegritet

Blant dataene som lagres i databasen er metadata for å spore opprinnelsen til de enkelte dataoppføringene. Metadataene inneholder en app-ID som viser hvilken app som lagret oppføringen. Dessuten kan et valgfritt metadataobjekt inneholde en kopi av oppføringen som er signert digitalt. Hensikten er å sørge for dataintegritet for oppføringer som er generert av en godkjent enhet. Formatet som brukes for den digitale signaturen, er Cryptographic Message Syntax (CMS) som er spesifisert i RFC 5652.

Tilgang til helsedata for tredjepartsapper

Tilgang til API-en for HealthKit styres av rettigheter, og apper må rette seg etter restriksjoner for bruken av dataene. Apper får for eksempel ikke lov til å benytte helsedata til reklameformål. Det er også et krav at apper opplyser brukerne om hvilke retningslinjer for personvern som gjelder for bruk av helsedata.

Appenes tilgang til helsedata styres av brukerens innstillinger for personvern. Brukerne blir bedt om å gi tilgang når apper ber om tilgang til helsedata, på samme måte som når de spør etter data fra Kontakter, Bilder og andre iOS-kilder. For helsedata gis imidlertid apper separat tilgang for lesing og skriving av data, og dessuten separat tilgang for de ulike typene helsedata. Brukerne kan se, og tilbakekalle, tillatelser de har gitt angående tilgang til helsedata i Innstillinger > Helse > Datatilgang og enheter.

Hvis appene får tillatelse til å skrive data, kan de også lese dataene de skriver. Hvis appene får tillatelse til å lese data, kan de lese data som er skrevet av alle kilder. Apper kan imidlertid ikke bestemme hvilken tilgang som gis til andre apper. Apper kan heller ikke vite sikkert om de har fått lesetilgang til helsedata. Når en app ikke har lesetilgang, returnerer alle forespørsler ingen data. Det er det samme svaret som ville blitt returnert om databasen var tom. Dette er utviklet for å forhindre at apper utleder brukerens helsestatus gjennom å fange opp hvilke typer data som brukeren registrerer.

Nødinfo for brukere

I Helse-appen kan brukerne fylle ut et Nødinfo-skjema med informasjon som kan være viktig i en nødssituasjon. Informasjonen oppgis og oppdateres manuelt og synkroniseres ikke med informasjonen i helsedatabasene.

Opplysningene under Nødinfo vises ved å trykke på Nødinfo-knappen på låst skjerm. Opplysningene lagres på enheten med databeskyttelsesklassen «Ingen beskyttelse», slik at det er mulig å få tilgang til dem uten å oppgi koden for enheten. Nødinfo er en valgfri funksjon der brukeren selv velger riktig balanse mellom sikkerhet og personvern. Disse dataene sikkerhetskopieres i iCloud-sikkerhetskopi i iOS 13 eller eldre. I iOS 14 synkroniseres Nødinfo mellom enheter ved hjelp av CloudKit og har samme krypteringsegenskaper som resten av helsedataene.

Digital signering og kryptering

Tilgangskontrollister

Nøkkelringdata fordeles og beskyttes med tilgangskontrollister (ACL-er). Som resultat kan ikke apper med andre identiteter få tilgang til akkreditiver som oppbevares av tredjepartsapper med mindre brukeren eksplisitt godkjenner dem. Denne beskyttelsen danner en mekanisme for sikring av autentiseringsakkreditiver i Apple-enheter på tvers av apper og tjenester i organisasjonen.

Mail

I Mail-appenkan brukere sende meldinger som er digitalt signert og kryptert. Mail gjenkjenner automatisk relevante RFC 5322-kompatible e-postadresseemner eller alternative subjektnavn på digitale signerings- og krypteringssertifikater på tilknyttede PIV-kjennemerker (Personal Identification Verification) i kompatible smartkort. Hvis en konfigurert e-postkonto stemmer overens med en e-postadresse på et digitalt signerings- eller krypteringssertifikat eller et tilknyttet PIV-kjennemerke, viser Mail automatisk signeringsknappen i verktøylinjen til et nytt melding-vindu. Hvis Mail har mottakerens krypteringssertifikat eller kan finne den i Microsoft Exchange sin globale adresseliste (GAL), vises et ulåst-symbol i ny melding-verktøylinjen. Et låst låsesymbol viser at meldingen vil bli sendt med kryptering til mottakerens offentlige nøkkel.

Meldingsbasert S/MIME

iOS, iPadOS og macOS støtter per-melding S/MIME. Dette betyr at S/MIME-brukere kan velge å alltid signere og kryptere meldinger som standard, eller å selektivt signere og kryptere enkeltmeldinger.

Identiteter som brukes med S/MIME, kan overføres til Apple-enheter med en konfigurasjonsprofil, en MDM-løsning, Simple Certificate Enrollment Protocol (SCEP) eller Microsoft Active Directory Certificate Authority.

Smartkort

macOS 10.12 eller nyere inkluderer innebygd støtte for PIV-kort. Disse kortene brukes i stort omfang i kommersielle og statlige organisasjoner for tofaktorautentisering, digital signering og kryptering.

Smartkort inkluderer en eller flere digitale identiteter som har et par med offentlige og private nøkler og et tilknyttet sertifikat. Opplåsing av et smartkort med det personlige identifikasjonsnummeret (PIN) gir tilgang til de private nøklene som brukes for

autentiserings-, krypterings- og signeringsoppgaver. Sertifikatet avgjør hva en nøkkel kan brukes til, hvilke attributter som er knyttet til den, og om den er validert (signert) av et CA-sertifikat (sertifikatautoritet).

Smartkort kan brukes til tofaktorautentisering. De to faktorene som er nødvendige for å låse opp et kort er «noe brukeren har» (kortet) og «noe brukeren vet» (PIN-koden). macOS 10.12 eller nyere har også innebygd støtte for smartkortpåloggingsvinduautentisering og klientsertifikatautentisering til nettsteder i Safari. Det støtter også Kerberos-autentisering ved hjelp av nøkkelpar (PKINIT) for Single Sign On for tjenester med Kerberos-støtte. Hvis du vil vite mer om smartkort og macOS, kan du lese Introduksjon til smartkortintegrering i *Håndbok for utrulling for Mac*.

Krypterte diskfiler

I macOS fungerer krypterte diskfiler som sikre beholdere der brukere kan lagre eller overføre sensitive dokumenter og andre filer. Krypterte diskfiler opprettes ved hjelp av Diskverktøy, som finnes i /Programmer/Verktøy/. Diskfiler kan krypteres med enten 128-bit eller 256-bit AES-kryptering. Ettersom en aktivert diskfil behandles som et lokalt volum koblet til en Mac, kan brukere kopiere, flytte og åpne filer og mapper som er lagret på den. På samme måte som FileVault, krypteres og dekrypteres innholdet på en diskfil i sanntid. Med krypterte diskfiler kan brukere trygt utveksle dokumenter, filer og mapper ved å arkivere en kryptert diskfil på uttakbare medier, sende den som et vedlegg i en e-postmelding eller oppbevare den på en ekstern tjener. Hvis du vil vite mer om krypterte diskfiler, kan du se Brukerveiledning for Diskverktøy.

Appsikkerhet

Oversikt over appsikkerhet

Apper er i dag noen av de mest kritiske elementene i en sikkerhetsarkitektur. Appene gir utrolige produktivitetsforbedringer for brukerne, men de kan også ha en negativ innvirkning på systemets sikkerhet, stabilitet og brukerdata hvis de ikke håndteres på riktig måte.

På grunn av dette tilbyr Apple flere lag med sikkerhet for å sikre at apper er frie for kjent skadelig programvare, og ikke har blitt tuklet med. Ytterligere beskyttelser håndhever at tilgang fra apper til brukerdata overvåkes nøye. Disse sikkerhetskontrollene gir en stabil og sikker plattform for apper og gjør det mulig for tusenvis av utviklere å lage hundretusenvis av apper for iOS, iPadOS og macOS uten at det påvirker systemets integritet. Og brukerne kan få tilgang til disse appene på Apple-enhetene uten å være redde for virus, skadelig programvare eller uautoriserte angrep.

På iPhone, iPad og iPod touch skaffes alle apper fra App Store, der alle appene bruker sandkasseteknologi, for å oppnå den strengeste kontrollen.

På Mac skaffes mange programmer fra App Store, men Mac-brukere laster også ned og bruker programmer fra internett. macOS har flere lag med kontroller for å støtte nedlasting fra internett. Først, som standard på macOS 10.15 og nyere, må alle Mac-programmer attesteres av Apple for å kunne starte. Dette kravet bidrar til at disse programmene er frie for kjent skadelig programvare uten at de må tilbys via App Store. I tillegg inkluderer macOS toppmoderne antivirusbeskyttelse for å blokkere, og hvis det er nødvendig, fjerne skadelig programvare.

Som en ekstra kontroll på tvers av plattformer, bidrar sandkasseteknologi til å beskytte brukerdata fra uautorisert tilgang fra programmer. Og i macOS er data i kritiske områder beskyttet. Det bidrar til å sikre at brukere beholder kontrollen over tilgang til filer i Skrivebord, Dokumenter, Nedlastinger og andre områder fra alle programmer, enten programmene som prøver å få tilgang, bruker sandkasseteknologi eller ikke.

Innebygd egenskap	Tredjepartsekvivalent
Tillegg ikke godkjent-liste, Safari-tillegg ikke godkjent-liste	Definisjoner for virus / skadelig programvare
Filkarantene	Definisjoner for virus / skadelig programvare
XProtect-/YARA-signaturer	Definisjoner for virus / skadelig programvare
MRT (Malware Removal Tool)	Endepunktbeskyttelse
Gatekeeper	Endepunktbeskyttelse; håndhever kodesignering av apper for å bidra til å sikre at kun godkjent programvare kjøres.

Innebygd egenskap	Tredjepartsekvivalent
eficheck (Kreves for Macer uten Apple T2-sikkerhetsbrikke)	Endepunktbeskyttelse; rootkit-detektering
Appbrannmur	Endepunktbeskyttelse; brannmur
Packet Filter (pf)	Brannmurløsninger
System Integrity Protection	Innebygd i macOS
Obligatoriske tilgangskontroller	Innebygd i macOS
Kext-ekskluderingsliste	Innebygd i macOS
Obligatorisk appkodesignering	Innebygd i macOS
Appattestering	Innebygd i macOS

Appsikkerhet i iOS og iPadOS

Oversikt over appsikkerhet for iOS og iPadOS

Til forskjell fra andre mobilplattformer, tillater ikke iOS og iPadOS at brukerne installerer usignerte apper som kan være skadelige, fra nettsider, eller at de kjører apper som ikke er godkjent. Når appen kjøres, kontrollerer kodesignaturen at alle kjørbare minnesider opprettes etter hvert som de lastes, for å bidra til å sikre at appen ikke er blitt endret etter at den ble installert eller siste gang den ble oppdatert.

Etter at en app er bekreftet å være fra en godkjent kilde, setter iOS og iPadOS i verk sikkerhetstiltak som er utarbeidet for å hindre at appen kompromitterer andre apper eller resten av systemet.

Kodesigneringsprosessen for apper i iOS og iPadOS

Obligatorisk kodesignering

Etter at iOS- eller iPadOS-kjernen har startet, kontrollerer den hvilke brukerprosesser og apper som kan kjøres. iOS og iPadOS krever at all kjørbar kode skal være signert med et sertifikat utstedt av Apple, for å bidra til å sikre at alle apper kommer fra en kjent og godkjent kilde og ikke er manipulert. Apper som følger med enheten, som Mail og Safari, er signert av Apple. Tredjepartsapper må også valideres og signeres med et sertifikat utstedt av Apple. Obligatorisk kodesignering utvider godkjenningskjedekonseptet fra operativsystemet til apper og bidrar til å hindre at tredjepartsapper laster inn usignerte koderessurser eller bruker selvmodifiserende kode.

Slik signerer utviklere appene sine

Utviklere kan signere appene via sertifikatvalidering (via Apple-utviklerprogrammet). De kan også bygge inn rammeverk i appene og få denne koden validert med et Apple-utstedt sertifikat (via en team-ID-streng).

- *Sertifikatvalidering:* For å kunne utvikle og installere apper på iOS- eller iPadOS-enheter må utviklere registrere seg hos Apple og bli med i Apple-utviklerprogrammet. Den virkelige identiteten til hver utvikler, uansett om dette er en person eller en bedrift, kontrolleres av Apple før sertifikatet utstedes. Sertifikatet gjør at utviklere kan signere apper og sende dem til App Store for distribusjon. Følgelig er alle apper i App Store sendt inn av en identifiserbar person eller organisasjon, og det virker forebyggende mot skadelige apper. Appene blir også gjennomgått av Apple for å bidra til å sikre at de fungerer mer eller mindre som beskrevet, og at de ikke inneholder åpenbare feil eller andre større problemer. I tillegg til teknologien vi allerede har nevnt, gir denne prosessen brukere tillit til kvaliteten på appene de kjøper.
- *Validering av kodesignatur:* iOS og iPadOS tillater at utviklerne bygger inn rammeverk i appene, som kan brukes av appen selv eller av tilleggene som er innebygd i appen. For å beskytte systemet og andre apper mot innlasting av tredjepartskode i adresseområdet, utfører systemet en validering av kodesignaturen til alle dynamiske bibliotek som en prosess oppretter koblinger til ved oppstart. Denne bekreftelsen gjennomføres ved hjelp av team-ID-en som trekkes ut fra et sertifikat utstedt av Apple. En team-ID er en alfanumerisk streng på 10 tegn, for eksempel 1A2B3C4D5F. Et program kan opprette en kobling til et hvilket som helst plattformbibliotek som leveres sammen med systemet, eller et hvilket som helst bibliotek med samme team-ID i kodesignaturen som hovedprogramfilen. Fordi de kjørbare filene som leveres som en del av systemet, ikke har en team-ID, kan de kun opprette en kobling til bibliotek som leveres med selve systemet.

Verifisering av bedriftsapper

Bedrifter har også muligheten til å skrive interne apper for bruk i organisasjonen og distribuere dem til de ansatte. Bedrifter og organisasjoner kan søke om opptak til Apple Developer Enterprise Program (ADEP) med et D-U-N-S-nummer. Apple godkjenner søkere etter å ha fått bekreftet identiteten deres og at de oppfyller kravene. Etter at en organisasjon blir medlem av ADEP, kan den registrere seg og få en klargjøringsprofil som tillater interne apper å kjøre på enheter som den godkjenner.

Brukerne må ha klargjøringsprofilen installert for å kjøre interne apper. Dette bidrar til å sikre at kun brukere som organisasjonen vil skal bruke appen, kan laste den inn på iOS- og iPadOS-enheten. Apper som er installert via MDM godkjennes underforstått fordi det allerede er etablert et forhold mellom organisasjonen og enheten. Hvis ikke må brukerne godkjenne appens klargjøringsprofil under Innstillinger. Organisasjoner kan hindre brukere i å godkjenne apper fra ukjente utviklere. Første gang en bedriftsapp startes, må enheten motta positiv bekreftelse fra Apple om at appen har tillatelse til å kjøre.

Sikkerhetsprosesser ved kjøring i iOS og iPadOS

Sandkasseteknologi

Alle tredjepartsapper begrenses av sandkasseteknologi, slik at de ikke får tilgang til filer som er lagret av andre apper, og ikke har lov til å gjøre endringer på enheten. Sandkasseteknologi er utviklet for å hindre at apper samler inn eller gjør endringer i informasjon som er lagret av andre apper. Alle apper har en unik hjemmekatalog for filene sine, som tilordnes tilfeldig når appen installeres. Hvis en tredjepartsapp trenger tilgang til informasjon som ikke tilhører den selv, får den det kun ved å bruke tjenester som uttrykkelig kommer fra iOS og iPadOS.

Systemfiler og -ressurser skjermes også mot brukerens apper. De fleste systemfilene og ressursene i iOS og iPadOS kjører som brukeren «mobile» uten rettigheter, og det samme gjør alle tredjepartsapper. Hele operativsystempartisjonen aktiveres med skrivebeskyttelse. Unødvendige verktøy, for eksempel tjenester for fjernpålogging, tas ikke med i systemprogramvaren, og API-er tillater ikke at apper utvider sine egne rettigheter for å endre andre apper eller iOS og iPadOS.

Bruk av rettigheter

Tilgangen som tredjepartsapper har til brukerinformasjon og til funksjoner som iCloud og tilleggsfunksjoner, kontrolleres ved hjelp av rettighetserklæringer. Rettigheter er nøkkelverdipar som er knyttet til en app, og som tillater autentisering utover kjøretidsfaktorer, for eksempel en UNIX-bruker-ID. Siden rettigheter er signert digitalt, kan de ikke endres. Rettigheter brukes i stor grad av systemapper og daemoner for å utføre spesifikke privilegerte operasjoner som ellers ville kreve at prosessen kjøres som «root»-bruker. Det reduserer muligheten for at noen lurer til seg tilgang gjennom en kompromittert systemapp eller -daemon.

Dessuten kan apper kun utføre bakgrunnsbehandling gjennom API-er levert av systemet. Dette gjør det mulig for apper å fortsette å fungere uten dårligere ytelse og uten at batteritiden påvirkes i stor grad.

Address Space Layout Randomization

ASLR (Address Space Layout Randomization) bidrar til å hindre uvedkommende i å utnytte feil i minnet. Innebygde apper bruker ASLR for å bidra til at alle minneområder genereres tilfeldig ved oppstart. Tilfeldig plassering av minneadressene til kjørbar kode, systembiblioteker og relatert programmering reduserer faren for mange former for utnyttelse. Et «return-to-libc»-angrep forsøker for eksempel å lure en enhet til å kjøre skadelig kode ved å manipulere minneadressene til stabler og systembiblioteker. Tilfeldig plassering av disse gjør det vanskeligere å utføre angrepet, særlig på flere enheter. Xcode, og utviklingsmiljøene til iOS og iPadOS, kompilerer automatisk tredjepartsapper med ASLR-støtte slått på.

Execute Never-funksjon

Ytterligere beskyttelse kommer fra iOS og iPadOS ved hjelp av ARMs XN-funksjon (Execute Never), som merker minnesider som ikke-kjørbare. Minnesider som er merket som både skrivbare og kjørbare, kan bare brukes av apper under nøye kontrollerte forhold: Kjernen kontrollerer at den finner den dynamiske kodesigneringsrettigheten som kun Apple har.

Selv da er det kun mulig å foreta ett enkelt `mmap`-anrop for å be om en kjørbar og skrivbar side, som blir gitt en tilfeldig adresse. Safari bruker denne funksjonaliteten til JavaScript JIT-kompilatoren.

Støtte for tillegg i iOS, iPadOS og macOS

iOS, iPadOS og macOS tillater at apper gir funksjonalitet til andre apper gjennom tillegg. Tillegg er signerte, kjørbare programfiler for spesielle formål som er pakket sammen med appen. Under installering gjenkjenner systemet automatisk tillegg og gjør dem tilgjengelige for andre apper ved hjelp av et gjenkjenningssystem.

Tilleggspunkter

Et systemområde som støtter tillegg, kalles et *tilleggspunkt*. Hvert tilleggspunkt leverer API-er og håndhever retningslinjer for dette området. Systemet avgjør hvilke tillegg som er tilgjengelige, basert på spesifikke gjenkjenningsregler for tilleggspunktet. Systemet starter automatisk tilleggsprosesser etter behov og administrerer levetiden deres. Rettigheter kan brukes til å begrense tilgang til tillegg til bestemte systemapper. En widget for I dag-visning vises for eksempel kun i Varslingssenter, og et delingstillegg er kun tilgjengelig i Deling-panelet. Eksempler på tilleggspunkter er I dag-widgeter, deling, handlinger, bilderedigering, filleverandør og tilpasset tastatur.

Slik kommuniserer tillegg

Tillegg kjører i sitt eget adresseområde. Kommunikasjon mellom tillegget og appen som den ble aktivert fra, skjer via kommunikasjon mellom prosesser med systemets rammeverk som mellomledd. De har ikke tilgang til hverandres filer eller minneområder. Tillegg er laget slik at de skal holdes atskilt fra hverandre, fra appen de tilhører og fra appene som bruker dem. Som alle andre tredjepartsapper begrenses de av sandkasseteknologi og har en beholder som er atskilt fra beholderen til appen de tilhører. De deler imidlertid samme tilgang til kontroller for personvern som beholderappen. Så hvis en bruker gir Kontakter tilgang til en app, videreføres denne tilgangen til tilleggene som er innebygd i appen, men ikke til tilleggene som aktiveres av appen.

Slik brukes tilpassede tastaturer

Tilpassede tastaturer er en spesiell type tillegg fordi de aktiveres av brukeren for hele systemet. Når et tastaturtillegg er aktivert, brukes det til alle tekstfelter unntatt kodeinntasting og eventuell visning av sikker tekst. For å begrense overføring av brukerdata er det standard at tilpassede tastaturer kjøres i en sandkasse med strenge restriksjoner som blokkerer tilgang til nettverket, til tjenester som utfører nettverksoperasjoner på vegne av en prosess, og til API-er som ville latt tillegget eksfiltrere inntastede data. Utviklere av egne tastaturer kan be om å få åpen tilgang for tillegget de lager. Da kan systemet kjøre tillegget i standardsandkassen etter å ha fått samtykke fra brukeren.

MDM og tillegg

For enheter som er registrert i en MDM-løsning, følger dokument- og tastaturtillegg Managed Open In-regler. MDM-løsningen kan for eksempel bidra til å hindre at brukere

eksporterer et dokument fra en administrert app til en ikke-administrert Document Provider, eller bidra til å hindre at de bruker et ikke-administrert tastatur med en administrert app. Dessuten kan apputviklere hindre at tastaturtillegg fra tredjeparter brukes i appen de lager.

Appbeskyttelse og appgrupper i iOS og iPadOS

Ta i bruk databeskyttelse i apper

iOS Software Development Kit (SDK) for iOS og iPadOS tilbyr en serie API-er som gjør det enkelt for tredjepartsutviklere og interne utviklere å bruke databeskyttelse og bidra til å ivareta det høyeste beskyttelsesnivået for appene sine. Databeskyttelse er tilgjengelig for API-er for filer og databaser, inkludert NSFileManager, CoreData, NSData og SQLite.

Mail-appdatabasen (inkludert vedlegg), administrerte bøker, Safari-bokmerker, startfiler for apper og stedsinformasjon lagres også gjennom kryptering med nøkler beskyttet av brukerens kode på enheten. Kalender (unntatt vedlegg), Kontakter, Påminnelser, Notater, Meldinger og Bilder implementerer databeskyttelsesrettigheten «Beskyttet til første brukerautentisering».

Brukerinstallerte apper som ikke velger en bestemt databeskyttelsesklasse, mottar som standard «Beskyttet til første brukerautentisering».

Bli med i en appgruppe

Apper og tillegg som eies av en gitt utviklerkonto, kan dele innhold når de er konfigurert til å være en del av en appgruppe. Det er opp til utvikleren å opprette de rette gruppene på Apple Developer Portal og inkludere det ønskede settet med apper og tillegg. Når apper er konfigurert til å være en del av en appgruppe, har de tilgang til følgende:

- en delt lagringsbeholder på volumet som blir værende på enheten så lenge minst én app fra gruppen er installert
- delte innstillinger
- delte nøkkelringobjekter

Apple Developer Portal bidrar til å sikre at appgruppe-ID-er (GID-er) er unike i hele appens økosystem.

Verifisere tilbehør i iOS og iPadOS

Lisensieringsprogrammet MFi (Made for iPhone, iPad og iPod touch) gir godkjente tilbehørsprodusenter tilgang til iAP-protokollen (iPod Accessories Protocol) og nødvendige maskinvarekomponentene for støtte.

Når MFi-tilbehør kommuniserer med en iOS- eller iPadOS-enhet via en Lightning- eller USB-C-tilkobling eller via Bluetooth, ber enheten tilbehøret om å bevise at det er blitt godkjent av Apple ved å svare med et sertifikat utstedt av Apple, som enheten bekrefter. Enheten sender deretter en utfordring som tilbehøret må sende et signert svar på. Denne prosessen håndteres i sin helhet av en tilpasset integrert krets (IC) som Apple leverer til godkjente tilbehørsprodusenter, og den er åpen for tilbehøret.

Tilbehør kan be om tilgang til ulike transportmetoder og -funksjonalitet. Det kan for eksempel be om tilgang til strømmer av digital lyd via Lightning- eller USB-C-kabelen eller stedsinformasjon som gis via Bluetooth. En autentiserings-IC er utformet for å sikre at

kun godkjent tilbehør gis full tilgang til enheten. Hvis et tilbehør ikke støtter autentisering, begrenses tilgangen og gis kun til analog lyd og et lite delsett av serielle (UART) kontroller for avspilling av lyd.

AirPlay tar også i bruk autentiserings-IC-en for å stadfeste at mottakere har blitt godkjent av Apple. Strømmer med AirPlay-lyd og CarPlay-video bruker MFi-SAP (Secure Association Protocol) som krypterer kommunikasjon mellom tilbehøret og enheten ved hjelp av AES128 i CTR-modus. Kortvarige nøkler utveksles ved hjelp av ECDH-nøkkelutveksling (Curve25519) og signeres med RSA-nøkkelen (1024-bit) til autentiserings-IC-en som en del av STS-protokollen (Station-to-Station).

Programsikkerhet i macOS

Oversikt over programsikkerhet i macOS

Programsikkerhet i macOS består av en rekke overlappende lag, hvorav det første er alternativet om å kjøre kun signerte og godkjente programmer fra App Store. I tillegg har macOS lagdelt beskyttelse for å bidra til å sikre at programmer som lastes ned fra internett, er frie for kjent skadelig programvare. macOS tilbyr teknologi for å gjenkjenne og fjerne skadelig programvare og ytterligere beskyttelser skapt for å hindre at programmer som ikke er godkjent, får tilgang til brukerdata. Tjenester fra Apple, som attestering og XProtect- og MRT-oppdateringer, er utviklet for å forhindre installering av skadelig programvare og for å sikre en rask og effektiv registrering og responsprosess for å blokkere og fjerne skadelig programvare som kan ha unngått registrering. I siste instans er macOS-brukere frie til å benytte den sikkerhetsmodellen som passer dem best, og det inkluderer å kjøre helt usignert og ikke-godkjent kode.

Kodesigneringsprosessen for programmer i macOS

Alle apper fra App Store er signert av Apple. Signeringen skal sikre at de ikke har blitt tuklet med eller endret. Apple signerer alle programmer som leveres med Apple-enheter.

I macOS 10.15 må alle programmer som distribueres utenfor App Store, være signert av utvikleren med et Apple-utstedt Developer ID-sertifikat (kombinert med en privat nøkkel) og attestert av Apple for å kjøre under standard Gatekeeper-innstillinger. Programmer utviklet internt bør også signeres med en Apple-utstedt Developer ID, slik at brukere kan bekrefte deres integritet.

I macOS fungerer kodesignering og attestering uavhengig, og kan utføres av forskjellige aktører for forskjellige formål. Kodesignering utføres av utvikleren ved hjelp av deres Developer ID-sertifikat (utstedt av Apple), og verifisering av denne signaturen beviser overfor brukeren at en utviklers programvare ikke har blitt tuklet med siden utvikleren bygget og signerte den. Attestering kan utføres av hvem som helst i programvaredistribusjonskjeden og beviser at Apple har fått en kopi av koden for å kontrollere den for skadelig programvare og bekrefte at ingen skadelig programvare ble funnet. Resultatet av attestering er en etikett som oppbevares på Apple-tjenere og som kan knyttes til programmet (av hvem som helst) uten å ugyldiggjøre utviklerens signatur.

Obligatoriske tilgangskontroller (MAC-er) krever kodesignering for å aktivere rettigheter beskyttet av systemet. For eksempel må programmer som krever tilgang gjennom brannmuren være kodesignert med den relevante MAC-rettigheten.

Gatekeeper og beskyttelse ved kjøring i macOS

Gatekeeper

macOS inkluderer en teknologi som kalles Gatekeeper. Den er utviklet for å sikre, som standard, at kun godkjent programvare kjører på en brukers Mac. Når en bruker laster ned og åpner et program, et programtillegg eller en installeringspakke fra utenfor App Store, verifiserer Gatekeeper at programvaren er fra en identifisert utvikler, er attestert av Apple til å være fri for skadelig innhold og ikke har blitt endret. Gatekeeper ber også om brukergodkjenning før nedlastet programvare åpnes første gang for å sikre at brukeren ikke har blitt lurt til å kjøre kjørbar kode den trodde var en vanlig datafil.

Standardinnstillingen er at Gatekeeper bidrar til å sikre at all nedlastet programvare har blitt signert av App Store eller signert av en registrert utvikler og attestert av Apple. Både vurderings- og attesteringsprosessen til App Store er utviklet for å sikre at apper ikke inneholder kjent skadelig programvare. Derfor *kontrolleres all programvare i macOS for kjent skadelig programvare som standard første gang det åpnes, uavhengig av hvordan den havnet på Macen*.

Brukere og organisasjoner har muligheten til å kun tillate programvare som installeres fra App Store. Brukere kan også overstyre Gatekeepers regler for å åpne all programvare, med mindre det begrenses av en MDM-løsning. Organisasjoner kan bruke MDM til å konfigurere Gatekeeper-innstillinger, inkludert å tillate programvare signert med alternative identiteter. Gatekeeper kan også deaktiveres helt, hvis nødvendig.

Gatekeeper beskytter også mot distribusjon av ondsinnede programtillegg som følger med ufarlige programmer. Her utløser bruken av programmet innlasting av et skadelig programtillegg uten at brukeren er klar over det. Når det er nødvendig, åpner Gatekeeper programmer fra tilfeldige, skrivebeskyttede plasseringer. Dette er designet for å hindre automatisk innlasting av programtillegg som er distribuert sammen med programmet.

Beskyttelse ved kjøring

Systemfiler, ressurser og kjernen er beskyttet mot en brukers programområde. Alle apper i App Store skilles fra hverandre ved hjelp av sandkasseteknologi, som begrenser tilgangen til data i andre apper. Hvis et program fra App Store trenger tilgang til data fra et annet program, må det gå via API-ene og tjenestene i macOS.

Beskyttelse mot skadelig programvare i macOS

Apple bruker en prosess med trusselinformasjon for raskt å identifisere og blokkere skadelig programvare. Beskyttelse mot skadelig programvare er delt inn i tre lag:

1. *Forhindre start eller kjøring av skadelig programvare:* App Store eller Gatekeeper og attestering.

2. *Blokkere skadelig programvare fra å kjøre på kundenes systemer:* Gatekeeper, attestering og XProtect.

3. *Avhjelpe skadelig programvare som har kjørt:* MRT

Det første laget med beskyttelse er utviklet for å sørge for at skadelig programvare ikke spres samt at den ikke kan startes i det hele tatt. Det er målet til *App Store* og *Gatekeeper* kombinert med *attestering*.

Det neste laget med beskyttelse bidrar til å sørge for at skadelig programvare som havner på Macen, raskt blir identifisert og blokkert, både for å hindre spredning og for å avhjelpe tilstanden på Macen der den har fått fotfeste. *XProtect* bidrar til denne beskyttelsen sammen med Gatekeeper og attestering.

Til slutt sørger *MRT* for å avhjelpe skadelig programvare som har blitt kjørt.

Disse beskyttelsene samarbeider for å sikre mønsterpraksisbeskyttelse mot virus og skadelig programvare. Det finnes andre former for beskyttelse, spesielt på Macer med Apple Silicon, for å begrense de potensielle skadene etter eventuell kjøring av skadelig programvare. Se Beskytte apptilgang til brukerdata for mer informasjon om hvordan macOS kan beskytte brukernes data mot skadelig programvare, og Operativsystemintegritet for mer informasjon om hvordan macOS kan begrense hva skadelig programvare kan gjøre på systemet.

Attestering

Attestering er en tjeneste fra Apple. Den skanner etter skadelig programvare. Utviklere som ønsker å distribuere programmer for macOS utenfor App Store, sender inn programmene sine for skanning som en del av distribusjonsprosessen. Apple skanner programvaren for å avdekke kjent skadelig programvare. Hvis slik ikke finnes, utstedes en attesteringsetikett. Utviklere inkluderer vanligvis denne etiketten i programmet, slik at Gatekeeper kan kontrollere og starte programmet selv når enheten ikke er koblet til internett.

Apple kan også utstede en tilbakekallingsetikett for programmer som er definert som skadelige, selv om de har blitt attestert tidligere. macOS ser regelmessig etter nye tilbakekallingsetiketter, slik at Gatekeeper har den nyeste informasjonen og kan forhindre at slike filer åpnes. Denne prosessen kan raskt blokkere skadelige programmer, siden oppdateringer gjøres mye hyppigere i bakgrunnen enn bakgrunnsoppdateringene med nye XProtect-signaturer. I tillegg kan denne beskyttelsen brukes både for programmer som har blitt attestert tidligere, og for programmer som ikke har det.

XProtect

macOS har innebygd antivirusteknologi, kalt *XProtect*, for signaturbasert gjenkjenning av skadelig programvare. Systemet bruker YARA-signaturer, et verktøy som brukes til å utføre signaturbasert gjenkjenning av skadelig programvare, som Apple oppdaterer regelmessig. Apple overvåker nye infeksjoner og arter av skadelig programvare og oppdaterer signaturene automatisk, uavhengig av systemoppdateringer, for å bidra til å beskytte en Mac mot angrep. XProtect oppdager og blokkerer kjøring av kjent skadelig programvare automatisk. I macOS 10.15 eller nyere søker XProtect etter kjent skadelig innhold hver gang:

- et program startes for første gang
- et program har blitt endret (i filsystemet)
- XProtect-signaturer oppdateres

Når XProtect oppdager kjent skadelig programvare, blokkeres programvaren samtidig som brukeren varsles og får muligheten til å flytte programvaren til papirkurven.

Merk: Attestering er effektivt mot kjente filer (eller filhasher) og kan brukes på programmer som har vært startet tidligere. XProtects signaturbaserte regler er mer generiske enn en

konkret filhash, og derfor kan de avdekke varianter som Apple ikke har sett. XProtect har tregere oppdateringssyklus enn attestering og skanner bare programmer som har blitt endret eller den første gangen de startes.

Verktøy for fjerning av skadelig programvare

Hvis skadelig programvare skulle finne veien til en Mac, inkluderer macOS også teknologi for å avhjelpe infeksjoner. *MRT (Malware Removal Tool)* er en motor i macOS som avhjelper infeksjoner basert på oppdateringer levert automatisk fra Apple (som en del av automatiske oppdateringer av systemdatafiler og sikkerhetsoppdateringer). MRT fjerner skadelig programvare når den mottar oppdatert informasjon, og den fortsetter å se etter infeksjoner ved oppstart og pålogging. MRT starter ikke Macen på nytt automatisk.

Automatiske sikkerhetsoppdateringer

Apple lanserer oppdateringer for XProtect og MRT basert på den nyeste tilgjengelige trusselinformasjonen. Standardinnstillingen er at macOS ser etter disse oppdateringene daglig. Attesteringsoppdateringer distribueres via CloudKit-synkronisering, noe som skjer mye oftere.

Responsprosess

Når ny skadelig programvare oppdages, kan en rekke tiltak iverksettes:

- Tilknyttede Developer ID-sertifikater tilbakekalles.
- Det utstedes tilbakekallingsetiketter for attestering for alle filer (programmer og tilknyttede filer).
- Det utvikles og distribueres XProtect-signaturer.
- Det utvikles og distribueres MRT-signaturer.
- Disse signaturene distribueres også for programvare som har blitt attestert tidligere, og nye registreringer kan føre til at én eller flere tidligere handlinger skjer.

Til slutt iverksetter en registrering av skadelig programvare flere trinn de neste sekundene, timene og dagene for å sikre optimal beskyttelse for Mac-brukere.

Kontroll av programtilgang til filer i macOS

Apple mener at brukere bør ha fullt innsyn i, samtykke og kontroll over hva programmer gjør med dataene deres. I macOS 10.15 håndheves denne modellen av systemet for å bidra til å sikre at alle programmer må innhente brukersamtykke før de får tilgang til filer i Dokumenter, Nedlastinger, Skrivebord, iCloud Drive og nettverksvolumer. I macOS 10.13 eller nyere må programmer som krever tilgang til hele lagringsenheten, legges til eksplisitt i Systemvalg. I tillegg krever tilgjengelighets- og automatiseringsfunksjoner brukertillatelse for å bidra til å sikre at de ikke omgår andre beskyttelsesmekanismer. Avhengig av tilgangsregelen, kan brukere bli bedt om eller måtte endre innstillingen i Systemvalg > Sikkerhet og personvern > Personvern:

Objekt	Bruker spurt av program	Bruker må redige-re systempersonverninnstillinger
Tilgjengelighet		✅

Objekt	Bruker spurt av program	Bruker må redige- re systempersonverninnstillinger
Full intern lagringstilgang		✓
Filer og mapper *Merk:* Inkluderer: Skrivebord, Doku- menter, Nedlastinger, nettverksvolumer og uttakbare volumer	✓	
Automatisering (Apple-hendelser)	✓	

Objekter i brukerens papirkurv beskyttes fra eventuelle programmer som bruker full disktilgang. Brukeren vil ikke bli bedt om programtilgang. Filer må først flyttes fra papirkurven til en annen plassering hvis brukeren vil ha tilgang til dem.

En bruker som aktiverer FileVault på en Mac bes om å oppgi gyldige akkreditiver før oppstartsprosessen fortsetter og tilgang gis til spesialiserte oppstartsmoduser. Uten gyldig påloggingsakkreditiver eller en gjenopprettingsnøkkel, forblir hele volumet kryptert og beskyttes mot uautorisert tilgang, selv hvis den fysiske lagringsenheten fjernes og kobles til en annen datamaskin.

For å beskytte data i et bedriftsoppsett, bør IT-avdelingen definere og håndheve FileVault-konfigurasjonsregler ved hjelp av MDM. Organisasjoner har flere muligheter for å administrere krypterte volumer, inkludert gjenopprettingsnøkler for institusjonen, personlige gjenopprettingsnøkler (som kan lagres med MDM for deponering) eller en kombinasjon av begge. Nøkkelrotering kan også angis som en regel i MDM.

Sikre funksjoner i Notater-appen

Sikre notater

Notater-appen har en funksjon for sikre notater som gir brukerne mulighet til å beskytte innholdet i bestemte notater. Sikre notater krypteres ved hjelp av et brukeroppgitt passord som kreves for å vise notater på iOS-, iPadOS- og macOS-enheter og på iCloud-nettstedet. Hver iCloud-konto (inkludert «På min»-enhetskontoer) kan ha et separat passord.

Når en bruker sikrer et notat, avledes en nøkkel på 16 bytes fra brukerens passord ved hjelp av PBKDF2 og SHA256. Notatet og alle vedleggene krypteres ved hjelp av AES med Galois/Counter Mode (AES-GCM). Nye oppføringer opprettes i Core Data og CloudKit for å lagre det krypterte notatet, vedlegg, etiketten og intialiseringsvektoren. Etter at de nye oppføringene er opprettet, slettes de originale, ukrypterte dataene. Vedlegg som støtter kryptering inkluderer bilder, tegninger, tabeller, kart og nettsider. Notater som inneholder andre typer vedlegg, kan ikke krypteres, og vedlegg som ikke støttes, kan ikke legges til i sikre notater.

Får å vise et sikkert notat, må brukeren angi passordet eller autentisere ved hjelp av Touch ID eller Face ID. Når autentisering av brukeren lykkes, enten det er for å vise eller opprette et sikkert notat, åpner Notater en sikker økt. Når den sikre økten er åpen, kan brukeren vise eller sikre andre notater uten ytterligere autentisering. Den sikre økten gjelder imidlertid kun notatene som er beskyttet med det oppgitte passordet. Brukeren må fortsatt autentisere for notater som er beskyttet av et annet passord. Den sikre økten lukkes når:

- brukeren trykker på Lås nå-knappen i Notater
- Notater flyttes til bakgrunnen i mer enn 3 minutter (8 minutter i macOS)
- iOS- eller iPadOS-enheten låses

For å endre passordet for et sikkert notat må brukeren angi gjeldende passord, ettersom Touch ID og Face ID ikke er tilgjengelige ved endring av passord. Når et nytt passord er valgt, pakker Notater nøklene til alle eksisterende notater i samme konto som er kryptert med det forrige passordet, på nytt.

Hvis en bruker skriver passordet feil tre ganger på rad, viser Notater et brukerhint, hvis brukeren har oppgitt et hint under konfigureringen. Hvis brukeren fortsatt ikke husker passordet, kan det nullstilles i Notater-innstillinger. Denne funksjonen gjør det mulig for brukerne å opprette sikre notater med nye passord, men den tillater ikke at de ser tidligere sikrede notater. Tidligere sikrede notater kan fortsatt vises hvis brukerne husker det gamle passordet. Tilbakestilling av passordet krever brukerens passord for iCloud-kontoen.

Delte notater

Notater som ikke er gjennomgående kryptert med et passord, kan deles med andre. Delte notater bruker fortsatt den CloudKit-krypterte datatypen til tekst eller vedlegg brukeren plasserer i et notat. Ressurser krypteres alltid med en nøkkel som krypteres i CKRecord. Metadata, for eksempel opprettet- og endret-datoene, krypteres ikke. CloudKit administrerer prosessen der deltakere kan kryptere og dekryptere hverandres data.

Sikre funksjoner i Snarveier-appen

Snarveier i Snarveier-appen kan etter ønske synkroniseres på tvers av Apple-enheter ved hjelp av iCloud. Snarveier kan også deles med andre brukere via iCloud. Snarveier lagres lokalt i et kryptert format.

Tilpassede snarveier er allsidige. De ligner på prosedyrer eller programmer. Når snarveier lastes ned fra internett, advares brukeren om at snarveien ikke er kontrollert av Apple og får muligheten til å inspisere snarveien. For å beskytte mot skadelige snarveier, lastes det ned oppdaterte definisjoner for skadelig programvare for å identifisere ondsinnede snarveier når de kjøres.

Tilpassede snarveier kan også kjøre brukerspesifisert JavaScript på nettsider i Safari når de aktiveres fra delingsarket. For å beskytte mot skadelig JavaScript som, for eksempel, lurer brukeren til å kjøre en prosedyre på et nettsted for et sosialt medium som samler inn dataene deres, valideres JavaScript mot de forannevnte definisjoner for skadelig programvare. Første gang en bruker kjører JavaScript på et domene, bes brukeren om å tillate at snarveier som inneholder JavaScript, kan kjøre på gjeldende nettside for det domenet.

Sikkerhetstjenester

Oversikt over sikkerhetstjenester

Apple har utviklet et robust sett med tjenester for å hjelpe brukere med å få enda større nytte og glede av enhetene sine. Disse tjenestene tilbyr kraftige egenskaper for skylagring, synkronisering, arkivering av passord, autentisering, betaling, meldinger, kommunikasjoner og annet, samtidig som de beskytter brukernes personvern og datasikkerhet.

Disse tjenestene inkluderer iCloud, Logg på med Apple, Apple Pay, iMessage, Spør bedriften, FaceTime, «Hvor er?» og Kontinuitet, og de krever en Apple-ID eller administrert Apple-ID. I enkelte tilfeller kan ikke en administrert Apple-ID brukes med enkelte tjenester, for eksempel Apple Pay.

Merk: Ikke alt av Apples tjenester og innhold er tilgjengelig i alle land eller områder.

Apple-ID og Administrert Apple-ID

Oversikt over Apple-ID-sikkerhet

Oversikt

En Apple-ID er kontoen som brukes til å logge på Apple-tjenester som blant annet iCloud, iMessage, FaceTime, iTunes Store, App Store, Apple TV-appen og Apple Books. Det er viktig at brukerne sørger for at Apple-ID-ene sikres, for å bidra til at uvedkommende ikke får tilgang til kontoene. For å bidra til dette krever Apple-ID-er sterke passord som:

- skal være minst åtte tegn lange
- skal bestå av både bokstaver og tall
- ikke kan inneholde flere enn tre like tegn etter hverandre
- ikke skal være et kjent passord

Brukerne oppfordres til å legge til ekstra bokstaver og tegnsetting for å gjøre passordene enda sterkere.

Apple varsler også brukere med e-postmeldinger og pushvarslinger når det gjøres viktige endringer for kontoen, for eksempel hvis et passord eller fakturainformasjon er blitt endret eller hvis Apple-ID-en er blitt brukt til å logge på en ny enhet. Hvis det er noe som ser ukjent ut, får brukerne beskjed om å endre Apple-ID-en og passordet umiddelbart.

I tillegg har Apple utformet en rekke retningslinjer og prosedyrer for å beskytte brukerkontoer. De går blant annet ut på å begrense antall forsøk på å logge inn og tilbakestille passord, effektiv overvåking av forsøk på svindel og regelmessig gjennomgang av regler og rutiner, slik at Apple kan tilpasse seg til eventuell ny informasjon som kan påvirke brukersikkerheten.

Merk: Passordreglene for administrert Apple-ID defineres av en administrator i Apple School Manager eller Apple Business Manager.

Tofaktorautentisering

For å hjelpe brukere med å sikre kontoene sine ytterligere tilbyr Apple *tofaktorautentisering*, som er et ekstra sikkerhetslag for Apple-ID-er. Den ble utformet for å sikre at kun kontoens eier kan få tilgang til kontoen, selv om andre kjenner passordet. Med tofaktorautentisering kan man bare få tilgang til brukerens konto fra godkjente enheter, for eksempel brukerens iPhone, iPad, iPod touch eller Mac, eller fra andre enheter etter at tilgangen har blitt bekreftet med en av de godkjente enhetene eller med et godkjent telefonnummer. Det kreves to opplysninger for å logge på en ny enhet første gang: Apple-ID-passordet og en sekssifret verifiseringskode som blir vist på brukerens godkjente enheter eller sendt til et godkjent telefonnummer. Ved å oppgi koden bekrefter brukerne at de godkjenner den nye enheten og at det er trygt å logge på. Fordi det ikke lenger er nok med passordet for å få tilgang til en brukerkonto, er tofaktorautentisering en forbedring av sikkerheten til brukernes Apple-ID og alle personopplysninger de lagrer hos Apple. Den er integrert direkte i iOS, iPadOS, macOS, tvOS, watchOS og autentiseringsløsningen som benyttes på Apples nettsteder.

Når en bruker logger seg på et Apple-nettsted med en nettleser, sendes en tofaktorforespørsel til alle godkjente enheter som er knyttet til brukerens iCloud-konto, for å be om godkjenning for nettøkten. Hvis brukeren logger seg på et Apple-nettsted med en nettleser på en godkjent enhet, vises verifiseringskoden lokalt på enheten som brukes. Når brukeren skriver inn koden på den enheten, godkjennes nettøkten.

Gjenoppretting av konto

Hvis brukeren glemmer passordet til en Apple-ID-konto, kan det tilbakestilles på en godkjent enhet. Hvis en godkjent enhet ikke er tilgjengelig, men passordet er kjent, kan brukeren bruke et godkjent telefonnummer til å autentisere via verifisering på tekstmelding. I tillegg kan en tidligere brukt kode brukes til å nullstille Apple-ID-en sammen med SMS for umiddelbar gjenoppretting. Hvis disse alternativene ikke er mulige, må prosessen for gjenoppretting av konto følges. Du finner mer informasjon i Apple-kundestøtteartikkelen Slik bruker du kontogjenoppretting når du ikke kan tilbakestille Apple-ID-passordet ditt.

Administrert Apple-ID-sikkerhet

Administrerte Apple-ID-er fungerer omtrent på samme måte som en Apple-ID, men de eies og kontrolleres av en bedrift eller av utdanningsorganisasjoner. Disse organisasjonene kan tilbakestille passord, begrense kjøp og kommunikasjon, for eksempel FaceTime og Meldinger, og angi tillatelser som er basert på roller for ansatte, lærere og elever.

For Administrerte Apple-ID-er er enkelte tjenester deaktivert (for eksempel, Apple Pay, iCloud-nøkkelring, HomeKit og «Hvor er?»).

Inspeksjon av administrerte Apple-ID-er

Administrerte Apple-ID-er støtter også *inspeksjon*, slik at det er mulig for organisasjoner å etterleve regelverket, herunder personvernregler. En Apple School Manager-administrator, -leder eller -lærer kan inspisere spesifikke administrerte Apple-ID-kontoer.

De som skal utføre inspeksjoner, kan kun overvåke kontoer som er på et lavere nivå enn dem i organisasjonens hierarki. Lærere kan for eksempel overvåke elever, ledere kan inspisere lærere og elever, og administratorer kan inspisere ledere, lærere og elever.

Når det bes om akkreditiver for å foreta inspeksjon gjennom Apple School Manager, utstedes det en spesialkonto som har tilgang kun til de administrerte Apple-ID-ene som det er bedt om inspeksjon av. Deretter kan inspektøren lese og endre brukerens innhold som er lagret i iCloud eller i apper med CloudKit-støtte. Alle forespørsler om inspeksjonstilgang loggføres i Apple School Manager. Loggene viser hvem inspektøren var, den administrerte Apple-ID-en som inspektøren ba om tilgang til, tidspunktet for forespørselen og om inspeksjonen ble utført eller ikke.

Administrerte Apple-ID-er og personlige enheter

Administrerte Apple-ID-er kan også brukes på personlig eide iOS- og iPadOS-enheter og Macer. Elevene logger på iCloud med den administrerte Apple-ID-en som de har fått av skolen, og et tilleggspassord for hjemmebruk som fungerer som den andre faktoren i Apple-ID-ens tofaktorautentiseringsprosess. Når elever bruker en administrert Apple-ID på en personlig enhet, er ikke iCloud-nøkkelring tilgjengelig, og institusjonen kan også begrense andre funksjoner som FaceTime eller Meldinger. iCloud-dokumenter som elevene oppretter når de er pålogget, kan bli inspisert, som beskrevet tidligere i denne delen.

iCloud

Oversikt over iCloud-sikkerhet

iCloud lagrer brukerens kontakter, kalendere, bilder, dokumenter og mer og holder automatisk informasjonen oppdatert på alle enhetene. iCloud kan også brukes av tredjepartsapper til å lagre og synkronisere dokumenter og nøkkelverdier for appdata på den måten som utvikleren har definert. Brukere konfigurerer iCloud ved å logge på med en Apple-ID og velge hvilke tjenester de ønsker å bruke. Enkelte iCloud-funksjoner, iCloud Drive og iCloud-sikkerhetskopi kan deaktiveres av IT-administratorer ved å bruke MDM-konfigurasjonsprofiler. Tjenesten har ingen meninger om hva som lagres og håndterer alt filinnhold på samme måte.

Alle filer deles opp i mindre deler og krypteres av iCloud ved hjelp av AES128 og en nøkkel som er avledet fra innholdet i de enkelte delene med nøklene som bruker SHA256. Nøklene og filens metadata lagres av Apple i brukerens iCloud-konto. De krypterte delene av filen lagres, uten noe brukeridentifiserende informasjon eller nøklene, ved hjelp av både Apple- og tredjepartslagringstjenester, eksempelvis Amazon Web Services eller Google Cloud Platform,— men disse partnerne har ikke nøkler til å dekryptere brukerens data som er lagret på tjenerne deres.

iCloud Drive-sikkerhet

iCloud Drive legger til kontobaserte nøkler for å beskytte dokumenter som er lagret i iCloud. iCloud Drive deler opp og krypterer filinnholdet og lagrer de krypterte delene ved hjelp av tjenester fra tredjeparter. Filinnholdsnøklene blir imidlertid pakket inn av oppføringsnøkler som er lagret med iCloud Drive-metadataene. Disse oppføringsnøklene beskyttes så av tjenestenøkkelen til brukerens iCloud Drive, som deretter lagres sammen med brukerens iCloud-konto. Brukerne får tilgang til metadataene for iCloud-dokumentene ved at de er blitt godkjent av iCloud, men de må også ha tjenestenøkkelen for iCloud Drive for å åpne beskyttede deler av iCloud Drive-lagringsplassen.

iCloud Drive-sikkerhetskopiering

iCloud sikkerhetskopierer også daglig informasjon via Wi-Fi, blant annet enhetsinnstillinger, appdata, bilder og videoer i Kamerarull og samtaler i Meldinger-appen. iCloud sikrer innholdet ved å kryptere det når det sendes via internett, ved å lagre det i et kryptert format og ved å bruke sikre kjennetegn for autentisering. Sikkerhetskopiering i iCloud skjer kun når enheten er låst, tilkoblet en strømkilde og har tilgang til internett via Wi-Fi. På grunn av krypteringen i iOS og iPadOS er systemet utviklet slik at data beskyttes mens sikkerhetskopiering og gjenoppretting utføres trinnvis i bakgrunnen.

Når det opprettes filer i databeskyttelsesklasser som ikke er tilgjengelige når enheten er låst, blir de filspesifikke nøklene kryptert ved hjelp av klassenøklene fra nøkkeletuiet for iCloud-sikkerhetskopiering, og filene sikkerhetskopieres til iCloud i deres originale krypterte tilstand. Alle filer krypteres under transport, og, når de lagres, ved hjelp av kontobaserte nøkler som beskrevet i CloudKit.

Nøkkeletuiet for iCloud-sikkerhetskopiering inneholder asymmetriske nøkler (Curve25519) for databeskyttelsesklasser som ikke er tilgjengelige når enheten er låst. Sikkerhetskopisettet lagres i brukerens iCloud-konto og består av en kopi av brukerens filer og nøkkeletuiet for iCloud-sikkerhetskopiering. Nøkkeletuiet for iCloud-sikkerhetskopiering er beskyttet av en tilfeldig nøkkel, som også lagres i sikkerhetskopisettet. (Brukerens iCloud-passord benyttes ikke til kryptering, slik at eksisterende sikkerhetskopier ikke blir ugyldiggjorte selv om iCloud-passordet endres.)

Når det legges en sikkerhetskopi av brukerens nøkkelringdatabase på iCloud, er den fortsatt beskyttet av en UID-integrert nøkkel. Dette gjør at nøkkelringen kan gjenopprettes kun til enheten den opprinnelig kom fra, og det betyr at ingen andre, heller ikke Apple, kan lese brukerens nøkkelringobjekter.

Ved en gjenoppretting hentes de sikkerhetskopierte filene, nøkkeletuiet for iCloud-sikkerhetskopiering og nøklene for nøkkeletuiet fra brukerens iCloud-konto. Nøkkeletuiet for iCloud-sikkerhetskopiering blir dekryptert ved hjelp av nøkkelen sin. De filspesifikke nøklene i nøkkeletuiet brukes deretter til å dekryptere filene i sikkerhetskopisettet, som så skrives som nye filer til filsystemet. På denne måten krypteres de på nytt i henhold til databeskyttelsesklassen.

Sikkerhet for iCloud-sikkerhetskopi

Dette innholdet sikkerhetskopieres med iCloud-sikkerhetskopiering:

- informasjon om kjøpt musikk, filmer, TV-programmer, apper og bøker. En brukers iCloud-sikkerhetskopier inneholder informasjon om kjøpt innhold som befinner seg på brukerens enhet, men ikke selve det kjøpte innholdet. Når brukeren gjenoppretter fra en iCloud-sikkerhetskopi, lastes kjøpt innhold automatisk ned fra iTunes Store, App Store, Apple TV-appen eller Apple Books. Noen typer innhold lastes ikke automatisk ned i alle land eller områder, og det kan hende at tidligere kjøp ikke er tilgjengelige hvis de er blitt refundert eller ikke lenger er tilgjengelige i butikken. Fullstendig kjøpshistorikk er tilknyttet brukerens Apple-ID.
- bilder og videoer på brukerens enheter. Vær oppmerksom på at hvis en bruker slår på iCloud-bilder i iOS 8.1 eller nyere, iPadOS 13.1 eller nyere eller OS X 10.10.3 eller nyere, er bildene og videoene allerede lagret i iCloud, så de tas ikke med i brukerens iCloud-sikkerhetskopi.
- kontakter, kalenderhendelser, påminnelser og notater
- enhetsinnstillinger
- appdata
- organisering av Hjem-skjermen og apper
- HomeKit-konfigurasjon
- Nødinfo
- Visual Voicemail-passord (krever SIM-kortet som ble brukt under sikkerhetskopieringen)
- iMessage-, Spør bedriften-, SMS- og MMS-meldinger (krever SIM-kortet som ble brukt under sikkerhetskopieringen)

Når Meldinger i iCloud er aktivert, fjernes iMessage-, Spør bedriften-, SMS- og MMS-meldinger fra brukerens eksisterende iCloud-sikkerhetskopi og lagres i stedet i en CloudKit-beholder med gjennomgående kryptering for Meldinger. Brukerens iCloud-sikkerhetskopi beholder en nøkkel til denne beholderen. Hvis brukeren senere deaktiverer iCloud-sikkerhetskopi, rulleres denne beholderens nøkkel, og den nye nøkkelen lagres kun i iCloud-nøkkelring (utilgjengelig for Apple og eventuelle tredjeparter), og nye data som skrives til beholderen, kan ikke dekrypteres med den gamle beholdernøkkelen.

Nøkkelen som brukes til å gjenopprette meldingene i iCloud-sikkerhetskopi, plasseres på to steder, iCloud-nøkkelring og en sikkerhetskopi i CloudKit. Sikkerhetskopieringen til CloudKit utføres hvis iCloud-sikkerhetskopi er aktivert og gjenopprettes uten betingelser uavhengig av om brukeren gjenoppretter en iCloud-sikkerhetskopi eller ikke.

Gjennomgående kryptering i CloudKit

Mange Apple-tjenester nevnt i Apple-kundestøtteartikkelen Oversikt over iCloud-sikkerhet bruker gjennomgående kryptering med en CloudKit-tjenestenøkkel som er beskyttet av iCloud-nøkkelringsynkronisering. For disse CloudKit-beholderne er nøkkelringhierarkiet rotfestet i iCloud-nøkkelring og deler derfor sikkerhetsegenskapene til iCloud-nøkkelring. Nøklene er nemlig kun tilgjengelige på brukerens godkjente enheter, ikke for Apple eller andre tredjeparter. Hvis tilgang til iCloud-nøkkelringdata går tapt, nullstilles dataene

i CloudKit, og hvis data er tilgjengelig fra den godkjente lokale enheten, lastes de opp på nytt til CloudKit. Du finner mer informasjon om dette under Deponeringssikkerhet for iCloud-nøkkelring.

Meldinger i iCloud bruker også gjennomgående CloudKit-kryptering med en CloudKit-tjenestenøkkel beskyttet av iCloud-nøkkelringsynkronisering. Hvis brukeren har aktivert iCloud-sikkerhetskopi, sikkerhetskopieres CloudKit-tjenestenøkkelen som brukes for Meldinger i iCloud-beholderen til iCloud for å gjøre det mulig for brukeren å gjenopprette meldingene selv om de har mistet tilgang til iCloud-nøkkelring og deres godkjente enheter. Denne iCloud-tjenestenøkkelen rulleres hver gang brukeren slår av iCloud-sikkerhetskopi.

Situasjon	Brukergjenopprettingsvalg for gjennomgående CloudKit-kryptering
Tilgang til godkjent enhet	Datagjenoppretting mulig ved hjelp av en godkjent enhet eller iCloud-nøkkelringgjenoppretting.
Ingen godkjente enheter	Datagjenoppretting kun mulig ved hjelp av iCloud-nøkkel-ringgjenoppretting.
iCloud-sikkerhetskopi aktivert og tilgang til godkjent enhet	Datagjenoppretting mulig ved hjelp av iCloud-sikkerhetsko-pi, tilgang til en godkjent enhet eller iCloud-nøkkelringg-jenoppretting.
iCloud-sikkerhetskopi aktivert og ingen tilgang til godkjent enhet	Datagjenoppretting mulig ved hjelp av iCloud-sikkerhetsko-pi eller iCloud-nøkkelringgjenoppretting.
iCloud-sikkerhetskopi deaktivert og tilgang til godkjent enhet	Datagjenoppretting mulig ved hjelp av en godkjent enhet eller iCloud-nøkkelringgjenoppretting.
Sikkerhetskopi deaktivert og ingen godkjente enheter	Datagjenoppretting kun mulig ved hjelp av iCloud-nøkkel-ringgjenoppretting.

Administrering av koder og passord

Oversikt over passordsikkerhet

iOS, iPadOS og macOS gjør det enkelt for brukere å autentisere til tredjepartsapper og nettsider som bruker passord. Den beste måten å administrere passord på er å ikke måtte bruke dem. Med Logg på med Apple kan brukere logge seg på tredjepartsapper og nettsteder uten å måtte opprette og administrere en ekstra konto eller et ekstra passord, samtidig som påloggingen beskyttes med tofaktorautentisering for Apple-ID. Hvis nettstedet ikke støtter Logg på med Apple, vil funksjonen for å lage automatiske, sterke passord gjøre at brukerens enhet automatisk kan opprette, synkronisere og oppgi unike, sterke passord for nettsteder og apper. I iOS og iPadOS arkiveres passord på en spesiell nøkkelring for autoutfylling av passord som er brukerstyrt og kan administreres ved å gå til Innstillinger > Passord.

I macOS kan arkiverte passord administreres i Safaris passordvalg. Denne funksjonen kan også brukes til å synkronisere passord som brukeren oppretter manuelt.

Logg på med Apple-sikkerhet

Logg på med Apple er et personvernvennlig alternativ til andre Single Sign On-systemer. Det er like enkelt og effektivt som ettrykkspålogging, og brukeren får bedre innsyn i og kontroll over personopplysningene sine.

Med Logg på med Apple kan brukere konfigurere en konto og logge seg på apper og nettsteder ved hjelp av Apple-ID-en de allerede har, og de får bedre kontroll over personopplysningene sine. Apper kan bare be om brukerens navn og e-postadresse når kontoen konfigureres, og brukeren har alltid et valg: De kan dele den private e-postadressen med en app eller velge å skjule e-postadressen og bruke Apples nye tjeneste for privat videresending av e-poster i stedet. Denne videresendingstjenesten deler en unik, anonymisert e-postadresse som videresender e-poster til brukerens private e-postadresse, slik at nyttig informasjon fra utvikleren fortsatt kommer frem, samtidig som brukeren får bedre personvern og kontroll over personopplysningene sine.

Logg på med Apple er utviklet for sikkerhet. Alle Logg på med Apple-brukere må aktivere tofaktorautentisering for Apple-ID-en. Tofaktorautentisering beskytter både brukerens Apple-ID og kontoene de har opprettet i appene. I tillegg har Apple utviklet et integrert og personvernvennlig signal for varsling av svindel i Logg på med Apple. Dermed kan utviklere være trygge på at de nye brukerne er ekte mennesker, og ikke roboter eller automatiserte kontoer.

Automatiske, sterke passord

Når iCloud-nøkkelring er aktivert, oppretter iOS, iPadOS og macOS sterke, tilfeldige, unike passord når brukere registrerer seg eller endrer passordet på et nettsted i Safari. I iOS og iPadOS er generering av automatiske, sterke passord også tilgjengelig i apper. Brukere må eventuelt aktivt velge å ikke bruke sterke passord. Genererte passord arkiveres i nøkkelringen og holdes oppdatert på alle enhetene med iCloud-nøkkelring, når det er aktivert.

Som standard har passord som genereres av iOS og iPadOS, en lengde på 20 tegn. De inneholder ett tall, én stor bokstav, to bindestreker og 16 små bokstaver. Disse genererte passordene er sterke, og inneholder 71 bits med entropi.

Passord genereres basert på erfaringer som avgjør om et passordfelt er for passordoppretting. Hvis erfaringen ikke lykkes i å gjenkjenne et kontekstspesifikt passord som brukes for passordoppretting, kan apputviklere sette `UITextContentType.newPassword` på tekstfeltet, og nettutviklere kan sette `autocomplete= “new–password”` på deres <input>-elementer.

Apper og nettsteder kan innføre regler for å bidra til å sikre at de genererte passordene er kompatible med de relevante tjenestene. Utviklere definerer disse reglene ved hjelp av `UITextInputPasswordRules` eller `passwordrules`-attributtet på deres input-elementer. Enheter genererer deretter det sterkeste passordet som er mulig for å overholde disse reglene.

Sikkerhet for autoutfylling av passord

Autoutfylling av passord fyller ut akkreditivene som er lagret i nøkkelringen. Passordadministratoren i iCloud-nøkkelringen og autoutfylling av passord tilbyr følgende funksjoner:

- Fylle ut akkreditiver i apper og på nettsteder
- Generering av sterke passord
- Arkivering av passord i både apper og på nettsteder i Safari
- Sikker deling av passord til en brukers kontakter
- Utdeling av passord til en Apple TV i nærheten som ber om akkreditiver

Generering og lagring av passord i apper og utdeling av passord til Apple TV er kun tilgjengelig i iOS og iPadOS.

Autoutfylling av passord for apper

Med iOS og iPadOS kan brukere oppgi lagrede brukernavn og passord i relevante akkreditivfelter i apper på samme måte som Autoutfylling av passord fungerer i Safari. I iOS og iPadOS trykker brukere på en nøkkelbruksmulighet i programvaretastaturets QuickType-linje. For macOS-apper som er laget med Mac Catalyst, vises nedtrekksmenyen Passord under akkreditivrelaterte felter.

Når en app er sterkt knyttet til et nettsted som bruker den samme assosiasjonsmekanismen for app og nettsted, og drives av samme apple-app-site-association-fil, foreslår QuickType-linjen i iOS og iPadOS og nedtrekksmenyen i macOS akkreditiver for appen hvis den er arkivert i nøkkelringen for autoutfylling av passord. Dette gjør det mulig for brukere å velge at Safari-arkiverte akkreditiver kan gjøres tilgjengelig i apper med samme sikkerhetsegenskaper uten at appene må ta i bruk en API.

Autoutfylling av passord utleverer ingen akkreditiver til appen før en bruker samtykker i å gi et akkreditiv til appen. Listen over akkreditiver genereres fra eller vises utenfor appens prosess.

Når en app og et nettsted har en godkjent tilknytning og en bruker angir akkreditiver i en app, kan iOS og iPadOS be brukeren om å arkivere akkreditivene i nøkkelringen for autoutfylling av passord for senere bruk.

Apptilgang til arkiverte passord

iOS- og iPadOS-apper og macOS-programmer kan be om hjelp fra nøkkelringen for autoutfylling av passord for å logge på en bruker ved hjelp av ASAuthorizationPasswordProvider og SecAddSharedWebCredential. Passordleverandøren og forespørselen kan brukes i forbindelse med Logg på med Apple, slik at den samme API-en kalles for å la brukere logge seg på en app uavhengig av om brukerens konto er passordbasert eller ble opprettet ved hjelp av Logg på med Apple.

Appene får bare tilgang til lagrede passord hvis det er godkjent av både apputvikleren og nettsidens administrator, og hvis brukeren har samtykket. Apputviklerne uttrykker at de har til hensikt å få tilgang til passord som er arkivert i Safari ved å inkludere en rettighet for dette i appen. Rettigheten inneholder en liste over de fullt kvalifiserte domenenavnene til tilknyttede nettsteder, og nettstedene må plassere en fil på tjeneren med en liste over de unike appidentifikatorene til apper som er godkjent av Apple.

Når en app med rettigheten com.apple.developer.associated-domains blir installert, foretar iOS og iPadOS en TLS-forespørsel til alle nettstedene som står på listen, og ber om en av følgende filer:

- apple-app-site-association
- .well-known/apple-app-site-association

Hvis app-ID-en til appen som installeres, står på listen i filen, gir iOS og iPadOS nettstedet og appen et merke som viser at de har en godkjent relasjon. Kun når det er en godkjent relasjon vil spørringer til disse to API-ene resultere i en forespørsel til brukeren, som må samtykke før passord gis til appen, oppdateres eller slettes.

Anbefalinger for passordsikkerhet

Oversikt

Passordlisten for autoutfylling av passord i iOS, iPadOS og macOS viser hvilke av brukerens arkiverte passord som brukes på flere nettsteder, hvilke passord som anses som svake og passord som har blitt kompromittert av en datalekkasje.

Hvis det samme passordet brukes til flere tjenester, kan kontoene bli sårbare for såkalte «credential stuffing»-angrep. Hvis noen bryter seg inn i en tjeneste og henter ut passordene, kan angriperne forsøke å bruke de samme akkreditivene i andre tjenester for å få tilgang til flere kontoer. Passordene merkes som *gjenbrukt* hvis det samme passordet brukes for mer enn ett arkivert passord på tvers av forskjellige domener.

Passord merkes som *svake* hvis de er enkle å gjette. iOS, iPadOS og macOS registrerer vanlige passord som brukes til å lage passord som det er enkelt å huske, for eksempel ord fra ordboken, vanlige tegnerstatninger (for eksempel å bruke «p4ss0rd» i stedet for «passord»), vanlige mønstre fra tastaturet (for eksempel «q12we34r» på et QWERTY-tastatur) eller gjentakelser (for eksempel «123123»). Disse mønstrene brukes ofte til å lage passord som oppfyller tjenestens minimumskrav, men de brukes også ofte av angripere som forsøker å finne et passord ved hjelp av brute-force-angrep.

Siden mange tjenester spesifikt krever en PIN-kode på fire eller seks sifre, evalueres slike korte passord basert på andre regler. PIN-koder anses som svake hvis de er blant de PIN-kodene som brukes hyppigst, hvis de består av en økende eller synkende sekvens, for eksempel «1234» eller «8765», eller hvis de inneholder en repetisjon, for eksempel «123123» eller «123321».

Passord merkes som *lekket* hvis Passordovervåking-funksjonen kan vise til at de har vært med i en datalekkasje. Du finner mer informasjon om dette under Passordovervåking.

Svake, gjenbrukte og lekkede passord indikeres enten i passordlisten (macOS) eller finnes i det dedikerte Sikkerhetsanbefalinger-grensesnittet (iOS og iPadOS). Hvis brukeren logger seg på et nettsted i Safari ved hjelp av et arkivert passord som er svært svakt eller har blitt kompromittert av en datalekkasje, vises en advarsel som ber brukeren om å oppgradere til et automatisk, sterkt passord.

Oppgradere kontoautentiseringssikkerhet med Account Authentication Modification-tillegg

Apper som implementerer Account Authentication Modification-tillegg, kan tilby enkle oppgraderinger med ett trykk for passordbaserte kontoer for å bytte til Logg på med Apple eller bruke et automatisk, sterkt passord. Dette tilleggspunktet er tilgjengelig i iOS og iPadOS.

Hvis en app har implementert tilleggspunktet og installeres på en enhet, ser brukere tilleggsoppgraderingsvalg ved visning av Sikkerhetsanbefalinger for akkreditiver som er knyttet til appen, i passordadministratoren i iCloud-nøkkelring i Innstillinger. Oppgraderingene tilbys også når brukere logger på appen med det risikoutsatte akkreditivet. Apper har muligheten til å be systemet om å ikke varsle brukeren om oppgraderingsvalg etter pålogging. Ved bruk av AuthenticationServices-API-en kan apper også aktivere tilleggene og utføre oppgraderinger selv, ideelt fra en kontoinnstillings- eller kontoadministreringsskjerm i appen.

Apper kan velge å støtte sterke passordoppgraderinger, Logg på med Apple eller begge. I en oppgradering av sterkt passord vil systemet generere et automatisk, sterkt passord for brukeren. Hvis det er nødvendig, kan appen tilby tilpassede passordregler som skal følges når det nye passordet genereres. Når en bruker bytter en konto fra å bruke et passord til å bruke Logg på med Apple, gir systemet et nytt Logg på med Apple-akkreditiv til tillegget som skal knyttes til kontoen. Brukerens e-postadresse for Apple-ID gis ikke som en del av akkreditivet. Etter en vellykket oppgradering av Logg på med Apple, sletter systemet det tidligere brukte passordakkreditivet fra brukerens nøkkelring, hvis det er arkivert der.

Account Authentication Modification-tillegg har muligheten til å utføre ytterligere brukerautentisering før en oppgradering utføres. Når det gjelder oppgraderinger startet i passordadministratoren eller etter å ha logget på en app, gir tillegget brukernavnet og passordet for kontoen skal oppgraderes. For oppgraderinger i app blir kun brukernavnet gitt. Hvis tillegget krever ytterligere brukerautentisering, kan det be om å vise tilpasset brukergrensesnitt før det går videre med oppgraderingen. Det tiltenkte brukstilfellet for å vise dette tilpassede brukergrensesnittet er å få brukeren til å oppgi en ekstra autentiseringsfaktor for å autorisere oppgraderingen.

Passordovervåking

Passordovervåking er en funksjon som sammenligner passord lagret i brukerens nøkkelring for autoutfylling av passord mot en kontinuerlig oppdatert og moderert liste med passord som kjent har blitt eksponert i lekkasjer fra forskjellige internettorganisasjoner. Hvis funksjonen slås på, sammenligner den overvåkende protokollen kontinuerlig passordene i brukerens nøkkelring for autoutfylling av passord mot den modererte listen.

Slik fungerer overvåking

Brukerens enhet gjennomfører kontinuerlig ringdistribusjonskontroller av brukerens passord, og sender forespørsler med et intervall som er uavhengig av brukerens passord. Dette bidrar til å sørge for at verifiseringstilstander forblir oppdatert med den gjeldende modererte listen med lekkede passord. For å bidra til å forhindre lekkasje av informasjon om hvor mange unike passord en bruker har, utføres forespørslene gruppevis og parallelt. Et fast antall passord verifiseres parallelt i hver kontroll, og hvis brukeren har færre enn dette tallet, genereres det tilfeldige passord som legges til i forespørslene for å utgjøre forskjellen.

Slik sammenlignes passord

Passord sammenlignes i en todelt prosess. Passordene som er mest vanlig å lekke, ligger i en lokal liste på brukerens enhet. Hvis brukerens passord dukker opp på listen, varsles brukeren umiddelbart uten noe ekstern interaksjon. Dette er utviklet for å sikre at det ikke lekkes noe informasjon om passordene til en bruker som er mest utsatt for passordlekkasje.

Hvis passordet ikke ligger i listen med passord som lekkes oftest, sammenlignes det med passord som lekkes sjeldnere.

Sammenligning av brukernes passord mot en moderert liste

Å verifisere om et passord som ikke er i den lokale listen, samsvarer, involverer noe interaksjon med Apple-tjenere. For å bidra til å sikre at legitime brukeres passord ikke sendes til Apple, rulles det ut en form for kryptografisk *private set intersection* som sammenligner brukernes passord mot et stort sett med lekkede passord. Dette er utviklet for å sørge for at lite informasjon deles med Apple for passord med mindre risiko for lekkasje. For en brukers passord er denne informasjonen begrenset til et 15-bit-prefiks av en kryptografisk hash. Hvis passordene som lekkes oftest, fjernes fra denne interaktive prosessen ved bruk av den lokale listen med passord som lekkes oftest, reduseres deltaverdien i relativ frekvens av passord i nettjenestesamlingene, noe som gjør det vanskelig å utlede brukerpassord fra disse søkene.

Den underliggende protokollen deler opp listen med modererte passord, som inneholdt omtrent 1,5 milliarder passord da dette ble skrevet, i 2^{15} forskjellige samlinger. Samlingen der et passord hører til, er basert på første 15 bit av SHA256-hashverdien til passordet. I tillegg er hvert lekket passord, pw, forbundet med et elliptisk kurvepunkt på NIST P256-kurven: $P_{pw} = \alpha \cdot H_{SWU}(pw)$, der α er en hemmelig, tilfeldig nøkkel som kun Apple kjenner til, og H_{SWU} er en tilfeldig orakelfunksjon som tilordner passord til kurvepunkter basert på Shallue-van de Woestijne-Ulas-metoden. Denne transformasjonen er designet for å skjule verdiene til passord kalkulert, og den bidrar til å hindre avsløring av nylig lekkede passord via Passordovervåking.

For å beregne «private set intersection» fastslår brukerens enhet samlingen som brukerens passord hører til, ved å bruke λ, 15-bit-prefikset til SHA256(upw), der upw er ett av brukerens passord. Enheten genererer sin egen tilfeldige konstant, β, og sender punktet $P_c = \beta \cdot H_{SWU}(upw)$ til tjeneren, sammen med en forespørsel om samlingen som tilsvarer λ. Her skjuler β informasjon om brukerens passord og begrenser for λ informasjonen som eksponeres fra passordet til Apple. Til slutt tar tjeneren punktet sendt av brukerens enhet, beregner $\alpha P_c = \alpha\beta \cdot H_{SWU}(upw)$, og returnerer det, sammen med den relevante samlingen med punkter – $B\lambda = \{ P_{pw} \mid \text{SHA256(pw) begynner med prefiks } \lambda\}$ – til enheten.

Den returnerte informasjonen tillater enheten å beregne $B'_\lambda = \{\beta \cdot P_{pw} \mid P_{pw} \in B_\lambda\}$, og fastslår at brukerens passord har blitt lekket hvis $\alpha P_c \in B'_\lambda$.

Sende passord til andre brukere eller Apple-enheter

Arkivere akkreditiver på en annen enhet med AirDrop

Når iCloud er aktivert, kan brukere via AirDrop sende et arkivert akkreditiv til en annen enhet, inkludert nettsidene det er arkivert for, brukernavnet og passordet. Sending av akkreditiver med AirDrop fungerer alltid i Kun kontakter-modus, uavhengig av brukerens innstillinger. På mottakerenheten, etter brukergodkjenning, lagres akkreditivet i brukerens nøkkelring for autoutfylling av passord.

Fylle ut akkreditiver i apper på Apple TV

Autoutfylling av passord er tilgjengelig for å fylle ut akkreditiver i apper på Apple TV. Når brukeren fokuserer på et brukernavn- eller passordtekstfelt i tvOS, begynner Apple TV å annonsere en forespørsel for autoutfylling av passord over Bluetooth Low Energy (BLE).

En iPhone-, iPad- eller iPod touch-enhet i nærheten viser et varsel som inviterer brukeren til å dele et akkreditiv med Apple TV. Slik etableres krypteringsmetoden:

- Hvis enheten og Apple TV bruker samme iCloud-konto, skjer kryptering mellom enhetene automatisk.
- Hvis enheten er logget på en annen iCloud-konto enn den som brukes av Apple TV, bes brukeren om å etablere en kryptert forbindelse via bruk av en PIN-kode. iPhonen må være ulåst og i nærheten av Siri Remote som er sammenkoblet med den Apple TV-enheten for å motta dette varselet.

Etter at den krypterte forbindelsen opprettes ved hjelp BLE-forbindelseskryptering, sendes akkreditivet til Apple TV og angis automatisk i de relevante tekstfeltene i appen.

Tillegg for akkreditiver

I iOS, iPadOS og macOS kan brukere angi en deltakende tredjepartsapp som en akkreditivleverandør for autoutfylling av passord i Passord-innstillinger (iOS og iPadOS) eller i Tillegg-innstillinger i Systemvalg (macOS). Denne mekanismen er bygget på apptillegg. Tillegget for akkreditivleverandør må tilby en visning for å velge akkreditiver, og tillegget kan også tilby metadata om arkiverte akkreditiver, slik at de kan være tilgjengelige direkte fra QuickType-linjen (iOS og iPadOS) eller i et autoutfyllingsforslag (macOS). Metadataene inkluderer nettstedet for akkreditivet og det tilknyttede brukernavnet, men ikke passordet. iOS, iPadOS og macOS kommuniserer med tillegget for å få passordet når brukeren velger å fylle et akkreditiv inn i en app eller på et nettsted i Safari. Akkreditivmetadata oppbevares i akkreditivleverandørens appbeholder og fjernes automatisk når en app avinstalleres.

iCloud-nøkkelring

Oversikt over sikkerhet for iCloud-nøkkelring

Med iCloud kan brukerne synkronisere passordene på en sikker måte mellom iOS- og iPadOS-enheter og Macer uten at informasjonen blir tilgjengelig for Apple. I tillegg til personvern og sikkerhet på høyt nivå var brukervennlighet og muligheten til å gjenopprette en nøkkelring viktige faktorer i utformingen av og arkitekturen til iCloud-nøkkelringen, som består av to tjenester: nøkkelringsynkronisering og nøkkelringgjenoppretting.

Apple utformet iCloud-nøkkelringen og nøkkelringgjenopprettingen slik at brukerens passord fortsatt er beskyttet i følgende situasjoner:

- brukerens iCloud-konto kompromitteres
- iCloud kompromitteres av en ekstern angriper eller ansatt
- en tredjepart får tilgang til brukerkontoer

integrering av passordadministrator med iCloud-nøkkelring

iOS, iPadOS og macOS kan automatisk generere kryptografisk sterke, tilfeldige strenger som brukes som kontopassord i Safari. iOS og iPadOS kan også generere sterke passord for apper. Genererte passord oppbevares i nøkkelringen og synkroniseres til andre enheter. Nøkkelringobjekter overføres fra en enhet til en annen og transporten går via Apple-tjenere, men de krypteres på en slik måte at Apple eller andre enheter ikke kan lese innholdet.

Sikker nøkkelringsynkronisering

Når en bruker aktiverer iCloud-nøkkelring for første gang, danner enheten en godkjent sirkel og oppretter en synkroniseringsidentitet for seg selv. Synkroniseringsidentiteten består av en privat nøkkel og en offentlig nøkkel. Den offentlige nøkkelen til synkroniseringsidentiteten legges i sirkelen, og sirkelen signeres to ganger: først av den private nøkkelen til synkroniseringsidentiteten og så en gang til med en asymmetrisk elliptisk nøkkel (som bruker P-256) avledet fra brukerens passord for iCloud-kontoen. Parameterne (tilfeldig salt og iterasjoner) som brukes til å lage nøkkelen som er basert på brukerens iCloud-passord, lagres også i sirkelen.

Sirkelen for synkronisering plasseres i brukerens iCloud

Den signerte sirkelen for synkronisering plasseres i brukerens iCloud-lagringsplass for nøkkelverdier. Den kan ikke leses uten å kjenne brukerens iCloud-passord og kan ikke endres på en gyldig måte uten å ha den private nøkkelen til synkroniseringsidentiteten til sirkelens medlem.

Når brukeren slår på iCloud-nøkkelringen på en annen enhet, oppfatter iCloud-nøkkelring at brukeren har en allerede etablert synkroniseringssirkel i iCloud som den ikke er medlem av. Enheten lager sitt nøkkelpar for synkroniseringsidentitet og lager deretter en applikasjonsbillett som ber om medlemskap i sirkelen. Billetten består av enhetens offentlige nøkkel for synkroniseringsidentiteten, og brukeren bes om å oppgi iCloud-passordet for å bli autentisert. Parameterne for generering av den elliptiske nøkkelen hentes fra iCloud og genererer en nøkkel som brukes til å signere applikasjonsbilletten. Til slutt plasseres applikasjonsbilletten i iCloud.

Hvordan en brukers andre enheter legges til i synkroniseringssirkelen

Når den første enheten ser at en applikasjonsbillett har kommet, ber den brukeren om å anerkjenne at en ny enhet ber om å få bli med i synkroniseringssirkelen. Brukeren oppgir iCloud-passordet, og applikasjonsbilletten blir bekreftet som signert av en tilhørende privat nøkkel. Nå kan brukerne som genererte forespørselen om å bli med i sirkelen, bli med i den.

Når brukeren godkjenner at den nye enheten kan legges til i sirkelen, legger den første enheten til det nye medlemmets offentlige nøkkel i synkroniseringssirkelen, og signerer den på nytt med både synkroniseringsidentiteten og nøkkelen avledet fra brukerens iCloud-passord. Den nye synkroniseringssirkelen plasseres i iCloud, hvor den signeres på tilsvarende måte av det nye medlemmet i sirkelen.

Det er nå to medlemmer i signeringssirkelen, og begge medlemmene har den offentlige nøkkelen til det andre medlemmet. De begynner så å utveksle individuelle nøkkelringobjekter via iCloud-lagringsplassen for nøkkelverdier eller lagre dem i CloudKit, avhengig av hva som er mest passende for situasjonen. Hvis begge medlemmene i sirkelen har det samme objektet, er det objektet med den nyeste endringsdatoen som blir synkronisert. Objekter utelates hvis det andre medlemmet har objektet og endringsdatoen

er den samme. Hvert objekt som synkroniseres, krypteres, slik at det kun kan dekrypteres av en enhet innenfor brukerens godkjente sirkel. Det kan ikke dekrypteres av andre enheter eller av Apple.

Denne prosessen gjentas for nye enheter som legges inn i sirkelen. Hvis for eksempel en tredje enhet blir lagt til, vises bekreftelsen på begge de andre enhetene til brukeren. Brukeren kan godkjenne det nye medlemmet fra begge disse enhetene. Etter hvert som det legges til nye enheter på samme nivå, synkroniseres alle enhetene med den nye. Dette er utviklet for å sikre at alle medlemmene har samme nøkkelringobjekter.

Kun enkelte objekter synkroniseres

Hele nøkkelringen synkroniseres imidlertid ikke. Noen objekter er enhetsspesifikke, for eksempel VPN-identiteter, og bør ikke forlate enheten. Kun objekter med attributtet `kSecAttrSynchronizable` blir synkronisert. Apple har satt dette attributtet for Safari-brukerdata (inkludert brukernavn, passord og kredittkortnumre) og for Wi-Fi-passord og HomeKit-krypteringsnøkler.

Dessuten er det standard at nøkkelringobjekter som legges til av tredjepartsapper, ikke synkroniseres. Utviklere må sette attributtet `kSecAttrSynchronizable` når de legger til objekter på nøkkelringen.

Sikker iCloud-nøkkelringgjenoppretting

iCloud-nøkkelring deponerer brukerens nøkkelring hos Apple *uten* at Apple kan lese passordene eller andre data den inneholder. Selv om brukeren bare har én enhet, gir nøkkelringgjenoppretting en sikkerhet mot datatap. Dette er særskilt viktig når Safari brukes til å generere tilfeldige, sterke passord for nettkontoer, fordi disse passordene kun registreres i nøkkelringen.

Et sentralt punkt ved nøkkelringgjenoppretting er sekundær autentisering og en sikker deponeringstjeneste, som Apple har spesialutviklet for denne funksjonen. Brukerens nøkkelring krypteres med en sterk kode, og deponeringstjenesten stiller strenge betingelser for å utlevere en kopi av nøkkelringen.

Bruk av sekundær autentisering

Det er flere måter å etablere en sterk kode på:

- Hvis tofaktorautentisering er aktivert for brukerens konto, brukes enhetens kode til å gjenopprette en deponert nøkkelring.
- Hvis tofaktorautentisering ikke er konfigurert, blir brukeren bedt om å opprette en iCloud-sikkerhetskode ved å angi en sekssifret kode. Uten tofaktorautentisering kan brukere imidlertid alternativt angi en egen lengre kode, eller de kan la enhetene lage en tilfeldig kryptografisk kode som de kan notere og ta vare på selv.

Prosessen for nøkkelringdeponering

Når koden er opprettet, deponeres nøkkelringen hos Apple. iOS iPadOS- eller macOS-enheten eksporterer først en kopi av brukerens nøkkelring, krypterer den deretter innpakket med nøkler i et asymmetrisk nøkkeletui og plasserer den i brukerens iCloud-lagringsplass for nøkkelverdi. Nøkkeletuiet pakkes med brukerens iCloud-sikkerhetskode og med den offentlige nøkkelen til klyngen med maskinvaresikkerhetsmodulen (HSM) som lagrer deponeringsoppføringen. Dette blir brukerens iCloud-deponeringsoppføring. For

HSA2-kontoer lagres også nøkkelringen i CloudKit og pakkes som mellomliggende nøkler som kun kan gjenopprettes med innholdet i iCloud-deponeringsoppføringen, for å gi samme beskyttelsesnivå.

Merk: Hvis brukeren bestemmer seg for å godta en tilfeldig kryptografisk sikkerhetskode i stedet for å angi en egen firesifret verdi, er det ikke nødvendig med en deponeringsoppføring. I stedet brukes iCloud-sikkerhetskoden til å pakke den tilfeldige nøkkelen direkte.

I tillegg til å opprette en sikkerhetskode, må brukere registrere et telefonnummer. Det brukes som ekstra autentiseringsnivå under nøkkelringgjenoppretting. Brukeren mottar en tekstmelding som må besvares for at gjenopprettingen skal utføres.

Deponeringssikkerhet for iCloud-nøkkelring

iCloud sørger for en sikker infrastruktur for nøkkelringdeponering for å bidra til å sikre at kun godkjente brukere og enheter kan utføre en gjenoppretting. Klynger med maskinvaresikkerhetsmoduler (HSM-er) som vokter deponeringsoppføringene, er topografisk plassert bak iCloud. Som tidligere beskrevet, har de en nøkkel hver som brukes til å kryptere deponeringsoppføringene de vokter.

For å gjenopprette en nøkkelring må brukerne autentisere med iCloud-kontoen og svare på en tekstmelding som sendes til det registrerte telefonnummeret. Etter at dette er gjort, må brukerne oppgi iCloud-sikkerhetskoden. HSM-klyngen bekrefter at en bruker kjenner iCloud-sikkerhetskoden ved hjelp av SRP-protokollen (Secure Remote Password). Selve koden sendes ikke til Apple. Medlemmene i klyngen bekrefter uavhengig av hverandre at brukeren ikke har overskredet maksimalt antall forsøk som er tillatt for å hente oppføringen, slik det er beskrevet nedenfor. Hvis flertallet er enig, pakker klyngen ut deponeringsoppføringen og sender den til brukerens enhet.

Deretter bruker enheten iCloud-sikkerhetskoden til å pakke ut de tilfeldige nøklene som ble brukt til å kryptere brukerens nøkkelring. Med den nøkkelen blir nøkkelringen, som hentes fra iCloud-lagringsplass for nøkkelverdier og CloudKit, dekryptert og gjenopprettet på enheten. iOS, iPadOS og macOS tillater kun 10 forsøk på å autentisere og hente en deponeringsoppføring. Etter flere mislykkede forsøk låses oppføringen, og brukeren må ringe Apple-kundestøtten for å få lov til å gjøre flere forsøk. Etter det tiende mislykkede forsøket ødelegger HSM-klyngen deponeringsoppføringen, og nøkkelringen er tapt for alltid. Dette gir beskyttelse mot brute-force-forsøk på å hente oppføringen, men til gjengjeld må nøkkelringdata ofres.

Disse retningslinjene ligger kodet i HSM-firmwaren. De administrative tilgangskortene som gir firmware lov til å bli endret, er blitt ødelagt. Alle forsøk på å endre firmwaren eller å få tilgang til den private nøkkelen fører til at HSM-klyngen sletter den private nøkkelen. Hvis dette skjer, får eieren av hver nøkkelring som er beskyttet av klyngen, en melding som informerer dem om at deponeringsoppføringene deres har gått tapt. De kan da velge å registreres på nytt.

Apple Pay

Oversikt over Apple Pay-sikkerhet

Med Apple Pay kan brukerne benytte støttede iPhone-, iPad-, Mac- og Apple Watch-enheter til å betale på en enkel, sikker og privat måte i butikker, apper og på nettet i Safari. Brukere kan også legge til Apple Pay-aktiverte reisekort og studentbevis i Apple Wallet. Det er enkelt for brukerne, og sikkerhet er integrert i både maskinvare og programvare.

Apple Pay er også designet for å beskytte brukerens personopplysninger. Apple Pay henter ikke inn informasjon om transaksjonen som kan knyttes til brukeren. Betalingstransaksjoner foregår mellom brukeren, forhandleren og utstederen av kortet.

Sikkerhet for Apple Pay-komponenter

Secure Element

Secure Element er en sertifisert brikke som er bransjestandard og kjører Java Card-plattformen. Den oppfyller finansbransjens krav til elektroniske betalinger. Secure Element-IC-en og Java-kortplattformen er sertifisert i samsvar med EMVCos prosess for sikkerhetsevaluering. Når sikkerhetsevalueringen er gjennomført, utsteder EMVCo en unik IC og et unikt plattformsertifikat.

Secure Element-IC-en er sertifisert basert på Common Criteria-standarden.

NFC-kontroller

NFC-kontrolleren håndterer Near Field Communication-protokoller og ruter kommunikasjon mellom applikasjonsprosessoren og Secure Element og mellom Secure Element og terminalen på utsalgsstedet.

Apple Wallet

Apple Wallet brukes til å legge til og administrere kreditt- og debetkort samt butikkutstedte kort og til å foreta betalinger med Apple Pay. I Apple Wallet kan brukerne se hvilke kort de har, og i noen tilfeller vise mer informasjon fra utstederen av kortet, for eksempel utstederens retningslinjer for personvern, siste transaksjoner og mer. Brukerne kan også legge til kort i Apple Pay i:

- Oppsettassistent og Innstillinger for iOS og iPadOS
- Watch-appen for Apple Watch
- Wallet og Apple Pay i Systemvalg for Macer med Touch ID

I tillegg gjør Apple Wallet det mulig for brukere å legge til og administrere reisekort, bonuskort, boardingkort, billetter, gavekort, studentbevis og mer.

Secure Enclave

På iPhone, iPad, Apple Watch og Macer med Touch ID administrerer Secure Enclave autentiseringsprosessen, slik at betalingstransaksjoner kan gjennomføres.

På Apple Watch må enheten være låst opp, og brukeren må dobbelttrykke på sideknappen. Dobbelttrykket fanges opp og blir sendt videre direkte til Secure Element, eller Secure Enclave hvis tilgjengelig, uten å gå veien om applikasjonsprosessoren.

Apple Pay-tjenere

Apple Pay-tjenerne administrerer oppsett og klargjøring av kredittkort, debetkort, reisekort og studentbevis i Wallet-appen. Tjenerne administrerer også enhetskontonumrene som er lagret i Secure Element. De kommuniserer både med enheten og med tjenerne for betalingsnettverket eller kortutstederen. Apple Pay-tjenerne er også ansvarlige for å kryptere betalingsakkreditiver på nytt når det skal foretas betaling i apper eller på internett.

Hvordan Apple Pay bruker Secure Element NFC-kontrolleren

Secure Element

Secure Element er vert for en applet som er laget spesielt for å administrere Apple Pay. Det inkluderer også appleter sertifisert av betalingsnettverk eller kortutstedere. Data fra kreditt- og debetkort samt forhåndsbetalte kort sendes kryptert fra betalingsnettverket eller kortutstederen til disse appletene ved hjelp av nøkler som kun betalingsnettverket eller kortutstederen og appletenes sikkerhetsdomene kjenner til. Dataene lagres i disse appletene og beskyttes ved hjelp av sikkerhetsfunksjonene i Secure Element. Når en transaksjon pågår, kommuniserer terminalen direkte med Secure Element gjennom NFC-kontrolleren (Near Field Communication) via en egen maskinvarebuss.

NFC-kontroller

NFC-kontrolleren er portalen til Secure Element og bidrar til å sikre at alle transaksjoner forbundet med kontaktløs betaling utføres ved hjelp av en terminal på utsalgsstedet som befinner seg i nærheten av enheten. Kun betalingsforespørsler fra en utplassert terminal merkes av NFC-kontrolleren som kontaktløse transaksjoner.

Etter at en betaling fra et kreditt-, debet- eller forhåndsbetalt kort (inkludert butikkort) er godkjent av korteieren ved hjelp av Touch ID, Face ID eller koden, eller på en ulåst Apple Watch ved å dobbelttrykke på sideknappen, ruter kontrolleren kontaktløse svar som betalings-appleten har forberedt i Secure Element, kun til NFC-feltet. Derfor holdes informasjon om betalingsgodkjenning for kontaktløse betalingstransaksjoner kun i det lokale NFC-feltet og blir aldri lagt åpen for applikasjonsprosessoren. Informasjon om betalingsgodkjenning i apper rutes derimot til applikasjonsprosessoren, men kun etter å ha blitt kryptert av Apple Pay-tjenerens Secure Element.

Kredittkort, debetkort og forhåndsbetalte kort

Oversikt over sikkerhet for klargjøring av kort

Når en bruker legger til et kredittkort, debetkort eller forhåndsbetalt kort (også butikkutstedte kort) i Apple Wallet, sender Apple kortopplysningene, samt annen informasjon om brukerens konto og enhet, til kortutstederen eller kortutstederens godkjente tjenesteleverandør på en sikker måte. Ved hjelp av denne informasjonen avgjør kortutstederen om den skal godkjenne at kortet legges til i Apple Wallet.

Som en del av klargjøringsprosessen for kort bruker Apple Pay tre tjenersidekall til å sende og motta kommunikasjon med kortutstederen eller nettverket: Required Fields, Check Card og Link and Provision. Kortutstederen eller nettverket bruker disse anropene til å bekrefte, godkjenne og legge til kort i Apple Wallet. Disse klient-tjener-øktene bruker TLS 1.2 til å overføre dataene.

Fullstendige kortnumre lagres ikke på enheten eller på Apple Pay-tjenere. I stedet lages det et unikt enhetskontonummer som blir kryptert og så lagret i Secure Element. Det unike enhetskontonummeret krypteres på en slik måte at Apple ikke får tilgang til det. Enhetskontonummeret er unikt og skiller seg fra de fleste betalingskortnumre ved at kortutstederen eller betalingsnettverket kan hindre at det brukes på et kort med magnetstripe, via telefonen eller på nettsider. Enhetskontonummeret i Secure Element lagres aldri på Apple Pay-tjenere eller sikkerhetskopieres til iCloud, og det er isolert fra iOS-, iPadOS- og watchOS-enheter med Touch ID.

Kort som skal brukes med Apple Watch, klargjøres for Apple Pay ved hjelp av Apple Watch-appen på iPhone eller i en kortutsteders iPhone-app. Hvis du vil legge til et kort på Apple Watch, kreves det at klokken må være innenfor rekkevidden til Bluetooth-kommunikasjon. Kort registreres spesielt for bruk med Apple Watch og har sine egne enhetskontonumre, som lagres i Secure Element på Apple Watch.

Når kredittkort, debetkort eller forhåndsbetalte kort (inkludert butikkort) legges til, vises de i en liste med kort under oppsettassistenten på enheter som er logget på samme iCloud-konto. Disse kortene forblir i denne listen så lenge de er aktive på minst én enhet. Kort fjernes fra denne listen etter at de har vært fjernet fra alle enheter i sju dager. Denne funksjonen krever at tofaktorautentisering er aktivert på den respektive iCloud-kontoen.

Legge til kreditt- eller debetkort i Apple Pay

Legge til kreditt- eller debetkort manuelt

Hvis du vil legge til et kort manuelt, brukes navnet, kortnummeret, utløpsdatoen og CVV for å legge til rette for klargjøringsprosessen. I Innstillinger, Wallet-appen eller Apple Watch-appen kan brukerne oppgi denne informasjonen ved å skrive den inn eller ved å bruke enhetens kamera. Når kameraet fanger opp kortinformasjonen, forsøker Apple å fylle ut navn, kortnummer og utløpsdato. Bildet arkiveres aldri på enheten eller i bildebiblioteket. Etter at alle feltene er fylt ut, kontrollerer Check Card-prosessen alle feltene unntatt CVV. De krypteres deretter og sendes til Apple Pay-tjeneren.

Hvis Check Card-prosessen returnerer en ID for vilkår og betingelser, laster Apple ned kortutstederens vilkår og betingelser og viser dem til brukeren. Hvis brukeren godtar vilkår og betingelser, sender Apple ID-en for de godtatte vilkårene, samt CVV, til Link and Provision-prosessen. Som en del av Link and Provision-prosessen deler dessuten Apple informasjon fra enheten med kortutstederen eller nettverket. Det kan være informasjon om aktiviteten på iTunes- og App Store-kontoen (for eksempel om brukeren har en lang historikk med transaksjoner i iTunes), informasjon om enheten (for eksempel telefonnummer, navn og enhetsmodell, pluss eventuell Apple-ledsagerenhet som er nødvendig for å konfigurere Apple Pay) og omtrent hvor brukeren befinner seg når vedkommende legger til kortet (hvis Stedstjenester er aktivert). Ved hjelp av denne informasjonen avgjør kortutstederen om den skal godkjenne at kortet legges til i Apple Pay.

Resultatet av Link and Provision-prosessen er at det skjer to ting:

- enheten begynner å laste ned Wallet-adgangsfilen som representerer kreditt- eller debetkortet.
- enheten begynner å binde kortet til Secure Element.

Adgangsfilen inneholder URL-er til nedlastingskortgrafikk, metadata om kortet, for eksempel kontaktinformasjon, utstederens app og støttede funksjoner. Den inneholder også adgangsstatusen, som for eksempel kan være om den personlige tilpasningen av Secure Element er blitt fullført, om kortet er midlertidig sperret av kortutstederen eller om ytterligere bekreftelse er nødvendig før kortet kan foreta betalinger med Apple Pay.

Legge til kreditt- eller debetkort som er registrert på en iTunes Store-konto

Det kan hende at brukeren må oppgi Apple-ID-passordet på nytt for et kreditt- eller debetkort som er lagt inn for iTunes. Kortnummeret hentes fra iTunes, og Check Card-prosessen igangsettes. Hvis kortet er kvalifisert for Apple Pay, laster enheten ned vilkår og betingelser og viser dem og sender deretter vilkårenes ID og kortets sikkerhetskode til Link and Provision-prosessen. Det kan hende at ytterligere bekreftelse foretas for kort som ligger på iTunes-kontoer.

Legge til kreditt- eller debetkort fra appen til en kortutsteder

Når en app er registrert for bruk med Apple Pay, dannes det nøkler for appen og for kortutstederens tjener. Disse nøklene brukes til å kryptere kortinformasjonen som sendes til kortutstederen. Dette er utviklet for å hindre at informasjonen leses av Apple-enheten. Klargjøringsflyten er lik den som brukes for å legge til kort manuelt, beskrevet tidligere, med unntak av at engangspassord brukes i stedet for CVV.

Legge til ytterligere bekreftelse

En kortutsteder kan avgjøre om kreditt- eller debetkortet krever ytterligere bekreftelse. Avhengig av hva kortutstederen tilbyr, kan det hende at brukeren kan velge mellom ulike valg for ytterligere bekreftelse, for eksempel en tekstmelding, e-post, telefon fra kundeservice eller en metode i en godkjent tredjepartsapp, for å fullføre bekreftelsen. Når det gjelder tekstmeldinger og e-post velger brukeren fra kontaktinformasjon som utstederen har lagret. Det blir sendt en kode, som må oppgis i Wallet-appen, Innstillinger eller Apple Watch-appen. Når det gjelder kundeservice eller bekreftelse ved hjelp av en app, utfører utstederen sin egen kommunikasjonsprosess.

Betalingsgodkjenning med Apple Pay

For enheter med Secure Enclave kan en betaling kun utføres etter å ha mottatt godkjenning fra Secure Enclave. På iPhone eller iPad innebærer det å få bekreftet at brukeren er blitt autentisert ved hjelp av Touch ID, Face ID eller ved å ha oppgitt koden på enheten. Touch ID eller Face ID er standardmetoden hvis tilgjengelig, men koden kan brukes når som helst. Du gis automatisk muligheten til å oppgi koden etter at det er gjort tre mislykkede forsøk med fingeravtrykk eller to mislykkede forsøk med ansiktsgjenkjenning. Etter fem mislykkede forsøk, er koden påkrevd. Koden er også påkrevd hvis Touch ID eller Face ID ikke er konfigurert eller ikke er aktivert for Apple Pay. For at en betaling skal kunne utføres på Apple Watch, må enheten være låst opp med koden og det må dobbelttrykkes på sideknappen.

Bruke en delt sammenkoblingsnøkkel

Kommunikasjonen mellom Secure Enclave og Secure Element finner sted via et serielt grensesnitt, der Secure Element er koblet til NFC-kontrolleren, som i sin tur er koblet til applikasjonsprosessoren. Selv om de ikke er koblet direkte til hverandre, kan Secure Enclave og Secure Element kommunisere på en sikker måte ved hjelp av en delt sammenkoblingsnøkkel som klargjøres i produksjonsprosessen. Krypteringen og autentiseringen av kommunikasjonen er basert på AES, der begge sider bruker kryptografiske «nonce»-verdier (tilfeldige engangskoder) som beskyttelse mot repetisjonsangrep. Sammenkoblingsnøkkelen genereres i Secure Enclave fra UID-nøkkelen og den unike ID-en til Secure Element. Sammenkoblingsnøkkelen overføres deretter på en sikker måte fra Secure Enclave til en maskinvaresikkerhetsmodul (HSM) i fabrikken, som har nøkkelmaterialet som kreves for så å legge inn sammenkoblingsnøkkelen i Secure Element.

Godkjenne en sikker transaksjon

Når brukeren godkjenner en transaksjon, som inkluderer en fysisk handling som kommuniseres direkte til Secure Enclave, sender Secure Enclave deretter signerte data om type autentisering og informasjon om type transaksjon (kontaktløs eller i apper) til Secure Element, knyttet til en tilfeldig godkjenningsverdi (AR-verdi). AR-verdien genereres i Secure Enclave første gang en bruker klargjør et kredittkort og opprettholdes mens Apple Pay aktiveres, beskyttet av krypteringen og anti-tilbakerullingsmekanismen i Secure Enclave. Den leveres på en sikker måte til Secure Element ved hjelp av sammenkoblingsnøkkelen. Når Secure Element mottar en ny AR-verdi, merker det eventuelle kort som er lagt til tidligere, som slettet.

Bruke et betalingskryptogram for dynamisk sikkerhet

Betalingstransaksjoner som har sitt utspring i betalings-appleten, inneholder et betalingskryptogram sammen med et enhetskontonummer. Dette kryptogrammet, en engangskode, beregnes ved hjelp av en transaksjonsteller og en nøkkel. Transaksjonstelleren økes trinnvis for hver nye transaksjon. Nøkkelen klargjøres i betalings-appleten under den personlige tilpasningen og betalingsnettverket og/eller kortutstederen kjenner den. Avhengig av hvilken betalingsform som brukes, kan det hende at også andre data brukes i beregningen, blant annet:

- Et Terminal Unpredictable Number for NFC-transaksjoner (Near-Field-Communication)
- En nonce-verdi for Apple Pay-tjener, for transaksjoner i apper

Disse sikkerhetskodene oppgis til betalingsnettverket og kortutstederen og gjør utstederen i stand til å bekrefte hver enkelt transaksjon. Lengden på disse sikkerhetskodene kan variere basert på transaksjonstypen.

Betale med kort med Apple Pay

Betale med kort i butikker

Hvis iPhone eller Apple Watch er slått på og fanger opp et NFC-felt, legger den fram det forespurte kortet (hvis automatisk valg er slått på for det kortet), eller standardkortet, for brukeren. Dette angis i Innstillinger. Brukeren kan også gå til Wallet-appen og velge et kort, eller når enheten er låst:

- Dobbelttrykk på Hjem-knappen på enheter med Touch ID
- Dobbelttrykk på sideknappen på enheter med Face ID

Deretter, før informasjon overføres, må brukeren autentiseres ved hjelp av Touch ID, Face ID eller ved å oppgi koden. Når Apple Watch er ulåst, aktiveres standardkortet for betaling ved å dobbelttrykke på knappen på siden. Ingen betalingsinformasjon blir sendt uten at brukeren er godkjent.

Etter at brukeren er godkjent, brukes enhetskontonummeret og en transaksjonsspesifikk dynamisk sikkerhetskode til å behandle betalingen. Verken Apple eller brukerens enhet sender fullstendige betalingskortnumre til forhandlere. Det kan hende Apple mottar anonym informasjon om transaksjonen, som omtrentlig tidspunkt og sted for transaksjonen, noe som hjelper med å forbedre Apple Pay og andre Apple-produkter og -tjenester.

Betale med kort i apper

Apple Pay kan også brukes til å utføre betalinger i apper på iPhone, iPad, Mac og Apple Watch. Når brukere betaler i apper ved hjelp av Apple Pay, mottar Apple den krypterte transaksjonsinformasjonen. Før informasjonen sendes til utvikleren eller forhandleren, rekrypterer Apple transaksjonen med en utviklerspesifikk nøkkel. Apple Pay beholder anonym informasjon om transaksjonen, deriblant omtrentlig kjøpsbeløp. Denne informasjonen kan ikke knyttes til brukeren, og den inkluderer aldri informasjon om hva brukeren kjøper.

Når en app setter i gang en Apple Pay-betalingstransaksjon, mottar Apple Pay-tjenerne den krypterte transaksjonen fra enheten før forhandleren. Apple Pay-tjenerne krypterer deretter transaksjonen på nytt med en forhandlerspesifikk nøkkel før de sender den til forhandleren.

Når en app ber om en betaling, sender den spørringer til en API for å avgjøre om enheten støtter Apple Pay og om brukeren har kreditt- eller debetkort som kan foreta betalinger på et betalingsnettverk som forhandleren godtar. Appen ber om alt den trenger av informasjon for å behandle og fullføre transaksjonen, for eksempel faktura- og leveringsadresse og kontaktinformasjon. Appen ber deretter iOS, iPadOS eller watchOS om å vise Apple Pay-vinduet, som ber om informasjon for appen, og i tillegg annen nødvendig informasjon eksempel hvilket kort som skal brukes.

På dette tidspunktet oppgis informasjon om postnummer og -sted til appen, slik at den kan beregne de endelige fraktkostnadene. Det fullstendige settet med informasjon oppgis ikke til appen før brukeren godkjenner betalingen med Touch ID, Face ID eller koden på enheten. Etter at betalingen er godkjent, blir informasjonen som vises i Apple Pay-vinduet, overført til forhandleren.

Betalingsgodkjenning i app

Når brukeren godkjenner betalingen, sendes et anrop til Apple Pay-tjenerne for å få tak i en kryptografisk «nonce»-verdi, som ligner verdien som returneres av NFC-terminalen som brukes for transaksjoner i butikk. «Nonce»-verdien sendes videre sammen med andre transaksjonsdata til Secure Element for å generere et betalingsakkreditiv som krypteres med en Apple-nøkkel. Når det krypterte betalingsakkreditivet kommer fra Secure Element, sendes det videre til Apple Pay-tjenerne, som dekrypterer akkreditivet, bekrefter «nonce»-verdien i akkreditivet mot «nonce»-verdien som opprinnelig ble sendt av Apple Pay-tjenerne, og krypterer betalingsakkreditivet på nytt med forhandlernøkkelen som er knyttet til forhandler-ID-en. Betalingen sendes så tilbake til enheten, som gir den tilbake

til appen via API-en. Appen sender den så videre til forhandlersystemet for behandling. Forhandleren kan deretter dekryptere betalingsakkreditivet med sin private nøkkel slik at det kan behandles. Sammen med signaturen fra Apples tjenere gjør dette det mulig for forhandleren å bekrefte at transaksjonen var ment for akkurat denne forhandleren.

API-ene krever en rettighet som angir ID-ene til de støttede forhandlerne. En app kan også inkludere ytterligere data som skal sendes til Secure Element for signering, for eksempel ordrenummer eller kundeidentitet, noe som sikrer at transaksjonen ikke blir omdirigert til en annen kunde. Dette gjennomføres av apputvikleren, som kan angi applicationData på PKPaymentRequest. En hash av disse dataene er inkludert i den krypterte betalingsinformasjonen. Deretter er det forhandlernes ansvar å bekrefte at deres applicationData-hash samsvarer med den som ligger i betalingsinformasjonen.

Betale med kort på nettsteder

Apple Pay kan brukes til å utføre betalinger på nettsteder på iPhone, iPad, Apple Watch og Macer med Touch ID. Apple Pay-transaksjoner kan også startes på Mac og fullføres på en Apple Pay-aktivert iPhone eller Apple Watch som bruker den samme iCloud-kontoen.

Apple Pay på nettet krever at alle nettsteder som deltar, registrerer seg hos Apple. Etter at domenet er registrert, utføres validering av domenenavnet kun etter at Apple utsteder et TLS-klientsertifikat. Det er et krav for nettsteder som støtter Apple Pay, at de overfører innholdet via HTTPS. For hver enkelt betalingstransaksjon må nettstedet opprette en sikker og unik forhandlerøkt på en Apple-tjener ved hjelp av TLS-klientsertifikatet Apple har utstedt. Forhandlerøktdata signeres av Apple. Etter at en forhandlerøktsignatur er bekreftet, kan et nettsted spørre om brukeren har en enhet med Apple Pay og om et kredittkort, debetkort eller forhåndsbetalt kort er aktivert på enheten. Ingen andre opplysninger blir delt. Hvis brukerne ikke ønsker å dele denne informasjonen, kan de deaktivere Apple Pay-spørringer i personverninnstillingene i Safari på iPhone-, iPad- og Mac-enheter.

Etter at en forhandlerøkt er validert, er alle sikkerhets- og personverntiltakene de samme som når brukeren betaler i en app.

Hvis brukeren overfører betalingsrelatert informasjon fra en Mac til en iPhone eller Apple Watch, bruker Apple Pay Handoff Apple identity service (IDS)-protokollen med gjennomgående kryptering til å overføre betalingsrelatert informasjon fra brukerens Mac til autoriseringsenheten. IDS bruker brukerens enhetsnøkler til å foreta kryptering, slik at ingen andre enheter kan dekryptere denne informasjonen og Apple ikke har tilgang til nøklene. Enhetsoppdaging for Handoff med Apple Pay inneholder typen og den unike ID-en til brukerens kredittkort sammen med noe metadata. Det enhetsspesifikke kontonummeret på brukerens kort deles ikke og lagres fortsatt trygt på brukerens iPhone eller Apple Watch. Apple overfører også brukerens nylig brukte kontakt-, leverings- og fakturaadresse på en sikker måte via iCloud-nøkkelring.

Etter at brukeren har godkjent betalingen ved hjelp av Touch ID, Face ID, koden eller ved å dobbelttrykke på sideknappen på Apple Watch, blir et betalingskjennetegn som er unikt, først kryptert til de enkelte nettstedenes forhandlersertifikat, deretter sikkert overført fra brukerens iPhone eller Apple Watch til Macen og så levert til forhandlerens nettsted.

Kun enheter som er i nærheten av hverandre, kan etterspørre og fullføre betalinger. Nærhet bestemmes av Bluetooth Low Energy-annonsering (BLE).

Kontaktløse kort i Apple Pay

For å overføre data fra støttede kort til kompatible NFC-terminaler, bruker Apple protokollen Apple Value Added Services (Apple VAS). VAS-protokollen kan implementeres på kontaktløse terminaler og bruker NFC til å kommunisere med støttede Apple-enheter. VAS-protokollen fungerer over korte avstander og kan brukes til å presentere kontaktløse kort uavhengig eller som en del av en Apple Pay-transaksjon.

Når enheten holdes nær NFC-terminalen, starter terminalen mottak av kortinformasjonen ved å sende en forespørsel om et kort. Hvis brukeren har et kort med utstederens ID, blir brukeren bedt om å godkjenne bruken ved hjelp av Touch ID, Face ID eller kode. Kortinformasjonen, et tidsmerke og en tilfeldig ECDH P-256-engangsnøkkel brukes sammen med utstederens offentlige nøkkel til å avlede en krypteringsnøkkel for kortdataene, som sendes til terminalen.

Fra iOS 12 til og med iOS 13 kan brukere manuelt velge et kort før det vises for forhandlerens NFC-terminal. I iOS 13.1 eller nyere kan utstedere konfigurere manuelt valgte kort til enten å kreve brukergodkjenning eller til å brukes uten godkjenning.

Gjøre kort ubrukelige i Apple Pay

Kredittkort, debetkort og forhåndsbetalte kort som legges til Secure Element, kan kun brukes hvis Secure Element forelegges godkjenning ved hjelp av den samme sammenkoblingsnøkkelen og den tilfeldige godkjenningsverdien (AR-verdien) som ble brukt da kortet ble lagt til. Når Secure Element mottar en ny AR-verdi, merker det eventuelle kort som er lagt til tidligere, som slettet. Dette gjør det mulig for operativsystemet å gi beskjed til Secure Enclave om at kort skal settes som ubrukelige ved at kortets eksemplar av AR-koden merkes som ugyldig hvis følgende situasjoner oppstår:

Metode	Enhet
Når koden deaktiveres	iPhone, iPad, Apple Watch
Når passordet deaktiveres	Mac
Brukeren logger ut fra iCloud	iPhone, iPad, Mac, Apple Watch
Brukeren velger Slett alt innhold og alle innstillinger	iPhone, iPad, Apple Watch
Enheten gjenopprettes fra gjenopprettingsmodus	iPhone, iPad, Mac, Apple Watch
Frakobling	Apple Watch

Midlertidig sperring, fjerning og sletting av kort

Brukere kan sperre Apple Pay midlertidig på iPhone, iPad og Apple Watch, ved å sette enheten i Mistet-modus ved hjelp av «Hvor er?». Brukerne har også muligheten til å fjerne og slette kortene fra Apple Pay ved hjelp av «Hvor er?», iCloud.com eller direkte på enheten ved hjelp av Wallet-appen. På Apple Watch kan kort fjernes ved hjelp av iCloud-innstillinger, Apple Watch-appen på iPhone eller direkte på klokken. Muligheten til å foreta betalinger på enheten ved hjelp av kort blir sperret midlertidig eller fjernes fra Apple Pay av kortutstederen eller det aktuelle betalingsnettverket selv om enheten er frakoblet og ikke koblet til et mobil- eller Wi-Fi-nettverk. Brukerne kan også ringe til kortutstederen for å sperre kortet midlertidig eller fjerne det fra Apple Pay.

Når en bruker sletter hele enheten ved hjelp av «Slett alt innhold og alle innstillinger» ved hjelp av «Hvor er?» eller gjenoppretter enheten, får dessuten Secure Element beskjed fra iPhone, iPad, iPod touch, Mac og Apple Watch om å merke alle kort som slettet. Da gis kortene umiddelbart statusen ubrukelig inntil Apple Pay-tjenerne kan kontaktes for å slette kortene fullstendig fra Secure Element. Uavhengig av dette merker Secure Enclave AR-en som ugyldig, slik at det ikke er mulig å gi flere betalingsgodkjenninger for kort som er registret tidligere. Når enheten er tilkoblet, forsøker den å kontakte Apple Pay-tjenerne for å bidra til å sikre at alle kortene i Secure Element er slettet.

Apple Cash-sikkerhet i iOS, iPadOS og watchOS

Oversikt

I iOS 11.2 eller nyere, iPadOS 13.1 eller nyere og watchOS 4.2 eller nyere kan Apple Pay brukes på iPhone, iPad eller Apple Watch til å sende, motta og be om penger fra andre brukere. Når en bruker mottar penger, legges de til på en Apple Cash-konto som er tilgjengelig i Wallet-appen eller i Innstillinger > Wallet og Apple Pay på alle kvalifiserte enheter der brukeren har logget på med Apple-ID-en.

I iOS 14, iPadOS 14 og watchOS 7 kan organisatoren av en iCloud-familie som har verifisert identiteten sin, aktivere Apple Cash for familiemedlemmer som er under 18 år. Organisatoren kan også begrense disse brukernes mulighet for å sende penger til kun familiemedlemmer eller kun kontakter. Hvis et familiemedlem under 18 år gjennomfører en gjenoppretting av Apple-ID-konto, må organisatoren i familien manuelt reaktivere Apple Cash-kortet for brukeren. Hvis et familiemedlem under 18 år ikke lenger er en del av iCloud-familien, overføres Apple Cash-saldoen automatisk til organisatorens konto.

Når brukeren konfigurerer Apple Cash, kan den samme informasjonen som når brukeren legger til et kreditt- eller debetkort, deles med vår partner Green Dot Bank og med Apple Payments Inc., et heleid datterselskap som er opprettet for å beskytte brukerens personvern ved å lagre og behandle informasjon separat fra resten av Apple og på en måte som resten av Apple ikke kjenner til. Denne informasjonen brukes kun til problemløsing, svindelforebygging og for lovregulerte formål.

Bruke Apple Cash i iMessage

For å bruke person-til-person-betalinger og Apple Cash må en bruker være logget på iCloud-kontoen på en enhet som er kompatibel med Apple Cash og der tofaktorautentisering er konfigurert på iCloud-kontoen. Pengeforespørsler og overføringer mellom brukere startes fra Meldinger-appen eller ved å spørre Siri. Når en bruker forsøker å sende penger, viser iMessage Apple Pay-arket. Apple Cash-saldoen brukes alltid først. Hvis det er nødvendig, vil ytterligere midler trekkes fra et annet kreditt- eller debetkort brukeren har lagt til i Wallet-appen.

Bruke Apple Cash i butikker, i apper og på internett

Apple Cash-kortet i Wallet-appen kan brukes med Apple Pay til å gjennomføre betalinger i butikker, i apper og på nettet. Penger på Apple Cash-kontoen kan også overføres til en bankkonto. I tillegg til å motta penger fra en annen bruker, kan penger legges til på Apple Cash-kontoen fra et debetkort eller forhåndsbetalt kort i Wallet-appen.

Apple Payments Inc. lagrer og kan bruke brukerens transaksjonsdata til problemløsing, svindelforebygging og lovregulerte formål når en transaksjon er fullført. Resten av Apple vet ikke hvem brukeren sendte penger til, mottok penger fra, eller hvor brukeren foretok et kjøp med Apple Cash-kortet.

Når brukeren sender penger med Apple Pay, legger til penger på en Apple Cash-konto eller overfører penger til en bankkonto, sendes en forespørsel til Apple Pay-tjenerne for å anskaffe en kryptografisk «nonce»-verdi som ligner verdien som returneres for Apple Pay i apper. «Nonce»-verdien sendes videre sammen med andre transaksjonsdata til Secure Element for å generere en betalingssignatur. Når betalingssignaturen kommer ut av Secure Element, sendes den til Apple Pay-tjenerne. Autentiseringen, integriteten og riktigheten av transaksjonen verifiseres via betalingssignaturen og «nonce»-verdien av Apple Pay-tjenere. Deretter starter pengeoverføringen, og brukeren varsles ved fullført transaksjon.

Hvis transaksjonen involverer:

- et debetkort for å overføre penger til Apple Cash
- overføring av penger til Apple Cash ved negativ saldo

Det produseres også et kryptert betalingsakkreditiv som sendes til Apple Pay-tjenere, på samme måte som Apple Pay fungerer i apper og på nettsteder.

Hvis saldoen på Apple Cash-kontoen overstiger et visst beløp eller hvis uvanlig aktivitet oppdages, blir brukeren bedt om å bekrefte identiteten. Informasjon som oppgis for å bekrefte brukerens identitet, for eksempel fødselsnummer eller svar på spørsmål (for eksempel bekrefte gateadressen for et tidligere bosted), overføres sikkert til Apples partner og krypteres med deres nøkkel. Apple kan ikke dekryptere disse dataene. Brukerne blir bedt om å verifisere identiteten igjen hvis de utfører en gjenoppretting av Apple-ID-konto, før de igjen får tilgang til Apple Cash-saldoen.

Apple Card-sikkerhet

Søk om Apple Card i Wallet-appen

I iOS 12.4 og nyere, macOS 10.14.6 og nyere og watchOS 5.3 og nyere kan Apple Card brukes i Apple Pay for å betale i butikker, apper og på nett.

For å søke om Apple Card må brukeren være logget på iCloud-kontoen sin på en iOS- eller iPadOS-enhet som er kompatibel med Apple Pay og der tofaktorautentisering er konfigurert på iCloud-kontoen. Når søknaden er godkjent, blir Apple Card tilgjengelig i Wallet-appen eller i Innstillinger > Wallet og Apple Pay på alle godkjente enheter som brukeren har logget på med Apple-ID-en.

Når en bruker søker om Apple Card, kontrolleres identitetsinformasjonen på en sikker måte av Apples partnere, før den deles med Goldman Sachs Bank USA for identifisering og kredittsjekk.

Informasjon som oppgis i forbindelse med søknaden, for eksempel personnummer eller legitimasjon, overføres sikkert til Apples partnere eller Goldman Sachs Bank USA, kryptert med deres respektive nøkler. Apple kan ikke dekryptere disse dataene.

Informasjonen om inntekt som blir oppgitt i forbindelse med søknaden, og informasjon om bankkontoen som brukes til å betale regninger, overføres til Goldman Sachs Bank USA på en sikker måte, kryptert med deres nøkkel. Informasjonen om bankkontoen lagres i nøkkelringen. Apple kan ikke dekryptere disse dataene.

Når Apple Card legges til i Wallet-appen, kan den samme informasjonen som når en bruker legger til et kreditt- eller debetkort, deles med Apples partnerbank Goldman Sachs Bank USA og med Apple Payments Inc. Denne informasjonen brukes kun til problemløsing, svindelforebygging og for lovregulerte formål.

Fysisk kort fra Apple Card kan bestilles i Wallet-appen. Når brukeren mottar det fysiske kortet, aktiveres det med NFC-brikken som er plassert i konvolutten som kortet leveres i. Brikken er knyttet til kortet, og den kan ikke brukes til å aktivere kortet til en annen bruker. Kortet kan også aktiveres manuelt i Wallet-innstillingene. Brukeren kan også når som helst låse eller låse opp det fysiske kortet med Wallet-appen.

Apple Card-betalinger og kortinformasjon i Apple Wallet

Betalinger til Apple Card-kontoen kan gjøres med Wallet-appen i iOS med Apple Cash og en bankkonto. Betalinger kan gjøres faste eller som engangsbetalinger på en bestemt dato med Apple Cash og en bankkonto. Når en bruker gjennomfører en betaling, gjøres et kall til Apple Pay-tjenerne for å skaffe en kryptografisk «nonce»-verdi som likner på Apple Cash. «Nonce»-verdien sendes videre sammen med betalingsinformasjonen til Secure Element for å generere en betalingssignatur. Når betalingssignaturen kommer ut av Secure Element, sendes den til Apple Pay-tjenerne. Autentiseringen, integriteten og korrektheten av betalingen verifiseres via signaturen og «nonce»-verdien av Apple Pay-tjenere, og bestillingen overføres til Goldman Sachs Bank USA for behandling.

For å vise Apple Card-kortnummeret i Wallet-appen må brukeren godkjennes med Face ID, Touch ID eller en kode. Det kan byttes ut av brukeren i delen for kortinformasjon, og det forrige blir deaktivert.

Legge til reisebevis og studentbevis i Wallet

Reisekort

I mange land kan brukere legge til støttede reisekort i Wallet-appen på iPhone- og Apple Watch-modeller som støtter det. Avhengig av operatøren kan dette gjøres ved enten å overføre verdien og reisekortet fra et fysisk kort til dets digitale Apple Wallet-fremstilling eller ved å klargjøre et nytt reisekort i Wallet-appen fra Wallet-appen eller reisekortutstederens app. Når reisekort er lagt til i Wallet-appen, kan brukere reise kollektivt ved å holde iPhone eller Apple Watch i nærheten av kortleseren. Enkelte kort kan også brukes til å betale med.

Reisekort som er lagt til, knyttes til brukerens iCloud-konto. Hvis brukeren legger til flere enn ett kort i Wallet-appen, er det mulig at Apple eller reisekortutstederen kan knytte sammen brukerens personlige data og den tilknyttede kontoinformasjonen mellom kort. Reisekort og transaksjoner beskyttes av et sett med hierarkiske kryptografiske nøkler.

Under prosessen med å overføre saldoen fra det fysiske kortet til Wallet-appen, må brukeren angi informasjon om kortet. Det er mulig at brukeren også må oppgi personopplysninger som bevis på korteierskap. Ved overføring av reisekort fra iPhone til Apple Watch må begge enhetene være tilkoblet.

Saldoen kan fylles opp med midler fra kredittkort, debetkort og forhåndsbetalte kort via Wallet eller fra reisekortutstederens app. Du kan finne ut mer om sikkerheten ved å fylle

opp saldoen ved bruk av Apple Pay i Betale med kort i apper. Hvis du vil finne ut hvordan reisekortet klargjøres i reisekortutstederens app, kan du lese Legge til kreditt- eller debetkort fra appen til en kortutsteder.

Hvis klargjøring fra et fysisk kort støttes, har reisekortutstederen de kryptografiske nøklene som er nødvendige for å autentisere det fysiske kortet og verifisere brukerens angitte data. Etter at dataene er verifisert, kan systemet opprette et enhetskontonummer for Secure Element og aktivere det nye reisekortet i Wallet-appen med den overførte saldoen. I enkelte byer, etter at klargjøring fra det fysiske kortet er fullført, deaktiveres det fysiske kortet.

Til slutt ved begge typer klargjøring, hvis reisekortsaldoen lagres på enheten, krypteres den og lagres til en anvist applet i Secure Element. Transportoperatøren har nøklene for å utføre kryptografiske operasjoner på kortdataene for saldotransaksjoner.

Som standard drar brukere fordeler av sømløs Ekspressreise som gjør det mulig å betale og få skyss uten å kreve Touch ID, Face ID eller en kode. Informasjon som nylig besøkte stasjoner, transaksjonshistorikk og ekstra billetter, er tilgjengelig for kontaktløse kortlesere i nærheten når Ekspressmodus er aktivert. Brukere kan aktivere Touch ID-, Face ID- eller kodegodkjenningskravet i Wallet og Apple Pay-innstillingene ved å deaktivere Ekspressreise.

Som med andre Apple Pay-kort kan brukere sperre eller fjerne reisekort ved å:

- fjernslette enheten med «Hvor er?»
- aktivere Mistet-modus med «Hvor er?»
- bruke en MDM-fjernslettingskommando
- fjerne alle kort fra Apple-ID-kontosiden
- fjerne alle kort fra iCloud.com
- fjerne alle kort fra Wallet-appen
- fjerne kortet i utstederens app

Apple Pay-tjenere sender beskjed til transportoperatøren om å sperre eller deaktivere de aktuelle kortene. Hvis en bruker fjerner et reisekort, kan saldoen gjenopprettes ved å legge det tilbake på en enhet som er pålogget med samme Apple-ID. Hvis en enhet er frakoblet, slått av eller ikke kan brukes, er det ikke sikkert at den kan gjenopprettes.

Kreditt- og debetkort

I enkelte byer godtar kortleserne EMV-kort (smartkort) som betaling for transport. Når brukere legger EMV kreditt- eller debetkort på slike lesere, kreves brukergodkjenning på samme måte som ved «Betal med kreditt- og debetkort i butikker.»

I iOS 12.3 og nyere kan enkelte EMV-kreditt-/debetkort i Wallet-appen kunne brukes for Ekspressreise, slik at brukeren kan betale for en reise med en støttet transportoperatør uten krav om Touch ID, Face ID eller en kode. Når en bruker klargjør et EMV-kreditt- eller debetkort, vil det første kortet som er klargjort i Wallet-appen, kunne brukes til Ekspressreise. Brukeren kan trykke på Mer-knappen på kortet i Wallet-appen og deaktivere Ekspressreise ved å sette Innstillinger for ekspressreise til Ingen. Brukeren kan også velge et annet kreditt- eller debetkort som Ekspressreise-kort i Wallet-appen. Touch ID, Face ID eller kode kreves for å velge et annet kort for Ekspressreise.

Apple Card og Apple Cash kan brukes som Ekspressreise.

Studentbevis

I iOS 12 og nyere kan elever, studenter, lærere og ansatte ved deltakende skoler legge til studentbeviset i Wallet-appen på iPhone- og Apple Watch-modeller som støtter dette, slik at de får tilgang til steder og kan betale der kortet kan brukes til det.

Brukeren legger til studentbeviset i Wallet-appen gjennom en app som tilbys av studentbevisutstederen eller en deltakende skole. Den tekniske prosessen som gjør dette mulig, er den samme som beskrives i Legge til kreditt- eller debetkort fra appen til en kortutsteder. I tillegg må utstedende apper støtte tofaktorautentisering på kontoene som beskytter tilgang til deres studentbevis. Et kort kan konfigureres samtidig på opptil to støttede Apple-enheter logget på med samme Apple-ID.

Når et studentbevis legges til i Wallet-appen, slås Ekspressmodus på som standard. Studentbevis med Ekspressmodus kan brukes uten Touch ID, Face ID og kode eller dobbelttrykking på sideknappen på Apple Watch, på terminaler som støtter dette. Brukeren kan trykke på Mer-knappen på forsiden av kortet i Wallet-appen og slå av Ekspressmodus for å deaktivere denne funksjonen. Touch ID, Face ID eller kode kreves for å aktivere Ekspressmodus på nytt.

Studentbevis kan deaktiveres eller fjernes ved å:

- fjernslette enheten med «Hvor er?»
- aktivere Mistet-modus med «Hvor er?»
- motta en MDM-fjernslettingskommando
- fjerne alle kort fra Apple-ID-kontosiden
- fjerne alle kort fra iCloud.com
- fjerne alle kort fra Wallet-appen
- fjerne kortet i utstederens app

iMessage

Oversikt over iMessage-sikkerhet

Apple iMessage er en meldingstjeneste for iOS- og iPadOS-enheter, Apple Watch og Macer. iMessage støtter tekst og vedlegg som bilder, kontakter, steder, lenker og vedlegg direkte i en melding, for eksempel et tommel opp-symbol. Meldinger vises på alle de registrerte enhetene til brukeren, slik at brukeren kan fortsette samtalen fra hvilken som helst av enhetene. iMessage bruker Apples pushvarslingstjeneste (APNs) mye. Apple loggfører ikke innholdet i meldinger eller vedlegg. De beskyttes av gjennomgående kryptering, slik at ingen andre enn sender og mottaker får tilgang til dem. Apple kan ikke dekryptere dataene.

Når en bruker aktiverer iMessage på en enhet, genererer enheten nøkkelpar for kryptering og signering for bruk med tjenesten. For kryptering finnes det en RSA 1280-bit- og en EC 256-bit-krypteringsnøkkel på NIST P-256-kurven. For signaturer brukes Elliptic Curve Digital Signature Algorithm (ECDSA) 256-bit-signeringsnøkler. De private nøklene lagres i enhetens nøkkelring, og de blir kun tilgjengelige etter første opplåsing. De offentlige nøklene sendes til Apples identitetstjeneste (IDS) hvor de deretter blir knyttet til brukerens telefonnummer og e-postadresse, samt enhetens APNs-adresse.

Etter hvert som brukerne aktiverer flere enheter for bruk med iMessage, legges de offentlige nøklene for kryptering og signering, APNs-adressen og tilknyttede telefonnumre til i katalogtjenesten. Brukerne kan også legge til flere e-postadresser. De verifiseres ved at det sendes en bekreftelseskobling. Telefonnumre bekreftes av telefonoperatøren og SIM-kortet. Med enkelte nettverk krever dette at tekstmeldinger brukes (brukeren ser en bekreftelsesdialogrute hvis tekstmeldingen ikke er gratis). Telefonnummerverifisering kan være nødvendig for flere systemtjenester i tillegg til iMessage, for eksempel FaceTime og iCloud. Når det blir lagt til en ny enhet, et nytt telefonnummer eller en ny e-postadresse, vises en melding på alle enhetene brukeren har registrert.

Hvordan iMessage sender og mottar meldinger på en sikker måte

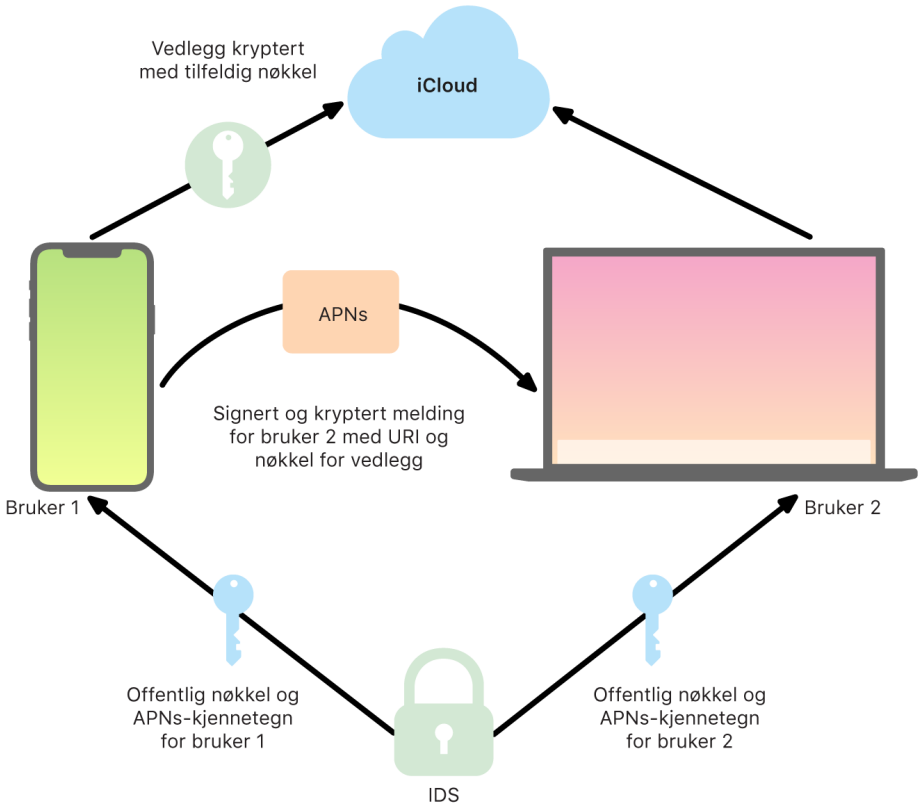
Brukerne starter en ny iMessage-samtale ved å skrive inn en adresse eller et navn. Hvis de oppgir et telefonnummer eller en e-postadresse, kontakter enheten Apple Identity Service (IDS) for å hente de offentlige nøklene og APNs-adressene til alle enhetene som er tilknyttet adressaten. Hvis brukeren oppgir et navn, benytter enheten først brukerens Kontakter-app til å samle inn telefonnumre og e-postadresser som er tilknyttet dette navnet, og deretter hentes de offentlige nøklene og APNs-adressene fra IDS.

Brukerens utgående melding krypteres individuelt for hver av mottakerens enheter. De offentlige krypterings- og signeringsnøklene til mottakerenhetene hentes fra IDS. Senderenheten genererer en tilfeldig 88-bit-verdi for hver mottakerenhet og bruker den som en HMAC-SHA256-nøkkel for å konstruere en 40-bit-verdi som er avledet fra senderens og mottakerens offentlige nøkkel og klarteksten. Sammenkjedingen av 88-bit- og 40-bit-verdiene danner en 128-bit-nøkkel, som krypterer meldingen med den ved hjelp av AES i CTR-modus. 40-bit-verdien brukes av mottakersiden til å bekrefte integriteten til den dekrypterte klarteksten. Denne meldingsspesifikke AES-nøkkelen krypteres ved hjelp av RSA-OAEP til den offentlige nøkkelen til mottakerenheten. Kombinasjonen av den krypterte meldingsteksten og den krypterte meldingsnøkkelen får så en SHA-1-hash, og hashen signeres med Elliptic Curve Digital Signature Algorithm (ECDSA) ved hjelp av senderenhetens private signeringsnøkkel. I iOS 13 og nyere og iPadOS 13.1 og nyere kan enheter bruke en Elliptic Curve Integrated Encryption Scheme-kryptering (ECIES-kryptering) i stedet for RSA-kryptering.

Meldingene som kommer ut av dette, én for hver mottakerenhet, består av den krypterte meldingsteksten, den krypterte meldingsnøkkelen og senderens digitale signatur. Deretter sendes de av gårde til APNs for levering. Metadata, for eksempel tidsmerker og ruteinformasjon for APNs, krypteres ikke. Kommunikasjon med APNs krypteres ved hjelp en såkalt «forward secret» TLS-kanal.

Avhengig av hvilken iOS- eller iPadOS-versjon som brukes, kan APNs kun videreformidle meldinger som har en størrelse på inntil 4 eller 16 kB. Hvis meldingsteksten er for lang, eller hvis et vedlegg som for eksempel et bilde er lagt ved, blir vedlegget kryptert ved hjelp av AES i CTR-modus med en 256-bit-nøkkel som genereres tilfeldig, og lastet opp til iCloud. AES-nøkkelen for vedlegget, Uniform Resource Identifier (URI) og en SHA-1-hash

for nøkkelens krypterte form blir så sendt til mottakeren som innhold i en iMessage, der konfidensialitet og integritet er beskyttet gjennom vanlig iMessage-kryptering, som vist i diagrammet nedenfor.



Hvordan iMessage sender og mottar meldinger.

Hvis det gjelder gruppesamtaler, gjentas denne prosessen for alle mottakerne og enhetene deres.

Alle mottakerenhetene mottar en kopi av meldingen fra APNs og mottar om nødvendig vedlegget fra iCloud. Senderens innkommende telefonnummer eller e-postadresse sammenlignes med mottakerens kontakter slik at navnet vises når det er mulig.

I likhet med alle pushvarslinger slettes meldingen fra APNs når den har blitt levert. Til forskjell fra andre APNs-varslinger legges imidlertid iMessage-meldinger i kø for å bli levert til frakoblede enheter. Meldinger lagres i inntil 30 dager.

Sikker deling av navn og bilde i iMessage

En bruker kan dele et navn og et bilde ved hjelp av iMessage. Brukeren kan velge informasjonen fra Mitt kort, eller tilpasse navnet og legge ved et valgfritt bilde. Deling av navn og bilde i iMessage bruker et totrinnssystem til å sende navnet og bildet.

Dataene deles inn i felt. Hvert felt krypteres og autentiseres uavhengig og sammen ved hjelp av prosessen nedenfor. Det finnes tre felt:

- Navn
- Bilde
- Bildefilnavn

Først genereres en vilkårlig 128-bit-nøkkel på enheten. Deretter deriveres nøkkelen med HKDF-HMAC-SHA256 for å opprette tre undernøkler: Nøkkel 1:Nøkkel 2:Nøkkel 3 = HKDF(registernøkkel, «kallenavn»). For hvert felt genereres en vilkårlig 96-bit-initialiseringsvektor (IV), og dataene krypteres med AES-CTR og Nøkkel 1. Deretter beregnes en meldingsautentiseringskode (MAC) med HMAC-SHA256 ved hjelp av Nøkkel 2, som dekker feltnavnet, felt IV og kryptert tekst fra feltet. Til slutt blir settet av MAC-verdier for de individuelle feltene knyttet sammen, og MAC blir beregnet med HMAC-SHA256 ved hjelp av Nøkkel 3. MAC (256-bit) lagres sammen med de krypterte dataene. De første 128 bitene av denne MAC brukes som RecordID.

Denne krypterte oppføringen lagres i den offentlige CloudKit-databasen under RecordID. Denne oppføringen muteres aldri, og når brukeren velger å bytte navn og bilde, genereres en ny kryptert oppføring hver gang. Når bruker 1 velger å dele navn og bilde med bruker 2, sendes oppføringsnøkkelen sammen med RecordID i iMessage-nyttelasten, som er kryptert.

Når enheten til bruker 2 mottar denne iMessage-nyttelasten, registrerer den at nyttelasten inneholder RecordID og nøkkel for kallenavn og bilde. Enheten til bruker 2 går til den offentlige CloudKit-databasen for å hente kryptert navn og bilde og RecordID, og sender informasjonen ved hjelp av iMessage.

Når meldingen er mottatt, dekrypterer enheten til bruker 2 nyttelasten og kontrollerer signaturen ved hjelp av RecordID. Hvis informasjonen blir godkjent, vises navnet og bildet for bruker 2, og brukeren kan velge å legge informasjonen til i kontaktene eller bruke den i Meldinger.

Sikker bruk av Spør bedriften med Meldinger-appen

Spør bedriften er en meldingstjeneste som gjør det mulig for brukere å kommunisere med bedrifter i Meldinger-appen. Med Spør bedriften har brukeren alltid kontroll over samtalen. De kan også slette samtalen og blokkere bedriften fra å sende dem meldinger i fremtiden. Av personvernhensyn mottar ikke bedriften brukerens telefonnummer, e-postadresse eller iCloud-kontoinformasjon. I stedet blir en tilpasset, unik identifikator som kalles *Opaque ID*, generert av Apple Identity Service (IDS) og delt med bedriften. Opaque ID er unik for relasjonen mellom brukerens Apple-ID og bedriftens bedrifts-ID. En bruker har en annen Opaque ID for hver bedrift de kontakter med Spør bedriften. Brukeren bestemmer hvis og når de vil dele personlig identifiserende informasjon med bedriften.

Spør bedriften støtter administrerte Apple-ID-er fra Apple Business Manager og finner ut om de er aktivert for iMessage og FaceTime i Apple School Manager.

Meldinger som sendes til bedriften, krypteres mellom brukerens enhet og Apples meldingstjenere og bruker samme sikkerhet og Apple-meldingstjenere som iMessage-meldinger. Apples meldingstjenere dekrypterer disse meldingene i RAM og videresender dem til bedriften i en kryptert lenke ved hjelp av TLS 1.2. Meldinger lagres aldri i ukryptert form på vei gjennom Apples Spør bedriften-tjeneste. Bedriftenes svar sendes også ved hjelp av TLS 1.2 til Apples meldingstjenere, der de krypteres ved å bruke de unike offentlige nøklene til hver mottakerenhet.

Hvis brukerenhetene er tilkoblet, leveres meldingen umiddelbart og bufres ikke på Apples meldingstjenere. Hvis en brukerenhet ikke er tilkoblet, bufres den krypterte meldingen i opptil 30 dager, slik at brukeren kan motta den når enheten er tilkoblet igjen. Så snart enheten er tilkoblet igjen, blir meldingen levert og slettet fra bufferen. Etter 30 dager utløper uleverte, bufrede meldinger og slettes permanent.

Spør bedriften-tjenesten lagrer aldri samtalehistorikk.

FaceTime-sikkerhet

FaceTime er Apples tjeneste for video- og lydsamtaler. I likhet med iMessage bruker FaceTime-anrop Apples pushvarslingstjeneste (APNs) til å etablere den første forbindelsen med brukerens registrerte enheter. Lyd- og bildeinnholdet i FaceTime-anrop beskyttes av gjennomgående kryptering, slik at ingen unntatt sender og mottaker får tilgang til det. Apple kan ikke dekryptere dataene.

Den første FaceTime-forbindelsen opprettes gjennom Apples tjenerinfrastruktur, som videresender datapakker mellom brukernes registrerte enheter. Ved hjelp av APNs-varslinger og STUN-meldinger over den viderekoblede forbindelsen bekrefter enhetene identitetssertifikatene sine og danner en delt hemmelighet for hver økt. Den delte hemmeligheten brukes til å utlede øktnøkler for mediekanaler som strømmes ved hjelp av Secure Real-time Transport Protocol (SRTP). SRTP-pakker krypteres ved hjelp av AES256 i Counter Mode og HMAC-SHA1. Etter den opprinnelige forbindelsen og sikkerhetsoppsettet, bruker FaceTime STUN og Internet Connectivity Establishment (ICE) til å etablere en peer-to-peer-forbindelse mellom enheter, hvis det er mulig.

Gruppesamtaler i FaceTime utvider FaceTime med støtte for opptil 33 samtidige deltakere. I likhet med vanlige FaceTime-samtaler med to deltakere, krypteres samtalene gjennomgående mellom de inviterte deltakernes enheter. Selv om gruppesamtaler

i FaceTime gjenbruker store deler av arkitekturen for vanlige FaceTime-samtaler, bruker gruppesamtaler i FaceTime også en ny nøkkeletableringsmekanisme i tillegg til autentiseringen som utføres med Apple Identity Service (IDS). Denne protokollen sørger for «forward secrecy», noe som innebærer at innholdet i tidligere samtaler ikke er tilgjengelig for noen hvis enheten blir utsatt for et sikkerhetsbrudd. Øktnøkler pakkes ved hjelp av AES-SIV og distribueres mellom deltakerne ved hjelp av en ECIES-konstruksjon med kortvarige P-256 ECDH-nøkler.

Når nye telefonnumre eller e-postadresser legges til i en aktiv gruppesamtale i FaceTime, oppretter aktive enheter nye medienøkler. Tidligere brukte nøkler deles aldri med enheter som er nye i samtalen.

Hvor er?

Hvor er?-sikkerhet

Oversikt

«Hvor er?»-appen kombinerer Finn iPhone og Finn vennene mine i én app i iOS, iPadOS og macOS. «Hvor er?» kan hjelpe brukere med å finne en enhet som har blitt borte, selv en Mac som ikke er tilkoblet internett. En tilkoblet enhet kan enkelt rapportere plasseringen sin til brukeren via iCloud. «Hvor er?[2] fungerer uten tilkobling til internett ved å sende ut Bluetooth-signaler med kort rekkevidde fra den forsvunne enheten som kan fanges opp av andre Apple-enheter som er i bruk i nærheten. Enhetene som er i nærheten, videresender så plasseringen til den forsvunne enheten til iCloud, slik at brukerne kan finne den i «Hvor er?»-appen samtidig som de beskytter alle de involverte brukernes personvern og sikkerhet. «Hvor er?» fungerer også med en Mac som ikke er tilkoblet internett, og er i dvale.

Ved hjelp av Bluetooth og de mange hundre millioner iOS-, iPadOS- og macOS-enhetene som brukes over hele verden, kan brukeren finne en enhet som har blitt borte, selv om den ikke er koblet til Wi-Fi eller et mobilnett. Alle iOS-, iPadOS- og macOS-enheter med «Finn frakoblede enheter» aktivert i «Hvor er?»-innstillingene, kan brukes til å finne andre enheter. Det betyr at enheten kan bruke Bluetooth til å registrere andre, frakoblede enheter som har blitt borte, og deretter bruke nettverkstilkoblingen til å varsle eieren om en omtrentlig plassering. Når «Finn frakoblede enheter» er aktivert på en enhet, betyr det også at den kan finnes av andre på samme måte. Hele interaksjonen er gjennomgående kryptert, helt anonym og utviklet for å bruke lite batteri og data, slik at batteritiden og mobilabonnementet påvirkes minimalt, og brukerens personvern ivaretas.

Merk: «Hvor er?» er kanskje ikke tilgjengelig i alle land eller områder.

Gjennomgående kryptering

«Hvor er?» er basert på avansert kryptering med offentlig nøkkel. Når «Finn frakoblede enheter» er aktivert i «Hvor er?»-innstillingene, genereres et Elliptic Curve (EC) privat nøkkelpar for P-224-kryptering, kalt {d,P}, direkte på enheten. *d* er den private nøkkelen, og *P* er den offentlige nøkkelen. I tillegg er en 256-bit hemmelig SK_0 og en teller $_i$ initialisert med null. Det private nøkkelparet og hemmeligheten sendes aldri til Apple, og de

synkroniseres bare mellom brukerens andre enheter på en gjennomgående kryptert måte ved hjelp av iCloud-nøkkelring. Hemmeligheten og telleren brukes til å utlede gjeldende symmetrisk nøkkel-SK_i med følgende rekursive oppbygging: $SK_i = KDF(SK_{i-1}, «oppdater»)$

Basert på nøkkel-SK_i-en beregnes to store heltall, u_i og v_i, med $(u_i,v_i) = KDF(SK_i, «diversifisere»)$. Deretter utledes både den private P-224-nøkkelen, kalt *d*, og den tilsvarende offentlige nøkkelen, kalt *P*, ved hjelp av en klar relasjon som bruker de to heltallene til å beregne et nøkkelpar med kort levetid: Den utledede private nøkkelen er d_i, hvor $d_i = u_i * d + v_i$ (modulo for rekkefølgen for P-224-kurven), og tilsvarende offentlig del er P_i, som kontrollerer at $P_i = u_i*P + v_i*G$.

Når en enhet blir borte og ikke kan kobles til Wi-Fi eller mobilnett, for eksempel hvis en MacBook Pro har blitt glemt igjen på en benk i parken, begynner den å periodisk kringkaste den utledede offentlige nøkkelen P_i i en tidsbegrenset periode som Bluetooth-nyttelast. Ved hjelp av P-224 får den offentlige nøkkelen plass i én Bluetooth-nyttelast. Enhetene i nærheten kan bidra til å finne den frakoblede enheten ved å kryptere plasseringen med den offentlige nøkkelen. Omtrent hvert 15. minutt blir den offentlige nøkkelen byttet ut med en ny nøkkel ved hjelp av en inkrementell verdi fra telleren og den nevnte prosessen, slik at brukeren ikke kan spores av en kontinuerlig identifikator. Utledningsmekanismen er utviklet for å sørge for at de forskjellige offentlige nøklene P_i ikke kan knyttes til den samme enheten.

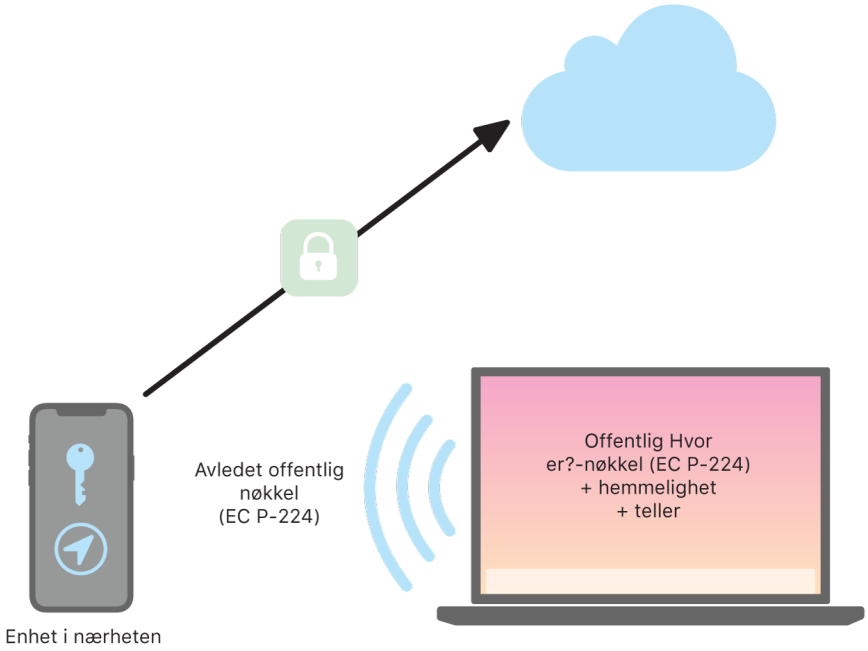
Anonymisering av brukere og enheter

I tillegg til å sørge for at informasjon om plassering og andre data krypteres, beskyttes deltakernes identitet mot andre deltakere og mot Apple. Trafikken som enheter sender til Apple, inneholder ingen informasjon om autentisering verken i innhold eller topptekst. Derfor vet ikke Apple hvem finneren er eller hvem sin enhet som har blitt funnet. Apple registrerer heller ikke informasjon som kan avsløre finnerens identitet, eller informasjon som andre kan bruke til å avdekke finneren og eieren. Den som eier enheten, får bare den krypterte stedsinformasjonen. Den dekrypteres og vises i «Hvor er?»-appen, uten ytterligere informasjon om hvem som fant enheten.

Bruke Hvor er? til å finne enheter som har blitt borte

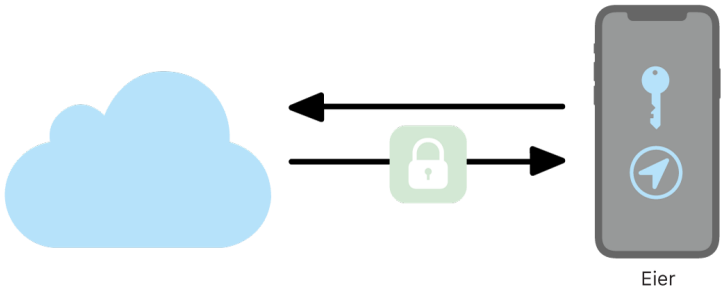
Alle Apple-enheter som er innen rekkevidde for Bluetooth og der «Finn frakoblede enheter» er aktivert, kan registrere et signal fra en annen Apple-enhet der «Hvor er?» er aktivert og lese aktuell sendenøkkel P_i. Ved hjelp av en ECIES-konstruksjon og den offentlige nøkkelen P_i fra enheten som sender, vil enheten som registrerer signalet, kryptere informasjon om plasseringen og overføre den til Apple. Den krypterte plasseringen knyttes til en tjenerindeks som beregnes som SHA256-hash av den offentlige P-224-nøkkelen P_i, som

hentes fra Bluetooth-nyttelasten. Apple har aldri dekrypteringsnøkkelen og kan aldri lese plasseringen som er kryptert av finneren. Den som eier enheten som har blitt borte, kan rekonstruere indeksen og dekryptere den krypterte plasseringen.



Slik finner «Hvor er?» enheter.

Når brukeren forsøker å finne enheten som har blitt borte, anslås det et forventet intervall for perioden det søkes etter plasseringen. Med kunnskap om den opprinnelige private P-224-nøkkelen *d* og de hemmelige verdiene SK_i i intervallet for tellerverdier for søkeperioden, kan eieren rekonstruere verdisettet $\{d_i, SHA256(P_i)\}$ for hele søkeperioden. Deretter kan eierens enhet som ble brukt til å finne enheten som har blitt borte, gjennomføre forespørsler til tjeneren ved hjelp av indeksverdisettet $SHA256(P_i)$ og laste ned de krypterte plasseringene fra tjeneren. «Hvor er?»-appen dekrypterer de krypterte plasseringene ved hjelp av tilsvarende private nøkler d_i, og viser en anslått plassering for enheten som har blitt borte. Stedsrapporter fra flere enheter kombineres i eierens app for å generere en mer nøyaktig plassering.



Slik får eieren enhetens plassering fra «Hvor er?»-appen.

Finne enheter som er frakoblet

Hvis en bruker har aktivert Finn iPhone på enheten sin, er søk etter frakoblede enheter aktivert som standard når enheten oppgraderes til iOS 13 eller nyere, iPadOS 13.1 eller nyere og macOS 10.15 eller nyere. Dette er utviklet for at alle brukere skal få best mulig sjanse til å finne enheten sin hvis den blir borte. Hvis brukeren velger å ikke delta, kan «Finn frakoblede enheter» deaktiveres i «Hvor er?»-innstillingene på enheten. Når «Finn frakoblede enheter» er deaktivert, vil enheten ikke lenger kunne finne andre enheter, og den vil heller ikke kunne finnes av andre enheter. Men brukeren kan fortsatt finne enheten hvis den er koblet til Wi-Fi eller mobilnett.

Når en forsvunnet, frakoblet enhet blir funnet, får brukeren en varsling og en e-postmelding med beskjed om at enheten er funnet. For å vise hvor den forsvunne enheten befinner seg, åpner brukeren «Hvor er?»-appen og velger fanen Enheter. I stedet for å vise enheten på et tomt kart, slik «Hvor er?»-appen ville gjort før enheten ble funnet, vises et kart med omtrentlig adresse og informasjon om hvor lenge det er siden enheten ble registrert. Hvis flere rapporter om plasseringen kommer inn, oppdateres både aktuell plassering og tiden automatisk. Selv om brukerne ikke kan spille av en lyd på en frakoblet enhet eller fjernslette den, kan de bruke informasjonen til å forsøke å finne enheten ved for eksempel å dra til det aktuelle stedet.

Kontinuitet

Oversikt over Kontinuitet-sikkerhet

Kontinuitet drar nytte av teknologier som iCloud, Bluetooth og Wi-Fi for å gjøre det mulig for brukerne å fortsette med en aktivitet fra en enhet til en annen, foreta og motta anrop, sende og motta tekstmeldinger og dele en internettforbindelse via mobilnettverket.

Handoff-sikkerhet

Oversikt

Når en brukers iOS-, iPadOS- og macOS-enheter er i nærheten av hverandre, kan brukeren ved hjelp av Handoff automatisk sende det han eller hun jobber med, fra den ene enheten til den andre. Med Handoff kan brukeren bytte enhet og fortsette å jobbe umiddelbart.

Når en bruker logger på iCloud på en annen enhet som er Handoff-aktivert, danner de to enhetene en Bluetooth Low Energy (BLE) 4.2 out-of-band-sammenkobling ved hjelp av APNs. Enkeltmeldingene krypteres på samme måte som meldinger i iMessage. Etter at enhetene er sammenkoblet, genererer hver av dem en symmetrisk 256-bit AES-nøkkel som lagres i enhetens nøkkelring. Denne nøkkelen kan kryptere og autentisere BLE-annonsering som formidler enhetens nåværende aktivitet til andre enheter som er sammenkoblet med iCloud ved hjelp av AES256 i GCM-modus, med beskyttelsestiltak mot repetisjoner.

Den første gangen en enhet mottar en annonsering fra en ny nøkkel, etablerer den en BLE-forbindelse til enheten der annonseringen kom fra og utfører en utveksling av krypteringsnøkler for annonseringen. Denne forbindelsen sikres ved hjelp av standard BLE 4.2-kryptering samt kryptering av hver enkelt melding. Det er omtrent samme måte som iMessage-krypteres på. I enkelte situasjoner sendes disse meldingene med APNs i stedet for med BLE. Aktivitetens nyttelast beskyttes og overføres på samme måte som en iMessage.

Handoff mellom installerte apper og nettsteder

Handoff gjør det mulig for en app som er installert i iOS, iPadOS eller macOS, å gjenopprette brukeraktivitet på en nettside i domener som rettmessig kontrolleres av apputvikleren. Det gjør det også mulig å gjenoppta brukeraktivitet i en installert app i en nettleser.

For å bidra til å hindre at installerte apper gjør krav på å gjenopprette nettsteder som ikke kontrolleres av utvikleren, må appen kunne bevise at den har rettmessig kontroll over nettdomenene den ønsker å gjenopprette. Kontroll over domenet til et nettsted etableres via mekanismen for delte nettakkreditiver. Se Apptilgang til arkiverte passord for mer informasjon. Systemet må validere domenenavnkontrollen til en app før appen får lov til å godta brukeraktivitet fra Handoff.

Kilden til en nettside fra Handoff kan være alle nettlesere som har tatt i bruk API-ene for Handoff. Når brukeren ser på en nettside, annonserer systemet domenenavnet til nettsiden i de krypterte annonseringsbytene til Handoff. Kun brukerens andre enheter kan dekryptere annonseringsbytene.

På en mottakerenhet oppfatter systemet at en installert app godtar Handoff fra det annonserte domenenavnet og viser symbolet for den installerte appen som Handoff-valget. Når den installerte appen startes, mottar den hele URL-en og tittelen til nettsiden. Ingen annen informasjon sendes videre fra nettleseren til den installerte appen.

I motsatt retning kan den installerte appen angi en reserve-URL når en Handoff-mottakerenhet ikke har den samme appen installert. I så fall viser systemet brukerens standardnettleser som app-alternativet for Handoff (hvis denne nettleseren har tatt i bruk API-er for Handoff). Når Handoff etterspørres, startes nettleseren og gis reserve-URL-en som kildeappen har oppgitt. Det er ingen krav om at reserve-URL-en skal være begrenset til domenenavn som kontrolleres av utvikleren av den installerte appen.

Handoff av større datamengder

I tillegg til å bruke den grunnleggende funksjonen i Handoff, kan det hende at noen apper velger å bruke API-er som støtter sending av større mengder data via peer-to-peer-Wi-Fi-teknologi laget av Apple (omtrent på samme måte som med AirDrop). Mail-appen bruker for eksempel disse API-ene til å støtte Handoff for et utkast til en e-post, som kan inneholde store vedlegg.

Når en app bruker denne funksjonaliteten, starter utvekslingen mellom de to enhetene på samme måte som i Handoff. Mottakerenheten setter imidlertid i gang en ny forbindelse via Wi-Fi etter å ha mottatt den første nyttelasten ved hjelp av Bluetooth Low Energy (BLE). Denne forbindelsen er kryptert (med TLS), som utveksler iCloud-identitetssertifikatene. Identiteten i sertifikatet bekreftes mot brukerens identitet. Ytterligere nyttelaster sendes via denne krypterte forbindelsen til overføringen er fullført.

Universelle utklipp

Universelle utklipp bruker Handoff til å overføre innholdet på utklippstavlen på en sikker måte fra en enhet til en annen slik at det er mulig å kopiere på én enhet og så lime inn på en annen. Innholdet beskyttes på samme måte som andre Handoff-data og deles som standard med Universelle utklipp, med mindre apputvikleren velger at deling ikke er tillatt.

Apper har tilgang til utklippsdata uavhengig av om brukeren har limt utklippstavlen inn i appen. Med Universelle utklipp er denne datatilgangen utvidet til apper på brukerens øvrige enheter (bestemmes av hvordan de er logget på iCloud).

Sikkerhet for sending av iPhone-mobilanrop

Når en brukers Mac eller iPad-, iPod touch- eller HomePod-enhet er koblet til det samme Wi-Fi-nettverket som iPhone-enheten, kan den ringe ut eller motta telefonsamtaler ved hjelp av mobilforbindelsen på iPhonen. Konfigurering krever at enhetene er pålogget både iCloud og FaceTime med den samme Apple-ID-kontoen.

Når det kommer et innkommende anrop, får alle konfigurerte enheter beskjed om dette via Apples pushvarslingstjeneste (APNs), der den enkelte varslingen bruker den samme gjennomgående krypteringen som iMessage. Enheter som er på samme nettverk, viser brukergrensesnittet for varselet om den innkommende samtalen. Når brukeren besvarer en samtale, overføres lyden sømløst fra brukerens iPhone ved hjelp av en sikker peer-to-peer-forbindelse mellom de to enhetene.

Hvis en samtale besvares på én enhet, avsluttes ringingen på iCloud-sammenkoblede enheter i nærheten med en kort beskjed via Bluetooth Low Energy (BLE). Annonseringsbytene krypteres ved hjelp av samme metode som Handoff-annonseringer.

Utgående samtaler sendes også til iPhone via APNs, og lyd overføres på samme måte via den sikre peer-to-peer-koblingen mellom enhetene. Brukerne kan deaktivere sending av telefonsamtaler på en enhet ved å slå av iPhone-mobilanrop i FaceTime-innstillinger.

Sikkerhet for videresending av tekstmelding på iPhone

Videresending av tekstmelding sender automatisk tekstmeldinger som mottas på iPhone, til brukerens registrerte iPad, iPod touch eller Mac. Alle enhetene må være pålogget iMessage-tjenestene med samme Apple-ID-konto. Når Videresending av tekstmeldinger er slått på, er registrering automatisk på enheter innenfor en brukers godkjente sirkel hvis tofaktorautentisering er aktivert. Ellers bekreftes registrering på den enkelte enhet ved å oppgi en tilfeldig sekssifret tallkode som iPhone genererer.

Etter at det er opprettet en kobling mellom enhetene, krypterer og videresender iPhone innkommende tekstmeldinger til enhetene ved å bruke metodene som er beskrevet i Oversikt over iMessage-sikkerhet, eller Svar sendes tilbake til iPhone ved hjelp av den samme metoden, og deretter sender iPhone svaret som en tekstmelding ved hjelp av operatørens mekanisme for sending av tekstmeldinger. Videresending av tekstmelding kan slås av eller på under innstillinger for Meldinger.

Instant Hotspot-sikkerhet

Instant Hotspot kobler andre Apple-enheter til et personlig iOS- og iPadOS-tilgangspunkt. iOS- og iPadOS-enheter som støtter Instant Hotspot, bruker Bluetooth Low Energy (BLE) til å oppdage og kommunisere med alle enheter som er logget på den samme individuelle iCloud-kontoen eller -kontoene som brukes med Familiedeling (i iOS 13 og iPadOS). Kompatible Macer med OS X 10.10 eller nyere bruker den samme teknologien for å oppdage og kommunisere med iOS- og iPadOS-enheter med Instant Hotspot.

Når en bruker første gang oppgir Wi-Fi-innstillinger på en enhet, sender den en BLE-annonsering som inneholder en ID som alle enheter som har logget på den samme

iCloud-kontoen, er enige om. ID-en genereres basert på en DSID (Destination Signaling Identifier) som er knyttet til iCloud-kontoen og byttes med jevne mellomrom. Når andre enheter som har logget på den samme iCloud-kontoen er i nærheten og har støtte for Delt internett, fanger de opp signalet og svarer med å angi at de er tilgjengelige for å bruke Instant Hotspot.

Når en bruker som ikke er en del av Familiedeling, velger en iPhone eller iPad for Delt internett, sendes en forespørsel om å slå på Delt internett til enheten. Forespørselen sendes over en kobling som er kryptert ved hjelp av BLE-kryptering, og forespørselen er kryptert på en måte som ligner iMessage-krypteringen. Enheten svarer deretter over den samme BLE-koblingen ved hjelp av den samme meldingsspesifikke krypteringen med tilkoblingsinformasjon for Delt internett.

For brukere som er en del av Familiedeling, deles tilkoblingsinformasjon for Delt internett på en sikker måte ved hjelp av en mekanisme som tilsvarer den som brukes av HomeKit-enheter til å synkronisere informasjon. Mer spesifikt sikres tilkoblingen som deler informasjon om Delt internett mellom brukere, med en kortvarig ECDH-nøkkel (Curve25519) som autentiseres med brukernes respektive enhetsspesifikke offentlige Ed25519-nøkler. De offentlige nøklene som brukes, er de som tidligere ble synkronisert mellom medlemmene i Familiedeling med IDS da Familiedelingen ble opprettet.

Bilnøkler-sikkerhet i iOS

Oversikt

Bilnøkler-funksjonen støttes som standard på iPhone-enheter og sammenkoblede Apple Watch-enheter som støttes. Bilnøkler vises som kort (opprettet av Apple på vegne av bilprodusenten) i Wallet-appen og støtter alle funksjonene til Apple Pay-kortet (Mistet-modus i iCloud, Fjernsletting, sletting av lokalt kort og Slett alt innhold og alle innstillinger). I tillegg til standard Apple Pay-kortadministrering, kan delte bilnøkler slettes fra eierens iPhone, Apple Watch og i bilens HMI (Human Machine Interface).

Bilnøkler kan brukes til å låse opp og låse bilen og starte motoren eller sette bilen i kjøremodus. «Standardtransaksjonen» har gjensidig autentisering og er obligatorisk for motorstart. Transaksjoner for å låse opp og låse bruker muligens den «raske transaksjonen» når det er nødvendig med tanke på ytelse.

Nøkler opprettes ved å sammenkoble en iPhone med en støttet bil du eier. Alle nøkler opprettes på integrert Secure Element basert på intern nøkkelgenerering med elliptisk kurve (NIST P-256) (ECC-OBKG) og de private nøklene forlater aldri Secure Element. Kommunikasjon mellom enhetene og bilen bruker NFC-standarden, og nøkkeladministrering bruker en tjener-API mellom Apple og bilprodusenten med gjensidig autentisert TLS. Etter at en nøkkel sammenkobles med en iPhone, vil også alle Apple Watch-enheter som er sammenkoblet med iPhonen, også motta en nøkkel. Når en nøkkel slettes i bilen eller på enheten, kan den ikke gjenopprettes. Nøkler på enheter som er mistet eller stjålet, kan sperres midlertidig og åpnes igjen, men hvis de skal klargjøres på nytt på en ny enhet, kreves det ny sammenkobling eller deling.

Sammenkobling med eier

Eieren må bekrefte eierskap av bilen (metoden avhenger av bilprodusenten) og kan starte sammenkoblingsprosessen i bilprodusentens app ved hjelp av en e-postlenke mottatt

fra bilprodusenten eller fra bilmenyen. I alle tilfeller må eieren fremvise et konfidensielt engangspassord for sammenkobling for iPhonen, som brukes til å generere en sikker sammenkoblingskanal ved hjelp av protokollen SPAKE2+ med NIST P-256-kurven. Ved bruk av appen eller e-postlenken overføres passordet automatisk til iPhonen, mens det må oppgis manuelt når sammenkobling startes fra enheten.

Deling av nøkkel

Eierens sammenkoblede iPhone kan dele nøkler med kvalifiserte familiemedlemmers og venners iPhone-enheter (og med sammenkoblede Apple Watch-enheter) ved å sende en enhetsspesifikk invitasjon ved hjelp av iMessage og Apple Identity Service (IDS). Alle delingskommandoer utveksles ved hjelp av IDS-funksjonen med gjennomgående kryptering. Eierens sammenkoblede iPhone sørger for at IDS-kanalen ikke endres under delingsprosessen.

Når invitasjonen godkjennes, oppretter familiemedlemmets eller vennens iPhone en digital nøkkel og sender sertifikatkjeden for nøkkeloppretting tilbake til eierens sammenkoblede iPhone for å bekrefte at nøkkelen ble opprettet på en autentisk Apple-enhet. Eierens sammenkoblede iPhone signerer den offentlig ECC-nøkkelen til det andre familiemedlemmets eller vennens iPhone og sender signaturen tilbake til familiemedlemmets eller vennens iPhone. Signeringsoperasjonen i eierenheten krever brukerautentisering (Touch ID, Face ID eller kode) og en sikker brukerintensjon beskrevet i Bruksområder for Touch ID og Face ID. Autoriseringen kreves når invitasjonen sendes, og den lagres i Secure Element for bruk når venneenheten sender tilbake signeringsforespørselen.

Sletting av nøkkel

Nøkler kan slettes på nøkkelholderenheten fra eierenheten og i bilen. Sletting på nøkkelholder-iPhonen trer i kraft umiddelbart, selv om nøkkelholderen bruker nøkkelen. Det vises derfor en sterk advarsel før sletting.

Sletting av nøkler i kjøretøyet avhenger av om bilprodusenten krever at kjøretøyet er tilkoblet internett for slettingen eller ikke.

I begge tilfeller rapporteres slettingen på nøkkelholderenheten eller i kjøretøyet til en nøkkelbeholdningstjener (KIS) på bilprodusentens side, som registrerer utstedte nøkler for en bil for forsikringsformål.

Eieren kan kreve en sletting fra baksiden av eierkortet. Forespørselen sendes først til bilprodusenten for nøkkelfjerning i bilen. Vilkårene for fjerning av nøkkelen fra kjøretøyet defineres av bilprodusenten. Kun når nøkkelen fjernes i bilen, sender bilprodusentens tjener en forespørsel om ekstern avslutning til nøkkelholderenheten.

Når en nøkkel avsluttes i en enhet, oppretter appleten som administrerer de digitale bilnøklene, en kryptografisk signert avslutningsattestasjon som brukes som bevis på slettingen av bilprodusenten, og brukes til å fjerne nøkkelen fra KIS.

Standardtransaksjoner

Det opprettes en sikker kanal mellom leseren og en iPhone ved å generere kortvarige nøkkelpar på leseren og iPhone-siden. Ved bruk av en nøkkelavtalemetode kan en delt

hemmelighet avledes på begge sider og brukes til generering av en delt symmetrisk nøkkel ved hjelp av Diffie-Hellman, en nøkkelavledingsfunksjon og signaturer fra den langvarige nøkkelen opprettet under sammenkobling.

Den kortvarige offentlige nøkkelen generert på bilsiden signeres med leserens langvarige private nøkkel, som fører til en autentisering av leseren av iPhonen. Fra iPhone-perspektivet er denne protokollen utviklet for å hindre at personvernsensitive data avsløres til uvedkommende som fanger opp kommunikasjonen.

Til slutt bruker iPhonen den opprettede sikre kanalen til å kryptere den offentlige nøkkelidentifikatoren i tillegg til signaturen beregnet på en lesers dataavledede utfordring og noen ekstra appspesifikke data. Denne verifiseringen av iPhone-signaturen av leseren gjør at leseren kan autentisere enheten.

Raske transaksjoner

iPhonen genererer et kryptogram basert på en tidligere delt hemmelighet under en standardtransaksjon. Dette kryptogrammet gjør det mulig for bilen å autentisere enheten raskt i ytelsessensitive scenarioer. Alternativt opprettes det en sikker kanal mellom bilen og enheten ved å avlede øktnøkler fra en hemmelighet tidligere delt under en standardtransaksjon og et nytt kortvarig nøkkelpar. Bilens evne til å opprette den sikre kanalen autentiserer bilen for iPhonen.

Personvern

Bilprodusentens KIS lagrer ikke enhets-ID, SEID eller Apple-ID. Den lagrer kun en uforanderlig identifikator, forekomstens CA-identifikator. Identifikatoren er ikke bundet til noen private data i enheten eller av tjeneren, og den slettes når brukeren sletter enheten fullstendig (med Slett alt innhold og alle innstillinger).

Nettverkssikkerhet

Oversikt over nettverkssikkerhet

I tillegg til de innebygde sikkerhetsmekanismene som Apple bruker for å beskytte data som er lagret på Apple-enheter, kan bedrifter innføre mange tiltak for å ivareta sikkerheten til informasjonen når den overføres til og fra en enhet. Alle disse sikkerhetsmekanismene og tiltakene er innenfor nettverkssikkerhet.

Siden brukere må kunne få tilgang til bedriftsnettverk fra hvor som helst i verden, er det viktig å bidra til å sikre at de autoriseres, og at data beskyttes under overføring. For å oppnå disse sikkerhetsmålene bruker iOS, iPadOS og macOS velprøvde teknologier og de nyeste standardene for tilkobling til både Wi-Fi- og mobildatanettverk. Det er derfor våre operativsystemer bruker, og tilbyr utviklertilgang til, standard nettverksprotokoller for autentisert, autorisert og kryptert kommunikasjon.

TLS-sikkerhet

iOS, iPadOS og macOS støtter Transport Layer Security (TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3) og Datagram Transport Layer Security (DTLS). TLS-protokollen støtter både AES128 og AES256 og foretrekker kodesamlinger med «forward secrecy». Internett-apper som Safari, Kalender og Mail bruker denne protokollen automatisk for å opprette en kryptert kommunikasjonskanal mellom enheten og nettverkstjenestene. Høynivå-API-er (for eksempel CFNetwork) gjør det enkelt for utviklere å ta i bruk TLS i appene, samtidig som lavnivå-API-er (for eksempel Network-rammeverket) sørger for finmasket kontroll. CFNetwork tillater ikke SSL 3, og det er forbudt for apper som bruker WebKit (for eksempel Safari), å opprette SSL 3-forbindelser.

Fra og med iOS 11 og macOS 10.13 tillates ikke lenger SHA-1-sertifikater til bruk med TLS-forbindelser, med mindre de er godkjent av brukeren. Sertifikater med RSA-nøkler som er kortere enn 2048 bit, tillates heller ikke. Kodesamlingen RC4 godtas ikke fra og med iOS 10 og macOS 10.12. TLS-klienter eller -tjenere som er implementert med SecureTransport API-er, har som standard ikke aktivert kodesamlingen RC4, og de kan ikke opprette forbindelse når RC4 er den eneste tilgjengelige kodesamlingen. For å øke sikkerheten bør tjenester eller apper som krever RC4, oppgraderes til å bruke sikre kodesamlinger. I iOS 12.1 kreves det at sertifikater utstedt etter 15. oktober 2018 fra et systemgodkjent rotsertifikat, må være logget i en godkjent Certificate Transparency-logg før de kan brukes med TLS-forbindelser. TLS 1.3 er aktivert som standard for Network-rammeverk og NSURLSession API-er i iOS 12.2. TLS-klienter som bruker SecureTransport-API-er, kan ikke bruke TLS 1.3.

App Transport Security

App Transport Security sørger for standard tilkoblingskrav slik at apper følger mønsterpraksiser for sikre forbindelser ved bruk av API-ene NSURLConnection, CFURL eller NSURLSession. App Transport Security begrenser som standard kodevalgene slik at det kun er mulig å velge samlinger som gir «forward secrecy», nærmere bestemt:

- ECDHE_ECDSA_AES og ECDHE_RSA_AES i Galois/Counter Mode (GCM)
- CBC-modus (Cipher Block Chaining)

Apper kan deaktivere kravet om «forward secrecy» per domene, og i så fall blir RSA_AES lagt til settet med tilgjengelige koder.

Tjenere må ha støtte for TLS 1,2 og «forward secrecy», og sertifikater må være gyldige og signerte ved hjelp av SHA256 eller sterkere med minimum en 2048-bit RSA-nøkkel eller en 256-bit-nøkkel basert på elliptisk kurve.

Nettverkstilkoblinger som ikke oppfyller disse kravene, kan ikke opprettes, med mindre appen overstyrer App Transport Security. Ugyldige sertifikater vil alltid føre til en feil som ikke kan overstyres, og at ingen tilkobling opprettes. App Transport Security tas automatisk i bruk for apper som er kompilert for iOS 9 eller nyere og macOS 10.11 eller nyere.

Kontroll av sertifikatets gyldighet

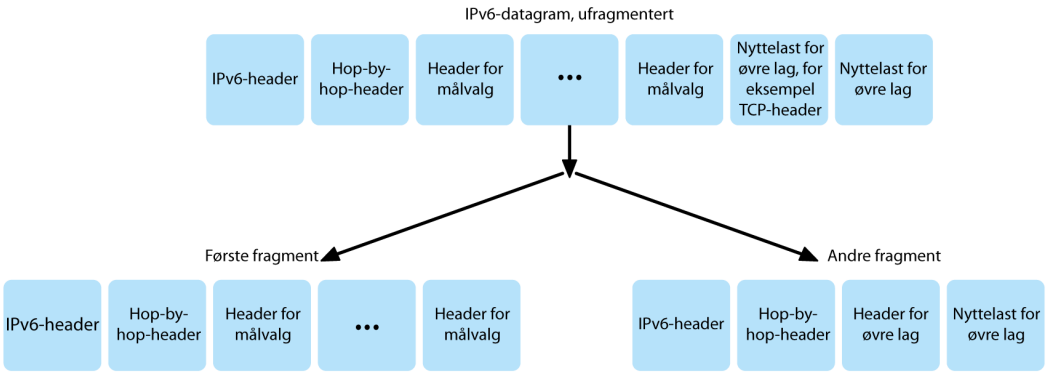
Kontroll av status for et TLS-sertifikat gjennomføres i tråd med etablerte bransjestandarder, som definert i RFC 5280, og ved hjelp av nye standarder, for eksempel RFC 6962 (sertifikat-transparens). I iOS 11 og nyere og macOS 10.13 og nyere oppdateres Apple-enheter regelmessig med en oppdatert liste over tilbakekalte og begrensede sertifikater. Listen lages på grunnlag av lister over tilbakekalte sertifikater (CRL-er) som publiseres av hver av de innebygde rotsertifikatmyndighetene som Apple stoler på, og deres underlagte sertifiseringsmyndigheter. Listen kan også inkludere andre begrensninger etter Apples skjønn. Denne informasjonen brukes når en API-funksjon for nettverk brukes til å opprette en sikker tilkobling. Hvis det finnes for mange tilbakekalte sertifikater fra en sertifiseringsmyndighet til at de kan vises enkeltvis, kan en kontroll i stedet kreve at det finnes Online Certificate Status Response (OCSP), og hvis det ikke finnes, vil ikke godkjenningskontrollen lykkes.

IPv6-sikkerhet

Alle Apple-operativsystemer støtter IPv6, og det er implementert flere mekanismer for å beskytte personvernet til brukerne og stabiliteten til nettverksstabelen. Når SLAAC (Stateless Address Autoconfiguration) brukes, genereres IPv6-adressene i alle grensesnitt på en måte som bidrar til å forhindre sporingsenheter på tvers av nettverk og samtidig gir en god brukeropplevelse ved å sørge for adressestabilitet når det ikke utføres noen nettverksendringer. Adressegenereringsalgoritmen er basert på kryptografisk genererte adresser fra og med RFC 3972, forsterket av en grensesnittspesifikk modifikator for å berettige at selv forskjellige grensesnitt på samme nettverk til slutt har forskjellige adresser. I tillegg opprettes det midlertidige adresser med en foretrukket levetid på 24 timer, og disse brukes som standard for alle nye tilkoblinger. På linje med funksjonen for privat Wi-Fi-adresse som ble introdusert i iOS 14, iPadOS 14 og watchOS 7, genereres det en unik link-local-adresse for hvert Wi-Fi-nettverk som en enhet kobles til. Nettverkets

SSID integreres som et ekstra element for adressegenereringen, i likhet med Network_ID-parameteren fra og med RFC 7217. Denne tilnærmingen brukes i iOS 14, iPadOS 14 og watchOS 7.

For å beskytte mot angrep basert på IPv6-utvidelseshoder og -fragmentering implementerer Apple-enheter beskyttelsestiltak spesifisert i RFC 6980, RFC 7112 og RFC 8021. Disse hindrer blant annet angrep der hodet i det øvre laget kun finnes i det andre fragmentet (som vist under), som igjen kan forårsake tvetydighet for sikkerhetskontroller som tilstandsløse pakkefiltre.



Et IPv6-datagram.

I tillegg, for å bidra til å sikre påliteligheten av IPv6-stabelen i Apple-operativsystemer håndhever Apple-enheter ulike grenser på IPv6-relaterte datastrukturer, for eksempel antall prefikser per grensesnitt.

Sikkerhet for virtuelt privat nettverk (VPN)

Sikre nettverkstjenester som VPN krever vanligvis minimalt med oppsett og konfigurering for å fungere med iOS-, iPadOS- og macOS-enheter.

Protokoller som støttes

Disse enhetene fungerer med VPN-tjenere som støtter følgende protokoller og autentiseringsmetoder:

- IKEv2/IPsec med autentisering med delt hemmelighet, RSA-sertifikater, Elliptic Curve Digital Signature Algorithm-sertifikater (ECDSA-sertifikater), EAP-MSCHAPv2 eller EAP-TLS
- SSL-VPN ved hjelp av egnet klientapp fra App Store
- L2TP/IPsec med brukerautentisering med MS-CHAPV2-passord og maskinautentisering med delt hemmelighet (iOS, iPadOS og macOS) og RSA SecurID eller CRYPTOCard (kun macOS)
- Cisco IPsec med brukerautentisering med passord, RSA SecurID eller CRYPTOCard, og maskinautentisering med delt hemmelighet og sertifikater (kun macOS)

VPN-utrullinger som støttes

iOS, iPadOS og macOS støtter følgende:

- *VPN On Demand:* For nettverk som bruker sertifikatbasert autentisering. IT-regler angir hvilke domener som krever en VPN-tilkobling ved hjelp av en VPN-konfigurasjonsprofil.
- *Per App VPN:* For langt mer spesifikk VPN-tilkobling. Mobile Device Management-løsninger (MDM) kan angi en tilkobling for hver av de administrerte appene og spesifikke domener i Safari. Det bidrar til å sørge for at sikre data alltid overføres til og fra bedriftsnettverket, og at brukerens personopplysninger ikke overføres.

iOS og iPadOS støtter følgende:

- *Always On VPN:* Kan konfigureres for enheter som administreres via en MDM-løsning og er satt under tilsyn med Apple Configurator 2, Apple School Manager eller Apple Business Manager. Med Always On VPN behøver ikke brukerne å slå på VPN for å aktivere beskyttelse ved tilkobling til mobil- eller Wi-Fi-nettverk. Always On VPN gir bedrifter full kontroll over enhetstrafikk ved at all IP-trafikk tunneleres tilbake til bedriften. Standard utveksling av parametere og nøkler for etterfølgende kryptering. IKEv2 sikrer trafikkoverføring med datakryptering. Bedriften kan overvåke og filtrere trafikken til og fra enhetene, sikre data i nettverket og begrense enhetstilgang til internett.

Wi-Fi-sikkerhet

Protokollsikkerhet

Sikker tilgang til trådløse nettverk

Alle Apple-plattformer støtter bransjestandard for protokoller for Wi-Fi-autentisering og -kryptering for å sikre autentisert tilgang og konfidensialitet når man kobler seg til følgende sikre trådløse nettverk:

- WPA2 Personal
- WPA2 Enterprise
- WPA2/WPA3 Transitional
- WPA3 Personal
- WPA3 Enterprise
- WPA3 Enterprise 192-bit-sikkerhet

WPA2 og WPA3 autentiserer hver tilkobling og gir 128-bit-AES-kryptering for å bidra til å sikre konfidensialitet for data som sendes trådløst. Det sørger for at brukernes data er beskyttet med sikkerhet på høyeste nivå når de sender og mottar informasjon via Wi-Fi-nettverkstilkoblinger.

WPA3-støtte

WPA3 støttes på følgende Apple-enheter:

- iPhone 7 eller nyere
- iPad femte generasjon eller nyere
- Apple TV 4K eller nyere
- Apple Watch Series 3 eller nyere
- Macer (sent 2013 og nyere, med 802.11ac eller nyere)

Nyere enheter støtter autentisering med WPA3 Enterprise 192-bit-sikkerhet, inkludert støtte for 256-bit-AES-kryptering ved tilkobling til kompatible trådløse tilgangspunkter (AP-er). Dette gir enda bedre beskyttelse av konfidensialiteten til trafikk som sendes trådløst. WPA3 Enterprise 192-bit-sikkerhet støttes på iPhone 11, iPhone 11 Pro, iPhone 11 Pro Max og nyere iOS- og iPadOS-enheter.

PMF-støtte

I tillegg til å beskytte data som sendes trådløst, utvider Apple-plattformer WPA2- og WPA3-beskyttelse til unicast- og multicast-rammeverk ved hjelp av tjenesten Protected Management Frame (PMF) som er definert i 802.11w. PMF-støtte er tilgjengelig på følgende Apple-enheter:

- iPhone 6 eller nyere
- iPad Air 2 eller nyere
- Apple TV HD eller nyere
- Apple Watch Series 3 eller nyere
- Macer (sent 2013 og nyere, med 802.11ac eller nyere)

Med støtte for 802.1X kan Apple-enheter integreres i en rekke RADIUS-autentiseringsmiljøer. Trådløse 802.1X-autentiseringsmetoder som støttes, inkluderer EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0 og PEAPv1.

Plattformbeskyttelse

Apple-operativsystemer beskytter enheten fra sårbarheter i nettverksprosessorens firmware. Dette betyr at nettverkskontrollere med Wi-Fi har begrenset tilgang til applikasjonsprosessorminne.

- Når USB eller SDIO (Secure Digital Input Output) brukes som grensesnitt mot nettverksprosessoren, kan ikke nettverksprosessoren starte transaksjoner for direkte minnetilgang (DMA) til applikasjonsprosessoren.
- Når PCIe brukes, er hver nettverksprosessor på sin egen, isolerte PCIe-databuss. En Input/Output Memory Management Unit (IOMMU) på hver PCIe-databuss begrenser ytterligere nettverksprosessorens DMA-tilgang til bare minne og ressurser som inneholder dens nettverkspakker og kontrollstrukturer.

Foreldede protokoller

Apple-produkter støtter følgende foreldede protokoller for Wi-Fi-autentisering og -kryptering:

- WEP Open, med både 40-bit- og 104-bit-nøkler
- WEP Shared, med både 40-bit- og 104-bit-nøkler
- Dynamic WEP
- Temporal Key Integrity Protocol (TKIP)
- WPA
- WPA/WPA2 Transitional

Disse protokollene anses ikke lenger som sikre, og det frarådes å bruke dem på grunn av utfordringer med kompatibilitet, pålitelighet, ytelse og sikkerhet. De støttes kun for å sikre bakoverkompatibilitet, og støtten kan fjernes i fremtidige programvareversjoner.

Det anbefales at alle Wi-Fi-implementeringer migreres til WPA3 Personal eller WPA3 Enterprise, slik at man får så robuste, sikre og kompatible Wi-Fi-tilkoblinger som mulig.

Wi-Fi-personvern

Tilfeldig MAC-adresse

Apple-plattformer bruker en tilfeldig MAC-adresse (Media Access Control) når de foretar et Wi-Fi-søk uten å være tilknyttet et Wi-Fi-nettverk. Disse søkene kan utføres for å finne og koble til et kjent Wi-Fi-nettverk eller for å bistå Stedstjenester med apper som bruker geografiske gjerder, for eksempel stedsbaserte påminnelser eller feste et sted i Apple Kart. Wi-Fi-søk som foregår mens det gjøres forsøk på å koble til et foretrukket Wi-Fi-nettverk, er ikke er tilfeldige. Støtte for tilfeldige MAC-adresser for Wi-Fi er tilgjengelig på iPhone 5 og nyere.

Apple-plattformer bruker også tilfeldige MAC-adresser når de foretar ePNO-søk (enhanced Preferred Network Offload) når en enhet ikke er tilknyttet et Wi-Fi-nettverk eller enhetens prosessor er i dvale. Søk med ePNO-søk utføres når en enhet bruker Stedstjenester for apper som bruker geografiske gjerder, for eksempel stedsbaserte påminnelser som bestemmer om enheten er i nærheten av et bestemt sted.

Siden en enhets MAC-adresse endres når den kobles fra et Wi-Fi-nettverk, kan den ikke brukes av passive observatører av Wi-Fi-trafikken til å spore en enhet, selv når enheten er koblet til et mobilnettverk. Apple har informert Wi-Fi-produsenter om at Wi-Fi-søk i iOS og iPadOS bruker en tilfeldig MAC-adresse og at verken Apple eller produsentene kan forutsi disse tilfeldige MAC-adressene.

iOS 14, iPadOS 14 og watchOS 7 har en ny privat Wi-Fi-funksjon: Når en iPhone, iPad, iPod touch eller Apple Watch kobles til et Wi-Fi-nettverk, identifiserer enheten seg med en unik (tilfeldig) MAC-adresse per nettverk. Denne funksjonen kan deaktiveres av brukeren eller med et nytt valg i Wi-Fi-nyttelasten. Under enkelte omstendigheter vil enheten bruke den faktiske MAC-adressen.

Du finner mer informasjon i Apple-kundestøtteartikkelen Bruke private Wi-Fi-adresser i iOS 14, iPadOS 14 og watchOS 7.

Tilfeldig sekvensnummer for Wi-Fi-rammer

Wi-Fi-rammer har et sekvensnummer, som brukes av lavnivå-protokollen 802.11 til å sikre effektiv og pålitelig Wi-Fi-kommunikasjon. Fordi disse sekvensnumrene øker for hver ramme som overføres, kan de brukes til å avdekke informasjon som blir overført under Wi-Fi-søk sammen med andre rammer som overføres av den samme enheten.

For å beskytte mot dette bruker Apple-enheter tilfeldige sekvensnumre når en MAC-adresse endres til en ny tilfeldig adresse. Dette inkluderer å bruke tilfeldige sekvensnummer for hver ny søkeforespørsel som startes mens enheten ikke er koblet til. Funksjonen støttes på følgende enheter:

- iPhone 7 eller nyere
- iPad femte generasjon eller nyere
- Apple TV 4K eller nyere
- Apple Watch Series 3 eller nyere
- iMac Pro (Retina 5K, 27 tommer, 2017) eller nyere
- MacBook Pro (13-tommer, 2018) eller nyere
- MacBook Pro (15-tommer, 2018) eller nyere
- MacBook Air (Retina, 13-tommer, 2018) eller nyere
- Mac mini (2018) eller nyere
- iMac (Retina 4K, 21,5-tommer, 2019) eller nyere
- iMac (Retina 5K, 27-tommer, 2019) eller nyere
- Mac Pro (2019) eller nyere

Wi-Fi-forbindelser

Apple genererer tilfeldige MAC-adresser for peer-to-peer-Wi-Fi-forbindelsene som brukes for AirDrop og AirPlay. Det brukes også tilfeldige adresser til Delt internett i iOS og iPadOS (med SIM-kort) og Internettdeling i macOS.

Det genereres nye, tilfeldige adresser når disse nettverksgrensesnittene startes, og unike adresser genereres uavhengig for hvert grensesnitt ved behov.

Skjulte nettverk

Wi-Fi-nettverk identifiseres av nettverksnavnet, som kalles *Service Set Identifier (SSID)*. Enkelte Wi-Fi-nettverk er satt opp med skjult SSID, noe som fører til at det trådløse tilgangspunktet ikke kringkaster nettverkets navn. Disse er kjent som *skjulte nettverk*. iPhone 6s og nyere enheter gjenkjenner automatisk når et nettverk er skjult. Hvis et nettverk er skjult, sender iOS- eller iPadOS-enheten en kontroll med SSID-en inkludert i forespørselen, men ikke ellers. Dette bidrar til å forhindre at enheten kringkaster navnet til tidligere skjulte nettverk en bruker var koblet til, for derved å sikre personvernet ytterligere.

Bluetooth-sikkerhet

Det finnes to typer Bluetooth i Apple-enheter: Bluetooth Classic og Bluetooth Low Energy (BLE). Modellen for Bluetooth-sikkerhet for begge versjoner har følgende unike sikkerhetsfunksjoner:

- *Sammenkobling:* Prosessen for å opprette en eller flere delte hemmelige nøkler.
- *Binding:* Når nøklene som opprettes under sammenkobling, lagres for bruk ved senere sammenkobling for å skape et godkjent enhetspar.
- *Autentisering:* En kontroll av at begge enhetene har de samme nøklene.
- *Kryptering:* Meldingskonfidensialitet.
- *Meldingsintegritet:* Beskyttelse mot forfalskede meldinger.
- *Secure Simple Pairing:* Beskyttelse mot massiv avlytting og mot man-in-the-middle-angrep.

Bluetooth versjon 4.1 fikk Secure Connections-funksjonen for Bluetooth Classic (BR/EDR) fysisk transport.

Sikkerhetsfunksjonene for hver Bluetooth-type vises i listen under.

Støtter	Bluetooth Classic	Bluetooth Low Energy
Sammenkobling	P-256 elliptisk kurve.	FIPS-godkjente algoritmer (AES-CMAC og P-256 elliptisk kurve).
Binding	Informasjon om sammenkobling lagres på et sikkert sted i iOS-, iPadOS-, macOS-, tvOS- og watchOS-enheter.	Informasjon om sammenkobling lagres på et sikkert sted i iOS-, iPadOS-, macOS-, tvOS- og watchOS-enheter.
Autentisering	FIPS-godkjente algoritmer (HMAC-SHA256 og AES-CTR).	FIPS-godkjente algoritmer.
Kryptering	AES-CCM-kryptografi utført i kontrolleren	AES-CCM-kryptografi utført i kontrolleren
Meldingsintegritet	AES-CCM brukes for meldingsintegritet	AES-CCM brukes for meldingsintegritet
Secure Simple Pairing: Beskyttelse mot passiv avlytting.	Elliptic Curve Diffie-Hellman Exchange (ECDHE)	Elliptic Curve Diffie-Hellman Exchange (ECDHE)
Secure Simple Pairing: Beskyttelse mot man-in-the-middle-angrep (MITM).	To brukerassisterte numeriske metoder: numerisk sammenligning eller passordtilgang.	To brukerassisterte numeriske metoder: numerisk sammenligning eller passordtilgang. Sammenkobling krever brukerrespons, inkludert alle ikke-MITIM sammenkoblingsmoduser.
Bluetooth 4.1 eller nyere	iMac sent 2015 eller nyere MacBook Pro – tidlig 2015 eller nyere	iOS 9 eller nyere iPadOS 13.1 eller nyere macOS 10.12 eller nyere tvOS 9 eller nyere watchOS 2.0 eller nyere

Støtter	Bluetooth Classic	Bluetooth Low Energy
Bluetooth 4.2 eller nyere	iPhone 6 eller nyere	iOS 9 eller nyere iPadOS 13.1 eller nyere macOS 10.12 eller nyere tvOS 9 eller nyere watchOS 2.0 eller nyere

Bluetooth Low Energy og personvern

For å ivareta brukerens personvern har BLE følgende to funksjoner: tilfeldig adresse og nøkkelderivering ved krysstransport.

Tilfeldig adresse er en funksjon som reduserer muligheten til å spore en BLE-enhet i en periode ved at Bluetooth-enhetens adresse endres med jevne mellomrom. For at en enhet som bruker personvernfunksjonen skal kunne koble seg til kjente enheter igjen, må den andre enheten kunne løse enhetsadressen, kalt den *private adressen*. Den private adressen genereres ved å bruke enhetens IRK (Identity Resolving Key), som ble utvekslet under sammenkoblingen.

iOS 13 eller nyere og iPadOS 13.1 eller nyere kan derivere lenkenøkler på tvers av transporter. Denne funksjonen kalles *nøkkelderivering ved krysstransport*. For eksempel kan en lenkenøkkel som ble generert med BLE, brukes til å derivere en lenkenøkkel for Bluetooth Classic. I tillegg inkluderte Apple Bluetooth Classic til BLE-støtte for enheter som støtter funksjonen for sikre tilkoblinger som ble innført i Bluetooth Core Specification 4.1 (se Bluetooth Core Specification 5.1).

Ultra Wideband-sikkerhet i iOS

Den nye, Apple-designede U1-brikken bruker Ultra Wideband-teknologi for gjenkjenning av andre enheters nærvær, slik at iPhone 11, iPhone 11 Pro og iPhone 11 Pro Max eller nyere iPhone-modeller kan registrere nøyaktig hvor andre Apple-enheter med U1-brikke befinner seg. Ultra Wideband-teknologi bruker den samme teknologien til å generere tilfeldige data av informasjon som finnes i andre støttede Apple-enheter:

- Tilfeldig MAC-adresse
- Tilfeldig sekvensnummer for Wi-Fi-ramme.

Single Sign On-sikkerhet

Single Sign On

iOS og iPadOS støtter autentisering for bedriftsnettverk gjennom Single Sign On (SSO). SSO arbeider sammen med Kerberos-baserte nettverk for å autentisere brukere for tjenester som de er autorisert for å få tilgang til. SSO kan brukes for en rekke nettverksaktiviteter, fra sikre Safari-økter til tredjepartsapper. Sertifikatbasert autentisering, som PKINIT, støttes også.

macOS støtter autentisering for bedriftsnettverk ved hjelp av Kerberos. Apper kan bruke Kerberos for å autentisere brukere for tjenester som de er autorisert for å få tilgang til. Kerberos kan brukes for en rekke nettverksaktiviteter, fra sikre Safari-økter og autentisering for filsystemer på nettverket til tredjepartsapper. Sertifikatbasert autentisering støttes, men krever apptilpasning av en utvikler-API.

SSO i iOS, iPadOS og macOS bruker SPNEGO-kjennetegn og HTTP Negotiate-protokollen for å samarbeide med Kerberos-baserte autentiseringsportaler og Windows Integrated Authentication-systemer som støtter Kerberos-billetter. SSO-støtte er basert på åpen kildekode-prosjektet Heimdal.

Følgende krypteringstyper støttes i iOS, iPadOS og macOS:

- AES-128-CTS-HMAC-SHA1-96
- AES-256-CTS-HMAC-SHA1-96
- DES3-CBC-SHA1
- ARCFOUR-HMAC-MD5

Safari har støtte for SSO, og tredjepartsapper som bruker standard nettverks-API-er i iOS og iPadOS, kan også konfigureres til å bruke det. For å konfigurere SSO har iOS og iPadOS støtte for en konfigurasjonsprofilnyttelast der MDM-løsninger kan overføre de nødvendige innstillingene via pushfunksjonalitet. Det inkluderer angivelse av hovednavnet (altså Active Directory-brukerkontoen) og innstillinger på Kerberos-nivå, samt konfigurering av hvilke apper eller Safari-nettadresser som skal kunne bruke SSO.

For å konfigurere Kerberos i macOS, kan du hente billetter med Ticket Viewer, logge på et Windows Active Directory-domene eller bruke `kinit`-kommandolinjeverktøyet.

Extensible Single Sign On

App-utviklere kan tilby sin egen implementering av Single Sign On ved hjelp av SSO-utvidelser. SSO-utvidelser aktiveres når en installert eller nettbasert app skal bruke en ID-leverandør for å godkjenne brukeren. Utviklere kan tilby to typer utvidelser: de som omdirigerer til HTTPS og de som bruker en challenge/response-mekanisme som Kerberos. Det gjør at autentiseringstypene OpenID, OAuth, SAML2 og Kerberos kan støttes av Extensible Single Sign On.

For å bruke en Single Sign On-utvidelse kan en app enten bruke AuthenticationServices-APIen eller stole på operativsystemets mekanisme for å fange opp URL. WebKit og CFNetwork har et innhentingslag som gir sømløs støtte for Single Sign On for alle installerte apper og WebKit-apper. For at en Single Sign On-utvidelse skal aktiveres, må

det installeres en konfigurasjon fra en administrator ved hjelp av en MDM-profil. I tillegg må utvidelser av redirect-typen bruke Associated Domains-nyttelasten til å bevise at identitetstjeneren de støtter, er klar over at de finnes.

Den eneste utvidelsen som leveres med operativsystemet, er Kerberos SSO-utvidelsen.

AirDrop-sikkerhet

Apple-enheter som støtter AirDrop, bruker Bluetooth Low Energy (BLE) og peer-to-peer Wi-Fi-teknologi laget av Apple for å sende filer og informasjon til enheter i nærheten, inkludert AirDrop-kompatible iOS-enheter som kjører iOS 7 eller nyere, og Macer som kjører OS X 10.11 eller nyere. Wi-Fi-radioen brukes til direkte kommunikasjon mellom enheter uten å benytte internettforbindelse eller trådløse tilgangspunkter (AP). I macOS krypteres denne tilkoblingen med TLS.

AirDrop er som standard stilt inn til å dele kun med kontakter. Brukerne kan også velge å bruke AirDrop til å dele med alle eller slå av funksjonen helt. Bedrifter kan angi restriksjoner for bruken av AirDrop for enheter eller apper som administreres ved hjelp av en MDM-løsning.

Bruk av AirDrop

AirDrop bruker iCloud-tjenester til å hjelpe brukere med autentisering. Når en bruker logger seg på iCloud, lagres en 2048-bit RSA-identitet på enheten, og når brukeren aktiverer AirDrop, opprettes en identitetshash for AirDrop basert på e-postadressene og telefonnumrene som er tilknyttet brukerens Apple-ID.

Når en bruker velger AirDrop som metode for å dele et objekt, vil senderenheten sende ut et AirDrop-signal via BLE som inneholder brukerens identitetshash for AirDrop. Andre våkne Apple-enheter som befinner seg i nærheten og har AirDrop slått på, fanger opp signalet via peer-to-peer Wi-Fi, slik at senderenheten kan oppdage identiteten til de enhetene som svarer.

I Kun kontakter-modus blir de mottatte identitetshashene for AirDrop sammenlignet med hashene til personer i mottakerens Kontakter-app. Hvis det blir funnet noen som stemmer overens, svarer mottakerenheten med sin identitetsinformasjon via peer-to-peer Wi-Fi. Hvis det ikke ble funnet noen som stemmer overens, svarer ikke enheten.

I Alle-modus brukes den samme generelle prosessen. Men mottakerenheten svarer selv om det ikke blir funnet noen som stemmer overens i enhetens Kontakter-app.

Senderenheten igangsetter en AirDrop-tilkobling via peer-to-peer Wi-Fi, og bruker denne tilkoblingen til å sende en identitetshash til mottakerenheten. Hvis den lange identitetshashen stemmer overens med hashen til en kjent kontakt i mottakerens Kontakter, vil mottakeren svare med sine lange identitetshasher.

Hvis hashene stemmer, vises mottakerens fornavn og bilde (hvis det finnes i Kontakter) i senderens AirDrop-delingsark. I iOS og iPadOS vises de i delen «Personer» eller «Enheter». Enheter som ikke godkjennes eller autentiseres, vises i avsenderens AirDrop-deleark med et silhuettsymbol og enhetens navn, som definert i Innstillinger > Generelt > Om > Navn. I iOS og iPadOS plasseres de i delen «Andre personer» på AirDrop-delearket.

Brukeren som sender, kan velge hvem de vil dele med. Når bruker er valgt, igangsetter senderenheten en kryptert forbindelse (TLS) med mottakerenheten, og de utveksler iCloud-identitetssertifikater. Identiteten i sertifikatet bekreftes mot Kontakter-appen til hver enkelt bruker.

Hvis sertifikatene bekreftes, blir brukeren som mottar, bedt om å godta den innkommende overføringen fra den identifiserte brukeren eller enheten. Hvis flere mottakere er valgt, gjentas denne prosessen for hver av dem.

Sikkerhet for deling av Wi-Fi-passord på iPhone og iPad

iOS- og iPadOS-enheter som støtter Wi-Fi-passorddeling, bruker en mekanisme som ligner på AirDrop, til å sende et Wi-Fi-passord fra én enhet til en annen.

Når en bruker velger et Wi-Fi-nettverk (anmoderen) og bes om å oppgi Wi-Fi-passordet, starter Apple-enheten en Bluetooth Low Energy-annonsering (BLE) som indikerer at den vil ha Wi-Fi-passordet. Andre Apple-enheter i nærheten som er våkne og som har passordet til det valgte Wi-Fi-nettverket, kobler til enheten som har sendt forespørselen, via BLE.

Enheten som har Wi-Fi-passordet (giveren), ber om kontaktinformasjonen til anmoderen, og anmoderen må bekrefte sin identitet ved hjelp av en mekanisme som tilsvarer den som brukes av AirDrop. Etter at identiteten er bekreftet, sender giveren koden til anmoderen. Denne kan brukes til å koble til nettverket.

Bedrifter kan angi restriksjoner for bruken av Wi-Fi-passorddeling for enheter eller apper som administreres ved hjelp av en MDM-løsning.

Brannmursikkerhet i macOS

macOS har en innebygd brannmur som beskytter Macen mot nettverkstilgang og tjenestenektangrep. Den kan konfigureres i Sikkerhet og personvern-panelet i Systemvalg, og den støtter følgende konfigurasjoner:

- Blokker alle innkommende tilkoblinger, uavhengig av program.
- Tillat automatisk innkommende tilkoblinger til innebygd programvare.
- Tillat automatisk innkommende tilkoblinger til nedlastet signert programvare.
- Tillat eller blokker tilgang basert på brukerspesifiserte programmer.
- Forhindre at Macen svarer testprogrammer som bruker ICMP (Internet Control Message Protocol) og portskanning.

Sikkerhet for utviklerrammeverk

Oversikt over sikkerhet for utviklerrammeverk

Apple tilbyr en rekke rammeverk, slik at tredjepartsutviklere kan utvide Apple-tjenester. Disse rammeverkene er utviklet med fokus på brukerens sikkerhet og personvern:

- HomeKit
- CloudKit
- SiriKit
- DriverKit
- ReplayKit
- ARKit

HomeKit

Sikkerhet for HomeKit-kommunikasjon

Oversikt

Hjemautomatisering med HomeKit har en infrastruktur der sikkerhetsstrukturen i iCloud, iOS, iPadOS og macOS brukes for å beskytte og synkronisere private data uten at de legges åpent for Apple.

HomeKit-identitet og -sikkerhet er basert på offentlig-private Ed25519-nøkkelpar. Et Ed25519-nøkkelpar genereres på iOS-, iPadOS- og macOS-enheten for den enkelte bruker for HomeKit, og nøkkelparet blir brukerens HomeKit-identitet. Det brukes til å autentisere kommunikasjon mellom iOS-, iPadOS- og macOS-enheter og mellom iOS-, iPadOS- og macOS-enheter og tilbehør.

Nøklene, som oppbevares i nøkkelringen og kun inkluderes i krypterte Nøkkelring-sikkerhetskopier, holdes oppdatert mellom enheter ved hjelp av iCloud-nøkkelring, hvis tilgjengelig. HomePod og Apple TV mottar nøkler ved hjelp av trykk-for-å-konfigurere eller konfigureringsmodusen beskrevet under. Nøkler deles fra en iPhone til en sammenkoblet Apple Watch ved hjelp av Apple Identity Service (IDS).

Kommunikasjon mellom HomeKit-tilbehør

HomeKit-tilbehør genererer sitt eget Ed25519-nøkkelpar til bruk i kommunikasjon med iOS-, iPadOS- og macOS-enheter. Hvis tilbehøret gjenopprettes til fabrikkinnstillingene, genereres et nytt nøkkelpar.

For å etablere et forhold mellom en iOS-, iPadOS- og macOS-enhet og et HomeKit-tilbehør utveksles nøkler ved hjelp av protokollen Secure Remote Password (3072-bit). Det benyttes en åttesifret kode som oppgis av tilbehørsprodusenten og som oppgis på iOS- eller iPadOS-enheten av brukeren, og deretter krypteres de ved hjelp av ChaCha20-Poly1305 AEAD med HKDF-SHA512-avledede nøkler. Tilbehørets MFi-sertifisering stadfestes også i oppsettprosessen. Tilbehør uten en MFi-brikke kan integrere støtte for programvareautentisering i iOS 11.3 eller nyere.

Når iOS-, iPadOS- og macOS-enheten og HomeKit-tilbehøret kommuniserer under bruk, autentiserer de hverandre ved hjelp av nøklene som ble utvekslet i prosessen ovenfor. Den enkelte økten opprettes ved hjelp av Station-to-Station-protokollen og krypteres med HKDF-SHA512-avledede nøkler basert på øktspesifikke Curve25519-nøkler. Dette gjelder for både IP-basert tilbehør og Bluetooth Low Energy-tilbehør (BLE).

For BLE-enheter som støtter kringkastingsvarslinger, klargjøres tilbehøret med en kringkastingskrypteringsnøkkel fra en sammenkoblet iOS-, iPadOS- eller macOS-enhet gjennom en sikker økt. Denne nøkkelen brukes til å kryptere data om tilstandsendringer på tilbehøret, som varsles ved hjelp av BLE-annonseringer. Kringkastingskrypteringsnøkkelen er en HKDF-SHA512-avledet nøkkel, og dataene krypteres med ChaCha20-Poly1305 AEAD-algoritmen. Kringkastingskrypteringsnøkkelen endres periodisk av iOS-, iPadOS- og macOS-enheten og oppdateres til andre enheter ved hjelp av iCloud som beskrevet i HomeKit-datasikkerhet.

HomeKit og Siri

Siri kan brukes til å sende forespørsler til tilbehør, styre det og til å aktivere stemninger. Minimalt med informasjon om konfigurasjonen av hjemmet gis anonymt til Siri for at Siri skal ha navn på rom, tilbehør og stemninger som er nødvendige for kommandogjenkjenning. Det kan hende at lyd som sendes til Siri, betegner spesifikt tilbehør eller spesifikke kommandoer, men slike Siri-data knyttes ikke til andre Apple-funksjoner som for eksempel HomeKit.

HomeKit-datasikkerhet

HomeKit-dataoppdateringer mellom enheter og brukere

HomeKit-data kan oppdateres på en brukers iOS-, iPadOS- og macOS-enheter ved hjelp av iCloud og iCloud-nøkkelring. Under denne prosessen krypteres HomeKit-dataene ved å bruke nøkler avledet fra brukerens HomeKit-identitet og en tilfeldig nonce-verdi og håndteres som en såkalt «opaque binary large object», eller *blob*. Den nyeste «bloben» lagres i iCloud, men brukes ikke til andre formål. Det er ikke mulig å få tilgang til innholdet under overføring og lagring på iCloud fordi det krypteres med nøkler som kun er tilgjengelige på brukerens iOS-, iPadOS- og macOS-enheter.

HomeKit-data synkroniseres også mellom flere brukere i samme hjem. Denne prosessen bruker samme autentisering og kryptering som den som brukes mellom en iOS-, iPadOS- og macOS-enhet og et HomeKit-tilbehør. Autentiseringen baseres på offentlige Ed25519-

nøkler som utveksles mellom enhetene når en bruker legges til i et hjem. Etter at en ny bruker er lagt til i et hjem, blir all videre kommunikasjon autentisert og kryptert ved hjelp av Station-to-Station-protokollen og øktspesifikke nøkler.

Nye brukere kan legges til av brukeren som opprinnelig opprettet hjemmet i HomeKit, eller av en annen bruker med redigeringstillatelse. Eierens enhet konfigurerer tilbehøret med den nye brukerens offentlige nøkkel, slik at tilbehøret kan autentisere og ta imot kommandoer fra den nye brukeren. Når en bruker med redigeringstillatelse legger til en ny bruker, delegeres prosessen til et hjemknutepunkt for at operasjonen skal bli fullført.

Prosessen med å klargjøre Apple TV for bruk med HomeKit utføres automatisk når brukeren logger på iCloud. Tofaktorautentisering må være aktivert for iCloud-kontoen. Apple TV og eierens enhet utveksler midlertidige offentlige Ed25519-nøkler via iCloud. Hvis eierens enhet og Apple TV er på det samme lokale nettverket, brukes de midlertidige nøklene til å sørge for en sikker forbindelse via det lokale nettverket ved hjelp av Station-to-Station-protokollen og øktspesifikke nøkler. Denne prosessen bruker samme autentisering og kryptering som den som brukes mellom en iOS-, iPadOS- og macOS-enhet og et HomeKit-tilbehør. Eierens enhet overfører brukerens offentlige-private Ed25519-nøkkelpar til Apple TV via denne sikre lokale forbindelsen. Disse nøklene brukes så til å sikre kommunikasjonen mellom Apple TV og HomeKit-tilbehør og også mellom Apple TV og andre iOS-, iPadOS- og macOS-enheter som er en del av HomeKit-hjemmet.

Hvis en bruker ikke har flere enheter og ikke gir andre brukere tilgang til hjemmet sitt, overføres ingen HomeKit-data til iCloud.

Hjemdata og -apper

Appenes tilgang til hjemdata styres av brukerens innstillinger for personvern. Brukerne blir bedt om å gi tilgang når apper spør etter hjemdata, på samme måte som når de spør etter data fra Kontakter, Bilder og andre iOS-, iPadOS- og macOS-kilder. Hvis brukeren tillater det, har apper tilgang til navn på rom, navn på tilbehør, hvilket rom tilbehøret befinner seg i, og annen informasjon slik den er beskrevet i utviklerdokumentasjonen for HomeKit på https://developer.apple.com/homekit/.

Lokal datalagring

HomeKit lagrer data om hjemmene, tilbehøret, scenene og brukerne på brukerens iOS-, iPadOS- og macOS-enheter. Disse lagrede dataene krypteres ved hjelp av nøkler avledet fra brukerens nøkler for HomeKit-identiteten, pluss en tilfeldig «nonce»-verdi (engangskode). HomeKit-data lagres også ved hjelp av databeskyttelsesklassen «Beskyttet til første brukerautentisering». HomeKit-data sikkerhetskopieres bare til krypterte sikkerhetskopier. Derfor vil for eksempel ukrypterte sikkerhetskopier til Finer (macOS 10.15 eller nyere) eller iTunes (i macOS 10.14 eller eldre) via USB ikke inneholde HomeKit-data.

Sikring av rutere med HomeKit

Rutere som støtter HomeKit, lar brukere forbedre sikkerheten på hjemmenettverket ved å administrere Wi-Fi-tilgangen som HomeKit-tilbehør har til det lokale nettverket og internett. Ruterne støtter også privat PPSK-autentisering, slik at tilbehør kan legges til på Wi-Fi-nettverket ved å bruke en nøkkel som er spesifikk for tilbehøret, og som kan tilbakekalles ved behov. PPSK-autentisering forbedrer sikkerheten ved at Wi-Fi-hovedpassordet ikke blir tilgjengelig for tilbehør, i tillegg til at ruteren kan identifisere et tilbehør på en sikker måte selv om det endrer MAC-adresse.

Med Hjem-appen kan en bruker konfigurere tilgangsrestriksjoner for grupper med tilbehør som følger:

- *Ingen begrensning:* Tillat ubegrenset tilgang til internett og det lokale nettverket.
- *Automatisk:* Dette er standardinnstillingen. Tillat tilgang til internett og det lokale nettverket basert på en liste med nettsteder og lokale porter gitt til Apple av tilbehørsprodusenten. Denne listen omfatter alle nettsteder og porter som tilbehøret trenger for å fungere slik det skal. (Ingen begrensning er aktivert frem til en slik liste er tilgjengelig.)
- *Begrens til Hjem:* Ingen tilgang til internett eller det lokale nettverket, med unntak av forbindelsene som kreves av HomeKit for å oppdage og styre tilbehøret fra det lokale nettverket (inkludert fra hjemknutepunktet for å støtte fjernstyring).

PPSK er et sterkt, tilbehørsspesifikt WPA2 Personal-passord som genereres automatisk av HomeKit, og tilbakekalles hvis og når tilbehøret senere fjernes fra hjemmet. PPSK brukes når et tilbehør legges til på Wi-Fi-nettverket av HomeKit i et Hjem som har blitt konfigurert med en HomeKit-ruter. Dette gjenspeiles som Wi-Fi-akkreditiv: HomeKit-administrert på innstillingsskjermen til tilbehøret i Hjem-appen. Tilbehør som ble lagt til på Wi-Fi-nettverket før ruteren ble lagt til, blir konfigurert på nytt til å bruke en PPSK hvis tilbehøret støtter dette. Ellers beholdes de eksisterende akkreditivene.

Som et ekstra sikkerhetstiltak må brukere konfigurere HomeKit-ruteren ved å bruke appen til ruterprodusenten, slik at appen kan validere at brukerne har tilgang til ruteren og kan legge den til i Hjem-appen.

Sikkerhet for HomeKit-kamera

Kameraer som har en IP-adresse (internettprotokoll) i HomeKit, sender video- og lydstrømmer direkte til iOS-, iPadOS-, tvOS og macOS-enheten på lokalnettverket som har tilgang til strømmen. Strømmene krypteres ved hjelp av tilfeldig genererte nøkler på enheten og et IP-kamera, og de utveksles over den sikre HomeKit-økten til kameraet. Når en enhet ikke er på det lokale nettverket, sendes de krypterte strømmene via hjemknutepunktet til enheten. Hjemknutepunktet dekrypterer ikke strømmene og fungerer kun som en forbindelse mellom enheten og IP-kameraet. Når en app viser HomeKit IP-kameravideoen til brukeren, gjengir HomeKit videobildene sikkert fra en separat systemprosess. Dermed får ikke appen tilgang til eller kan lagre videostrømmen. I tillegg har ikke apper tillatelse til å ta skjermbilder fra denne strømmen.

Sikker video med HomeKit

HomeKit er en gjennomgående sikker og privat plattform som kan brukes til opptak, analysering og visning av klipp fra HomeKit IP-kameraer uten at videoinnholdet legges åpent for Apple og tredjeparter. Når bevegelse registreres av IP-kameraet, sendes videoklipp direkte til en Apple-enhet som fungerer som et hjemknutepunkt, ved hjelp av en separat lokal nettverksforbindelse mellom hjemknutepunktet og IP-kameraet. Den lokale nettverksforbindelsen krypteres med et øktspesifikt HKDF-SHA512-utledet nøkkelpar som formidles i HomeKit-økten mellom hjemknutepunktet og IP-kameraet. HomeKit dekrypterer lyd- og videostrømmene på hjemknutepunktet og analyserer videobildene lokalt for eventuelle viktige hendelser. Hvis det oppdages en viktig hendelse, krypterer HomeKit videoklippet ved hjelp av AES-256-GCM med en AES256-nøkkel som genereres tilfeldig. HomeKit genererer også stillbilder for hvert klipp, og disse krypteres ved hjelp av

den samme AES256-nøkkelen. Det krypterte stillbildet og lyd- og videodata lastes opp til iCloud-tjenere. De tilknyttede metadataene for hvert klipp, inkludert krypteringsnøkkelen, lastes opp til CloudKit ved hjelp av gjennomgående iCloud-kryptering.

For ansiktsklassifisering lagrer HomeKit alle data som brukes til å klassifisere en bestemt persons ansikt, i CloudKit ved hjelp av gjennomgående iCloud-kryptering. De lagrede dataene inkluderer informasjon om hver person, for eksempel navn, i tillegg til bilder som representerer personens ansikt. Disse ansiktsbildene kan hentes fra en brukers Bilder hvis de velger det, eller de kan samles inn fra tidligere analysert video fra IP-kameraer. En analyseøkt fra sikker video med HomeKit bruker disse klassifiseringsdataene til å identifisere ansikter i den sikre videostrømmen den mottar direkte fra IP-kameraet, og den inkluderer denne identifikasjonsinformasjonen i klippmetadataene nevnt tidligere.

Når Hjem-appen brukes til å vise klippene fra et kamera, lastes dataene ned fra iCloud, og nøkler for dekryptering av strømmene pakkes ut lokalt ved hjelp av gjennomgående iCloud-kryptering. Det krypterte videoinnholdet strømmes fra serverne og dekrypteres lokalt på iOS-enheten før det vises. Hver videoklippøkt kan brytes ned i underdeler der hver underdel krypterer innholdsstrømmen med sin egen unike nøkkel.

HomeKit-sikkerhet med Apple TV

Bruke fjernkontrolltilbehør fra tredjeparter med Apple TV

Enkelte fjernkontrolltilbehør fra tredjeparter leverer HID-hendelser og Siri-lyd til en tilknyttet Apple TV som er lagt til via Hjem-appen. Fjernkontrollen sender HID-hendelsene over den sikre økten til Apple TV-enheten. En TV-fjernkontroll med Siri-støtte sender lyddata til Apple TV når brukeren uttrykkelig aktiverer mikrofonen på fjernkontrollen ved hjelp av en egen Siri-knapp. Fjernkontrollen sender lyden direkte til Apple TV ved hjelp av en separat lokal nettverksforbindelse. Det brukes et øktspesifikt HKDF-SHA512-avledet nøkkelpar som formidles i HomeKit-økten mellom Apple TV og TV-fjernkontrollen, til å kryptere den lokale nettverksforbindelsen. HomeKit dekrypterer lyden på Apple TV og videresender dem til Siri-appen, der de behandles med den samme personvernsbeskyttelsen som all Siri-lydinndata.

Apple TV-profiler for HomeKit-hjem

Når en bruker i et HomeKit-hjem legger profilen sin til eieren av hjemmets Apple TV, får brukeren tilgang til TV-programmene, musikken og podkastene. Innstillinger for hver brukers profilbruk på Apple TV deles til eierens iCloud-konto ved hjelp av gjennomgående iCloud-kryptering. Dataene eies av hver bruker, og de deles skrivebeskyttet til eieren. Hver bruker av hjemmet kan endre disse verdiene i Hjem-appen, og eierens Apple TV bruker disse innstillingene.

Når en innstilling slås på, vil brukerens iTunes-konto bli tilgjengelig på Apple TV. Når en innstilling slås av, vil brukerens konto og data slettes fra Apple TV. Den første CloudKit-delingen startes av brukerens enhet, og kjennetegnet for å opprette sikker CloudKit-deling sendes via samme sikre kanal som brukes til å synkronisere data mellom brukerne av hjemmet.

HomeKit-tilbehør og iCloud

Merk: Bruk tjenester direkte via et hjemknutepunkt i stedet for gjennom iCloud når det er mulig. Bruk for eksempel et hjemknutepunkt som HomePod, Apple TV eller iPad.

iCloud-fjerntilgang støttes fortsatt for eldre HomeKit-enheter. Apple utformet disse enhetene nøye slik at brukerne kan styre dem og sende varslinger til dem uten at Apple får vite hva slags tilbehør det er eller hvilke kommandoer og varslinger som sendes. HomeKit sender aldri informasjon om hjemmet via iCloud-fjerntilgang.

Gjensidig autentisering av et tilbehør og en Apple-enhet

Når en bruker sender en kommando ved hjelp av iCloud-fjerntilgang, blir tilbehøret og iOS-, iPadOS- og macOS-enheten gjensidig autentisert, og data krypteres ved hjelp av den samme prosedyren som den som er beskrevet for lokale tilkoblinger. Innholdet i det som kommuniseres krypteres og er ikke synlig for Apple. Adressetildelingen via iCloud foretas basert på iCloud-ID-ene som ble registrert i oppsettprosessen.

Oppsettprosess for tilbehør

Tilbehør som har støtte for iCloud-fjerntilgang klargjøres i oppsettprosessen for tilbehøret. Klargjøringsprosessen begynner med at brukeren logger på iCloud. Deretter ber iOS- og iPadOS-enheten tilbehøret om å signere en utfordring ved hjelp av Apple Authentication Coprocessor som er bygd inn i alt Built for HomeKit-tilbehør. Tilbehøret genererer også prime256v1-nøkler basert på en elliptisk kurve, og den offentlige nøkkelen sendes til iOS- og iPadOS-enheten sammen med den signerte utfordringen og X.509-sertifikatet til autentiserings-koprosessoren. Dette brukes til å be om et sertifikat for tilbehøret fra iClouds klargjøringstjener. Sertifikatet lagres av tilbehøret, men det inneholder ingen informasjon som kan identifisere tilbehøret, med unntak av at det har fått tilgang til iCloud-fjerntilgang for HomeKit. iOS- og iPadOS-enheten som utfører klargjøringen, sender også et etui til tilbehøret. Etuiet inneholder URL-ene og annen informasjon som er nødvendig for å opprette forbindelse til tjeneren for iCloud-fjerntilgang. Informasjonen er ikke spesifikk for bruker eller tilbehør.

Tilbehørsliste over tillatte brukere

Alt tilbehør fører en liste over tillatte brukere på tjeneren for iCloud-fjerntilgang. Disse brukerne har blitt gitt muligheten til å styre tilbehøret av brukeren som la til tilbehøret i hjemmet. Brukerne gis en ID av iCloud-tjeneren og kan bli tilordnet en iCloud-konto for levering av varslinger og svar fra tilbehøret. Tilbehør har på tilsvarende måte ID-er utstedt av iCloud, men disse ID-ene er ugjennomsiktige og avslører ingen informasjon om selve tilbehøret.

Hvordan tilbehør kobler til tjeneren for iCloud-fjerntilgang

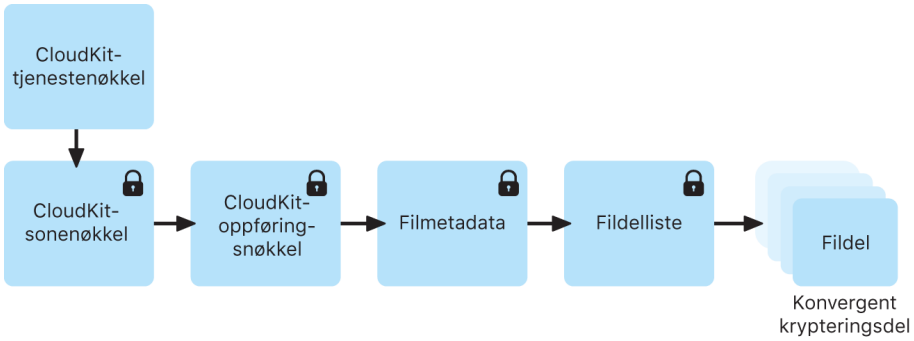
Når tilbehør oppretter forbindelse til tjeneren for iCloud-fjerntilgang for HomeKit, viser det fram sertifikatet og et adgangsbevis. Adgangsbeviset innhentes fra en annen iCloud-tjener og er ikke unikt for det enkelte tilbehøret. Når tilbehør ber om adgangsbevis, tar det med produsent, modell og firmwareversjon i forespørselen. Ingen informasjon som kan identifisere bruker eller hjem, tas med i denne forespørselen. For å beskytte personvernet autentiseres ikke forbindelsen til adgangsbevistjeneren.

Tilbehør kobles til tjeneren for iCloud-fjerntilgang ved hjelp av HTTP/2, sikret ved hjelp av TLS 1.2 med AES128-GCM og SHA256. Tilbehøret holder forbindelsen til tjeneren for iCloud-fjerntilgang åpen slik at det kan motta innkommende meldinger og sende svar og utgående varslinger til iOS-, iPadOS- og macOS-enheter.

CloudKit-sikkerhet

CloudKit er et rammeverk der apputviklere kan lagre nøkkelverdidata, strukturerte data og ressurser i iCloud. Tilgangen til CloudKit kontrolleres ved hjelp av apprettigheter. CloudKit støtter både offentlige og private databaser. Offentlige databaser brukes av alle eksemplarene av appen, vanligvis for generelle ressurser, og krypteres ikke. Private databaser lagrer brukerens data.

I likhet med iCloud Drive bruker også CloudKit kontobaserte nøkler for å beskytte informasjonen som er lagret i brukerens private database, og filer blir delt opp i deler, kryptert og lagret ved hjelp av tjenester fra tredjeparter, slik det gjøres i andre iCloud-tjenester. CloudKit benytter et nøkkelhierarki som ligner på databeskyttelsesfunksjonen. De filspesifikke nøklene pakkes med CloudKit Record-nøkler. Record-nøklene beskyttes så av en nøkkel som gjelder for hele sonen, som er beskyttet av brukerens CloudKit Service-nøkkel. CloudKit Service-nøkkelen lagres i brukerens iCloud-konto og er tilgjengelig kun etter at brukeren har gjennomført en autentiseringsprosess med iCloud.



Gjennomgående kryptering i CloudKit.

SiriKit-sikkerhet for iOS, iPadOS og watchOS

Siri bruker funksjonaliteten for tillegg for apper til å kommunisere med tredjepartsapper. På en enhet kan Siri kan få tilgang til brukerens kontaktinformasjon og enhetenes aktuelle plassering. Men før beskyttede data leveres til en app, kontrollerer Siri appens brukerstyrte tilgangsrettigheter. Basert på rettighetene leverer Siri kun det relevante fragmentet av den opprinnelige brukerdelen til apptillegget. Hvis appen for eksempel ikke har tilgang til kontaktinformasjon, behandler ikke Siri en relasjon i en brukerforespørsel, for eksempel «Betal 100 kr til mamma med Betalingsapp». I dette eksempelet ser Siri bare den bokstavelige betegnelsen «mamma».

Men hvis brukeren har gitt appen tilgang til kontaktinformasjon, får appen informasjon om brukerens mor. Hvis en relasjon har blitt omtalt i hoveddelen i en meldingstekst, for eksempel «Skriv til mamma i meldingsapp at broren min er fantastisk», behandler ikke Siri «broren min» uansett hvilken tilgang appen har.

SiriKit-aktiverte apper kan sende appspesifikt eller brukerspesifikt vokabular til Siri, for eksempel navnet på brukerens kontakter. Med denne informasjonen kan Siris talegjenkjenning og naturlige språkforståelse gjenkjenne vokabular for den aktuelle appen og det knyttes til en tilfeldig ID. Den tilpassede informasjonen er tilgjengelig så lenge ID-en er i bruk, eller til brukeren deaktiverer appens Siri-integrering i Innstillinger, eller til den SiriKit-aktiverte appen avinstalleres.

For en setning som «Bestill transport hjem til mamma med TransportApp», krever forespørselen stedsdata fra brukerens kontakter. For denne aktuelle forespørselen gir Siri nødvendig informasjon til appens tillegg, uavhengig av appens brukertillatelse for stedsinformasjon eller kontaktinformasjon.

DriverKit-sikkerhet for macOS 10.15

DriverKit er rammeverket som lar utviklere lage enhetsdrivere som brukeren installerer på Macen. Drivere som er laget med DriverKit, kjøres i brukerområdet i stedet for som kjerneutvidelser, noe som gir bedre systemsikkerhet og stabilitet. Dette forenkler installasjonen og øker stabiliteten og sikkerheten i macOS.

Brukeren laster ned appen (det er ikke behov for installerere når systemtillegg eller DriverKit brukes), og tillegget er kun aktivert når det er behov for det. Disse erstatter kext-er for mange brukertilfeller som krever administratorrettigheter for å kunne installeres i /System/Bibliotek eller /Bibliotek.

IT-administratorer som bruker enhetsdrivere, løsninger for skylagring, nettverks- og sikkerhetsprogrammer som krever kjerneutvidelser, bør oppgradere til nyere versjoner som er basert på systemtillegg. Disse nyere versjonene reduserer i stor grad muligheten for kjernepanikk på Macen, samtidig som de reduserer angrepsflaten. Disse nye tilleggene kjøres i brukerområdet, krever ikke spesiell tilgang for installasjon og de fjernes automatisk når appen flyttes til papirkurven.

DriverKit-rammeverket har C++-klasser for I/O-tjenester, enhetssamsvar, minnedeskriptorer og køer for sending. Det definerer også I/O-egnede typer for tall, samlinger, strenger og andre vanlige typer. Brukeren bruker disse med familie-spesifikke rammeverk for drivere, som USBDriverKit og HIDDriverKit. Bruk System Extensions-rammeverket til å installere og oppgradere en driver.

ReplayKit-sikkerhet i iOS og iPadOS

ReplayKit er et betarammeverk som gjør det mulig for utviklere å legge til funksjonalitet for opptak og direktesending i appene. Dessuten gir det brukerne mulighet til å kommentere opptakene og sendingene ved hjelp av enhetens framovervendte kamera og mikrofon.

Filmopptak

Det er flere innebygde sikkerhetslag for filmopptak:

- *Tillatelsesdialogruten:* Før opptaket starter viser ReplayKit et varsel der brukeren blir bedt om å bekrefte at han eller hun ønsker å gjøre opptak fra skjermbildet, mikrofonen og det framovervendte kameraet. Dette varselet vises én gang for hver app-prosess, og det vises på nytt hvis appen ligger i bakgrunnen i mer enn 8 minutter.

- *Bilde- og lydopptak:* Bilde- og lydopptak foregår utenfor appens prosess i ReplayKits replayd-daemon. Dette er laget for å sikre at innholdet som tas opp, aldri er tilgjengelig for app-prosessen.
- *Bilde- og lydopptak i appen:* Gir appen tilgang til video- og prøvebuffer, som beskyttes av tillatelsesdialogruten.
- *Lage og lagre film:* Filmfilen skrives til en katalog som kun er tilgjengelig for ReplayKits undersystemer, og som aldri er tilgjengelig for apper. Dette bidrar til å forhindre at opptak brukes av tredjeparter uten brukerens samtykke.
- *Forhåndsvise og dele med sluttbruker:* Brukeren kan forhåndsvise og dele filmen med et brukergrensesnitt som gjøres tilgjengelig av ReplayKit. Brukergrensesnittet presenteres utenfor prosessen via tilleggsinfrastrukturen i iOS og har tilgang til den genererte filmfilen.

ReplayKit-kringkasting

Det er flere innebygde sikkerhetslag for kringkasting av en film:

- *Bilde- og lydopptak:* Det er samme bilde- og lydopptaksfunksjonalitet som brukes ved kringkasting som ved filmopptak, og den foregår i `replayd`.
- *Kringkastingstillegg:* For at tredjepartstjenester skal kunne delta i ReplayKit-kringkasting, kreves det at de oppretter to nye tillegg som er konfigurert med com.apple.broadcast-services-sluttpunktet:
  - Et brukergrensesnittillegg som tillater brukeren å sette opp kringkastingen
  - Et opplastingstillegg som håndterer opplasting av video- og lyddata til tjenestens tjenersystemer

Arkitekturen bidrar til å sikre at vertsapper ikke har noen rettigheter til kringkastet video- og lydinnhold. Kun ReplayKit og kringkastingstilleggene fra tredjeparter har tilgang.

- *Kringkastervelger:* Med kringkastingsvelgeren kan brukere starte systemkringkasting direkte fra appen ved hjelp av det samme systemdefinerte brukergrensesnittet som er tilgjengelig via Kontrollsenter. Grensesnittet implementeres ved hjelp av en privat API og er et tillegg som opererer innenfor ReplayKit-rammeverket. Den er utenfor prosessen fra vertsappen.
- *Opplastingstillegg:* Tillegget som kringkastingstjenester fra tredjeparter implementerer for å håndtere video- og lydinnhold når kringkasting pågår, bruker rå, ikke-kodede prøvebuffere. I denne håndteringsmodusen blir video- og lyddata serialisert og videresendt til opplastingstillegget fra tredjeparten i sanntid via en direkte XPC-tilkobling. Videodata kodes ved at IOSurface-objektet trekkes ut fra videoprøvebufferen, kodes på en sikker måte som et XPC-objekt, sendes via XPC til tredjepartstillegget og dekodes på en sikker måte tilbake til et IOSurface-objekt.

ARKit-sikkerhet i iOS og iPadOS

ARKit er et rammeverk der utviklere kan produsere opplevelser med utvidet virkelighet i en app eller et spill. Utviklere kan legge til 2D- eller 3D-elementer ved å bruke kameraene foran eller bak på en iOS- eller iPadOS-enhet.

Apple utviklet kameraer med fokus på personvern, og tredjepartsapper må få samtykke fra brukeren før de får tilgang til kameraet. I iOS og iPadOS får apper som brukeren gir tilgang til kameraet, tilgang til sanntidsbilder fra kameraer foran og bak på enheten. Apper kan ikke bruke kameraet uten å informere om at kameraet brukes.

Bilder og videoer som tas med kameraet, kan inneholde annen informasjon, for eksempel om hvor og når de ble tatt, dybdeskarphet og bilde utenfor utsnittet. Hvis brukerne ikke ønsker at bilder og videoer som tas med Kamera-appen, skal inneholde stedsinformasjon, kan dette velges i Innstillinger > Personvern > Stedstjenester > Kamera. Hvis brukere ikke ønsker at bilder og videoer skal inneholde stedsinformasjon når de deles, kan dette velges i Valg-menyen fra delingsarket.

For å bedre plassere brukerens AR-opplevelse kan apper som bruker ARKit, bruke verdens- eller ansiktssporingsinformasjon fra det andre kameraet. Verdenssporing bruker algoritmer på brukerens enhet til å behandle informasjon fra disse sensorene for å finne posisjonen i forhold til de fysiske omgivelsene. Med verdenssporing får man tilgang til funksjoner som Optisk retning i Kart.

Sikker administrering av enheter

Oversikt over sikker administrering av enheter

iOS, iPadOS, macOS og tvOS støtter fleksible sikkerhetsregler og konfigurasjoner som er enkle å håndheve og administrere. Ved hjelp av disse kan bedrifter beskytte virksomhetsinformasjon og bidra til å sikre at de ansatte følger bedriftens regelverk selv om de bruker sine egne enheter, for eksempel som en del av et BYOD-program (Bring Your Own Device).

Bedrifter kan bruke ressurser som for eksempel passordbeskyttelse, konfigurasjonsprofiler, fjernsletting og MDM-løsninger fra tredjeparter for å administrere mange enheter og bidra til at bedriftsinformasjon sikres selv når de ansatte bruker sine egne enheter for å få tilgang til informasjonen.

I iOS 13 eller nyere, iPadOS 13.1 eller nyere og macOS 10.15 eller nyere har Apple-enheter støtte for en ny form for brukerregistrering spesielt utviklet for BYOD-programmer. Brukerregistrering gir brukerne flere muligheter med sine egne enheter, samtidig som det øker sikkerheten for bedriftens data ved å lagre dem på et separat, kryptografisk beskyttet APFS-volum (Apple File System). Det gir BYOD-programmer en bedre balanse mellom sikkerhet, personvern og brukeropplevelse.

Sikkerhet for sammenkoblingsmodell for iPhone og iPad

iOS og iPadOS bruker en sammenkoblingsmodell for å kontrollere tilgangen til en enhet fra en vertsdatamaskin. Sammenkobling etablerer en godkjent relasjon mellom enheten og den tilkoblede verten, signalisert ved utveksling av offentlig nøkkel. iOS og iPadOS bruker også dette tegnet på tillit til å aktivere ytterligere funksjonalitet med den tilkoblede verten, for eksempel datasynkronisering. I iOS 9 eller nyere, vil tjenester:

- som krever sammenkobling ikke kunne startes før enheten er blitt låst opp av brukeren
- ikke starte med mindre enheten nylig er blitt låst opp
- (for eksempel med synkronisering av bilder) kunne kreve at enheten låses opp før den starter

Sammenkoblingsprosessen krever at brukeren låser opp enheten og godtar sammenkoblingsforespørselen fra verten. I iOS 9 og nyere kreves det også at brukeren oppgir koden på nytt før verten og enheten utveksler og lagrer 2048-bit offentlige RSA-nøkler. Verten får deretter en 256-bit-nøkkel som kan låse opp et deponeringsnøkkeletui som er lagret på enheten. Nøklene som utveksles, brukes til å starte en kryptert SSL-økt,

som enheten krever før den sender beskyttede data til verten eller starter en tjeneste (iTunes- eller Finder-synkronisering, filoverføringer, Xcode-utvikling og så videre). For å bruke denne krypterte økten for all kommunikasjon krever enheten forbindelser fra en vert via Wi-Fi. De må derfor ha blitt sammenkoblet tidligere over USB. Sammenkobling aktiverer også flere diagnosefunksjoner. I iOS 9 utløper sammenkoblingsoppføringer hvis de ikke har vært i bruk på mer enn seks måneder. I iOS 11 eller nyere er denne tidsperioden forkortet til 30 dager.

Det er satt restriksjoner for enkelte diagnostiske tjenester, inkludert com.apple.mobile.pcapd, slik at de kun fungerer via USB. Dessuten krever com.apple.file_relay-tjenesten en Apple-signert konfigurasjonsprofil for å bli installert. Fra og med iOS 11 kan Apple TV bruke Secure Remote Password-protokollen til å trådløst etablere et sammenkoblingsforhold.

En bruker kan tømme listen over godkjente verter ved å bruke valgene Nullstill nettverksinnstillinger eller Nullstill Sted og personvern.

MDM (Mobile Device Management)

Oversikt over MDM-sikkerhet

Oversikt

Apple-operativsystemer støtter MDM som lar organisasjoner konfigurere og administrere skalert utrulling av Apple-enheter på en sikker måte. MDM-funksjonene er bygd på operativsystemteknologi som for eksempel konfigurasjonsprofiler, trådløs registrering og Apple Push Notification service (APNs). APNs brukes for eksempel til å vekke enheten, slik at den kan kommunisere direkte med MDM-løsningen over en sikker tilkobling. Med APNs overføres det ingen konfidensiell eller beskyttet informasjon.

Ved hjelp av MDM kan IT-avdelingen registrere Apple-enheter i et bedriftsmiljø på en trygg måte, konfigurere og oppdatere innstillinger trådløst, følge med på om bedriftens regler overholdes, administrere programvareoppdateringsregler og til og med fjernslette eller fjernlåse administrerte enheter.

I tillegg til de tradisjonelle enhetsregistreringene som støttes av iOS, iPadOS, macOS og tvOS, har en registreringstype blitt lagt til i iOS 13 eller nyere, iPadOS 13.1 eller nyere og macOS 10.15 eller nyere – brukerregistrering. Brukerregistreringer er MDM-registreringer som er tilpasset BYOD-utrulling (Bring Your Own Device) der brukeren selv eier enheten, men bruker den i et administrert miljø. Brukerregistrering gir MDM-løsningen med mer begrensede rettigheter enn ved enhetsregistrering uten tilsyn, og sørger for kryptografisk skille mellom brukerens og bedriftens data.

Registreringstyper

- *Brukerregistrering:* Brukerregistrering er designet for enheter som eies av brukeren, og er integrert med administrerte Apple-ID-er for å etablere en brukeridentitet på enheten. Administrerte Apple-ID-er er en del av Brukerregistrering-profilen, og brukeren må lykkes med autentisering for å fullføre registreringen. Administrerte Apple-ID-er kan brukes sammen med en personlig Apple-ID som brukeren allerede har logget på med. Administrerte apper og kontoer bruker en administrert Apple-ID, og personlige apper og kontoer bruker en personlig Apple-ID.

- *Enhetsregistrering:* Med Enhetsregistrering kan organisasjoner be brukere om å registrere enheter manuelt og deretter administrere mange forskjellige sider ved enhetsbruk, inkludert muligheten til å slette enheten. Enhetsregistrering har også er større sett med nyttelaster og restriksjoner som kan anvendes på enheten. Når en bruker fjerner en registreringsprofil, fjernes også alle konfigurasjonsprofiler, tilhørende innstillinger og administrerte apper basert på registreringsprofilen.
- *Automatisert enhetsregistrering:* Automatisert enhetsregistrering gjør det mulig for organisasjoner å konfigurere og administrere enheter fra det øyeblikket enhetene tas ut av esken (i en prosess som kalles *Auto Advance-utrulling*). Disse enhetene kalles enheter *under tilsyn*, og brukerne har muligheten til å hindre at MDM-profilen fjernes av brukeren. Automatisert enhetsregistrering er laget for enheter som eies av organisasjonen.

Enhetsrestriksjoner

Restriksjoner kan aktiveres, og i noen tilfeller deaktiveres, av administratorer for å bidra til å hindre brukere i å få tilgang til en bestemt app, tjeneste eller funksjon på en iPhone, iPad, Mac eller Apple TV som er registrert i en MDM-løsning. Restriksjoner sendes til enheter i en restriksjonsnyttelast, som er en del av en konfigurasjonsprofil. Enkelte restriksjoner på en administrert iPhone kan speiles på en sammenkoblet Apple Watch.

Administrering av koder og passordinnstillinger

Det er satt som standard at brukerens kode kan defineres som en numerisk PIN-kode. På iOS- og iPadOS-enheter med Touch ID eller Face ID kreves minimum en firesifret kode. Lengre og mer kompliserte koder anbefales. De er vanskeligere å gjette og angripe.

Administratorer kan håndheve krav til komplekse koder og andre regler ved hjelp av MDM eller Microsoft Exchange ActiveSync, eller ved å kreve at brukerne installerer konfigurasjonsprofiler manuelt. Det kreves et administratorpassord for installering av koderegelnyttelasten i macOS. Noen koderegler kan kreve en bestemt kodelengde, sammensetning eller andre attributter.

Håndhevelse av konfigurasjonsprofil

Konfigurasjonsprofiler er den primære måten en MDM-løsning leverer og administrerer regler og restriksjoner på administrerte enheter. Hvis organisasjonene trenger å konfigurere et stort antall enheter eller sende mange tilpassede e-postinnstillinger, nettverksinnstillinger eller sertifikater til et stort antall enheter, er konfigurasjonsprofiler en trygg og sikker måte å gjøre det på.

Konfigurasjonsprofiler

En *konfigurasjonsprofil* er en XML-fil (slutter på .mobileconfig) som består av nyttelaster som laster innstillinger og godkjenningsinformasjon inn på Apple-enheter. Konfigurasjonsprofiler automatiserer konfigureringen av innstillinger, kontoer, restriksjoner og akkreditiver. Disse filene kan opprettes av en MDM-løsning eller Apple Configurator 2 eller de kan opprettes manuelt. Før organisasjoner sender en konfigurasjonsprofil til en Apple-enhet, må de registrere enheten i MDM-løsningen ved hjelp av en registreringsprofil.

Registreringsprofiler

En *registreringsprofil* er en konfigurasjonsprofil med en MDM-nyttelast som registrerer enheten i MDM-løsningen som er spesifisert for enheten. Dermed kan MDM-løsningen sende kommandoer og konfigurasjonsprofiler til enheten og sende forespørsler om bestemte aspekter ved enheten. Når en bruker fjerner en registreringsprofil, fjernes også alle konfigurasjonsprofiler, tilhørende innstillinger og administrerte apper basert på registreringsprofilen. Det kan kun være én registreringsprofil på en enhet om gangen.

Innstillinger for konfigurasjonsprofil

En konfigurasjonsprofil inneholder en rekke innstillinger i konkrete nyttelaster som kan angis, inkludert (men ikke begrenset til):

- Regler for kode og passord
- Begrensninger av enhetsfunksjoner (for eksempel at kameraet er deaktivert)
- Nettverksinnstillinger og VPN-innstillinger
- Microsoft Exchange-innstillinger
- Innstillinger for e-post
- Kontoinnstillinger
- Innstillinger for LDAP-katalogtjenesten
- Innstillinger for CalDAV-kalendertjenesten
- Akkreditiver og nøkler
- Programvareoppdateringer

Profilsignering og -kryptering

Konfigurasjonsprofiler kan signeres for å validere opprinnelsen, og krypteres for å bidra til å sikre integriteten og beskytte innholdet. Konfigurasjonsprofiler for iOS og iPadOS krypteres med Cryptographic Message Syntax (CMS) som angitt i RFC 5652, med støtte for 3DES og AES128.

Installering av profil

Brukerne kan installere konfigurasjonsprofiler direkte på enheten ved hjelp av Apple Configurator 2, eller de kan laste dem ned via Safari, fra vedlegg til en e-postmelding, overført via AirDrop eller Filer-appen i iOS og iPadOS, eller tilsendt trådløst via en MDM-løsning. Når en bruker konfigurerer en enhet i Apple School Manager eller Apple Business Manager, laster enheten ned og installerer en profil for MDM-registrering. Du finner mer informasjon om hvordan du fjerner profiler, under MDM-oversikt i MDM-innstillinger for IT-administratorer.

Merk: På enheter som er under tilsyn, kan konfigurasjonsprofiler også låses til en enhet. Dette er utviklet for å gjøre det umulig å fjerne dem, eller slik at de kun kan fjernes med kode. Ettersom mange organisasjoner eier sine egne iOS- og iPadOS-enheter, er det mulig å fjerne konfigurasjonsprofiler som binder en enhet til en MDM-løsning, men da fjernes også all administrert konfigurasjonsinformasjon, data og apper.

Automatisert enhetsregistrering

Organisasjoner kan registrere iOS-, iPadOS-, macOS- og tvOS-enheter automatisk i MDM uten at de må ha fysisk tilgang til enhetene eller klargjøre dem før brukerne får dem. Etter registreringen i en av tjenestene, kan administratorer logge på via nettsiden til tjenesten og koble programmet til MDM-løsningen. Deretter kan enhetene de kjøpte, tildeles brukere via MDM. Når enhetene skal konfigureres, kan sikkerheten for sensitive data økes ved at egnede sikkerhetstiltak implementeres. For eksempel:

- Krev at brukerne må utføre en autentisering som ledd i det innledende oppsettet i Apples oppsettassistent ved aktivering.
- Gjør tilgjengelig en midlertidig konfigurasjon som gir begrenset tilgang, og krev ytterligere konfigurering av enheten for å gi tilgang til sensitive data.

Etter at en bruker er tildelt, blir MDM-spesifikke konfigurasjoner, restriksjoner og kontroller installert automatisk. All kommunikasjon mellom enheter og Apple-tjenere krypteres under overføring via HTTPS (TLS).

Brukernes oppsettprosess kan dessuten forenkles ytterligere ved å fjerne enkelte trinn i oppsettassistenten for enheter, slik at brukerne kommer raskt i gang. Administratorer kan også bestemme om brukere kan fjerne MDM-profilen fra enheten og bidra til å sørge for at enhetsrestriksjoner er på plass gjennom hele enhetens livssyklus. Etter at enheten er pakket opp og aktivert, kan den registreres i organisasjonens MDM-løsning, og alle administreringsinnstillinger, apper og bøker installeres, som definert av MDM-administratoren.

Apple School Manager og Apple Business Manager

Apple School Manager og Apple Business Manager er tjenester for IT-administratorer som skal rulle ut Apple-enheter som organisasjonen har kjøpt direkte fra Apple eller gjennom Apple-autoriserte forhandlere eller operatører.

Når det brukes sammen med en MDM-løsning, kan administratorer forenkle oppsettprosessen for brukere, konfigurere enhetsinnstillinger og distribuere apper og bøker kjøpt i Apple School Manager eller Apple Business Manager. Apple School Manager kan også integreres med elevinformasjonssystemer (SIS) direkte eller ved hjelp av SFTP, og Apple School Manager og Apple Business Manager kan bruke SCIM (System for Cross-domain Identity Management) eller forent autentisering med Microsoft Azure Active Directory (Azure AD), slik at administratorer kan opprette kontoer raskt.

Enheter med iOS 11 eller nyere og tvOS 10.2 eller nyere kan også legges til i Apple School Manager og Apple Business Manager etter kjøp ved hjelp av Apple Configurator 2.

Apple opprettholder sertifikater i samsvar med standardene ISO/IEC 27001 og 27018, slik at Apple-kunder kan oppfylle de regulatoriske og kontraktsmessige forpliktelsene sine. Disse sertifiseringene gir kundene våre en uavhengig attestasjon av Apples informasjonssikkerhets- og personvernpraksis for aktuelle systemer. Du finner mer informasjon i Apple-kundestøtteartikkelen Apple Internet Services-sertifiseringer.

Merk: Du kan finne ut om et Apple-program er tilgjengelig i aktuelt land eller område i Apple-kundestøtteartikkelen Tilgjengelighet av Apple-programmer og -betalingsmåter for utdanning og bedrifter.

Tilsyn av enheter

Tilsyn innebærer vanligvis at enheten eies av organisasjonen, noe som gir dem ytterligere kontroll over enhetens konfigurering og restriksjoner.

iPhone- og iPad-enheter med iOS 5 eller nyere og Apple TV-enheter med tvOS 10.2 eller nyere blir satt under tilsyn ved å:

- bruke Apple Configurator 2 for å sette enheten under tilsyn

  Under denne prosessen slettes enheten og alle data.
- registrere enheten i en MDM-løsning og velge tilsyn som en del av registreringsprosessen

Macer kan settes under tilsyn hvis de:

- kjører macOS 11 registrert i MDM ved hjelp av enhetsregistrering
- oppgraderes til macOS 11, og registreringen i MDM *var* en brukergodkjent MDM-registrering
- kjører macOS 10.14.4 eller nyere og:
  - enhetenes serienumre vises i Apple School Manager eller Apple Business Manager
  - registreres i en MDM-løsning ved hjelp av Apple School Manager eller Apple Business Manager

Følgende enheter settes automatisk under tilsyn når de registreres i Apple School Manager eller Apple Business Manager:

- iPhone og iPod touch med iOS 13 eller nyere
- iPad med iPadOS 13.1 eller nyere
- Apple TV med tvOS 13 eller nyere
- Macer med macOS 10.14.4 eller nyere

Viktig: Hvis brukeren kjenner koden, kan manuelt installerte konfigurasjonsprofiler fjernes på iPhone- og iPad-enheter som ikke er under tilsyn, selv om valget er satt til «aldri». Manuelt installerte konfigurasjonsprofiler for Macer kan fjernes ved å bruke `profiles`-kommandolinjeverktøyet eller Systemvalg hvis brukeren kjenner brukernavnet og passordet til en administrator. Fra og med macOS 10.15, som på iOS og iPadOS, må profiler som er installert med MDM, fjernes med MDM, eller så fjernes de automatisk ved når de avregistreres fra MDM.

Aktiveringslås-sikkerhet

Hvordan Apple iverksetter Aktiveringslås varierer avhengig av hvorvidt enheten er en iPhone, iPad, Mac med Apple Silicon eller Intel-basert Mac med Apple T2-sikkerhetsbrikken.

Adferd på iPhone and iPad

På iPhone- og iPad-enheter håndheves Aktiveringslås gjennom aktiveringsprosessen etter skjermen med Wi-Fi-valg i oppsettassistenten i iOS og iPadOS. Når enheten indikerer at den aktiveres, sender den en forespørsel til en Apple-tjener for å få et aktiveringssertifikat. På dette tidspunktet ber enheter med Aktiveringslås brukeren om iCloud-akkreditivene til brukeren som aktiverte Aktiveringslås. Oppsettassistenten i iOS og iPadOS fortsetter ikke med mindre det innhentes et gyldig sertifikat.

Adferd på Macer med Apple Silicon

På Macer med Apple Silicon verifiserer LLB at det eksisterer en gyldig LocalPolicy for enheten og at nonce-verdiene til LocalPolicy-regelen stemmer overens med verdiene lagret i komponenten for sikker lagring. LLB starter opp til recoveryOS hvis:

- det ikke finnes en LocalPolicy for den nåværende macOS-versjonen
- LocalPolicy er ugyldig for den macOS-versjonen
- noncehash-verdiene i LocalPolicy ikke stemmer overens med hashene for verdiene lagret i komponenten for sikker lagring

recoveryOS oppdager at Macen ikke er aktivert og kontakter aktiveringstjeneren for å få et aktiveringssertifikat. Hvis enheten har Aktiveringslås, ber recoveryOS på dette tidspunktet brukeren om iCloud-akkreditivene til brukeren som aktiverte Aktiveringslås. Etter at et gyldig aktiveringssertifikat er innhentet, brukes denne aktiveringssertifikatnøkkelen til å innhente et RemotePolicy-sertifikat. Macen bruker LocalPolicy-nøkkelen og RemotePolicy-sertifikat til å lage en gyldig LocalPolicy. LLB tillater ikke oppstart av macOS med mindre det finnes en gyldig LocalPolicy.

Adferd på Intel-baserte Macer

På Intel-baserte Macer med T2-brikke bekrefter firmwaren i T2-brikken at det finnes et gyldig aktiveringssertifikat før datamaskinen får lov til å starte opp til macOS. UEFI-firmware lastet av T2-brikken er ansvarlig for å sende forespørsler om aktiveringsstatusen til enheten fra T2-brikken og starte opp til recoveryOS i stedet for å starte opp til macOS hvis det ikke finnes et gyldig aktiveringssertifikat. recoveryOS oppdager at Macen ikke er aktivert og kontakter aktiveringstjeneren for å få et aktiveringssertifikat. Hvis enheten har Aktiveringslås, ber recoveryOS på dette tidspunktet brukeren om iCloud-akkreditivene til brukeren som aktiverte Aktiveringslås. UEFI-firmware tillater ikke oppstart av macOS med mindre det finnes et gyldig aktiveringssertifikat.

Administrert Mistet-modus og fjernsletting

Administrert Mistet-modus brukes til å finne enheter som er under tilsyn, når de blir stjålet. Når de blir funnet, kan de fjernlåses eller fjernslettes.

Administrert Mistet-modus

Hvis en iOS- eller iPadOS-enhet med iOS 9 som er under tilsyn, blir mistet eller stjålet, kan en MDM-administrator fjernaktivere Mistet-modus (kalt Administrert Mistet-modus) på enheten. Når Administrert Mistet-modus aktiveres, blir den nåværende brukeren logget av, og det er ikke mulig å låse opp enheten. På skjermen vises en melding som administratoren kan tilpasse, for eksempel et telefonnummer den som finner enheten kan ringe. Administratoren kan også be om at enheten sender sin nåværende plassering

(selv om Stedstjenester er av) og eventuelt spille av en lyd. Når en administrator slår av Administrert Mistet-modus, som er den eneste måten å få enheten ut av modusen på, blir brukeren informert om denne handlingen via en melding på låst skjerm eller et varsel på Hjem-skjermen.

Fjernsletting

En administrator eller bruker kan fjernslette iOS-, iPadOS- og macOS-enheter (umiddelbar fjernsletting er bare tilgjengelig hvis FileVault er aktivert på Macen). Umiddelbar fjernsletting utføres ved å kaste medienøkkelen fra Effaceable Storage på en sikker måte. Da blir alle data uleselige. Hvis fjernslettingen gjøres via Microsoft Exchange ActiveSync, kontakter enheten Microsoft Exchange-tjeneren før slettingen utføres.

Hvis en fjernslettingskommando kommer fra MDM eller iCloud, sender iPhone-, iPad-, iPod touch- eller Mac-enheten en bekreftelse til MDM-løsningen og utfører slettingen.

Fjernsletting er ikke mulig i følgende situasjoner:

- med Brukerregistrering
- via Microsoft Exchange ActiveSync når kontoen ble installert med Brukerregistrering
- Bruke Microsoft Exchange ActiveSync hvis enheten er under tilsyn

Brukere kan også slette iOS- og iPadOS-enheter de har råderetten over, ved hjelp av Innstillinger-appen. Og som nevnt kan iOS- og iPadOS-enheter stilles inn på automatisk sletting etter at det er gjort flere mislykkede forsøk på å oppgi koden.

Delt iPad-sikkerhet i iPadOS

Oversikt

Delt iPad er en flerbrukermodus for bruk i iPad-utrullinger. Det gjør det mulig for brukerne å dele en iPad samtidig som dokumenter og data holdes atskilt for hver bruker. Hver bruker får sin egen private, reserverte lagringsplass, som implementeres som et APFS-volum (Apple File System) beskyttet med brukerens akkreditiver. Delt iPad krever at det brukes en administrert Apple-ID som utstedes og eies av organisasjonen, og gjør det mulig for en bruker å logge på en hvilken som helst organisasjonseid enhet som er konfigurert til bruk av flere brukere. Brukerdataene deles opp i adskilte kataloger, alle i sine egne databeskyttelsesdomener og beskyttet ved hjelp av både UNIX-tillatelser og sandkasseteknologi. I iPadOS 13.4 eller nyere kan brukere også logge på med en midlertidig økt. Når brukeren logger av en midlertidig økt, slettes APFS-volumet deres og det reserverte området returneres til systemet.

Logge på Delt iPad

Både innebygde og forente administrerte Apple-ID-er støttes når man logger på Delt iPad. Når en forent konto brukes for første gang, sendes brukeren til ID-leverandørens (IdP) påloggingsportal. Hvis påloggingen er vellykket, gis det et tilgangskjennetegn med kort levetid for de administrerte Apple-ID-ene, og påloggingsprosessen fortsetter på liknende måte som den integrerte påloggingsprosessen for administrerte Apple-ID-er. Når påloggingen er fullført, vil oppsettassistenten for Delt iPad be brukeren om å velge en kode (akkreditiver) for å sikre de lokale dataene på enheten og for autentisering i påloggingsvinduet i fremtiden. Som på en enhet med én bruker der brukeren må logge seg

på med administrert Apple-ID ved hjelp av forent konto og deretter låse opp enheten med koden, logger brukeren seg på én gang på Delt iPad ved hjelp av forent konto, og deretter brukes brukerens kode.

Når en bruker logger seg på uten forent autentisering, autentiseres den administrerte Apple-ID-en med Apple Identity Service (IDS) ved hjelp av SRP-protokollen. Hvis autentisering lykkes, gis det et enhetsspesifikt tilgangskjennetegn med kort levetid. Hvis brukeren har brukt enheten tidligere, har han eller hun allerede en lokal brukerkonto som låses opp med de samme akkreditivene.

Hvis brukeren ikke har brukt enheten før eller bruker en midlertidig økt, klargjør Delt iPad en ny UNIX-bruker-ID, et APFS-volum til å lagre brukerens personlige data og en lokal nøkkelring. Siden lagring tilordnes (reserveres) for brukeren når APFS-volumet opprettes, kan det hende at det ikke er nok plass til å opprette et nytt volum. I et slikt tilfelle vil systemet identifisere en eksisterende bruker med data som er ferdig synkronisert til skyen, og kaste den brukeren ut av enheten, slik at den nye brukeren kan logge på. I det usannsynlige tilfellet at alle eksisterende brukere ikke har fullført opplastingen av skydataene, mislykkes påloggingen av den nye brukeren. For å logge på må den nye brukeren vente til en brukers data er ferdig med å synkronisere eller få administratoren til å slette en eksisterende brukerkonto med tvang og dermed risikere datatap.

Hvis enheten ikke er koblet til internett (for eksempel hvis brukeren ikke har et Wi-Fi-tilgangspunkt), kan autentisering skje mot den lokale kontoen i et begrenset antall dager. I slike situasjoner kan kun brukere med eksisterende lokale kontoer eller en midlertidig økt logge på. Etter at tidsbegrensningen utløper, må brukere autentisere på nettet selv om en lokal konto allerede eksisterer.

Når en brukers lokale konto har blitt låst opp eller opprettet, hvis den er autentisert via fjerntilgang, omgjøres kjennetegnet med kort levetid som utstedes av Apples tjenere, til et iCloud-kjennetegn som tillater innlogging på iCloud. Deretter gjenopprettes brukerens innstillinger og dokumentene og dataene synkroniseres fra iCloud.

Når en brukerøkt er aktiv og enheten er tilkoblet internett, lagres dokumenter og data på iCloud etter hvert som de opprettes eller endres. Dessuten er det en synkroniseringsmekanisme i bakgrunnen som bidrar til å sikre at endringer sendes til iCloud eller andre nettjenester ved hjelp av NSURLSession-bakgrunnsøkter etter at brukeren har logget av. Etter at bakgrunnssynkronisering for denne brukeren er fullført, deaktiveres brukerens APFS-volum og kan ikke aktiveres uten at brukeren logger på igjen.

Midlertidige økter synkroniserer ikke data med iCloud, og selv om en midlertidig økt kan logge på en synkroniseringstjeneste fra tredjepart som Box eller Google Drive, er det ikke mulig å fortsette synkronisering av data når den midlertidige økten avsluttes.

Logge av Delt iPad

Når en bruker logger av Delt iPad, låses brukerens nøkkeletui umiddelbart og alle apper lukkes. For å fremskynde pålogging av en ny bruker utsetter iPadOS noen vanlige avloggingshandlinger midlertidig og presenterer et påloggingsvindu til den nye brukeren. Hvis en bruker logger på i denne perioden (cirka 30 sekunder), utfører Delt iPad den utsatte oppryddingen som en del av påloggingen til den nye brukerkontoen. Hvis Delt iPad imidlertid ikke brukes, utløses den utsatte oppryddingen. Under opprydningsfasen startes påloggingsvinduet på nytt som om en annen utlogging hadde funnet sted.

Når en midlertidig økt avsluttes, utfører Delt iPad den fullstendige avloggingssekvensen og sletter APFS-volumet til den midlertidige økten umiddelbart.

Apple Configurator 2-sikkerhet

Apple Configurator 2 har en fleksibel, sikker og enhetsfokusert design som lar administratoren raskt og enkelt konfigurere fra én til flere titalls iOS-, iPadOS- og tvOS-enheter ved å koble dem til en Mac via USB (eller tvOS-enheter sammenkoblet via Bonjour) før de gis til brukere. Med Apple Configurator 2 kan administratoren oppdatere programvare, installere apper og konfigurasjonsprofiler, endre navn på og bytte bakgrunnsbilde på enheter, eksportere enhetsinformasjon og dokumenter og mye mer.

Administratorer kan også velge å inkludere iOS-, iPadOS- og tvOS-enheter i Apple School Manager eller Apple Business Manager ved hjelp av Apple Configurator 2, selv om enhetene ikke ble kjøpt direkte fra Apple, en Apple-autorisert forhandler eller en autorisert operatør. Når administratoren konfigurerer en enhet som er registrert manuelt, oppfører den seg på akkurat samme måte som andre registrerte enheter, med obligatorisk tilsyn og MDM-registrering. Brukere av enheter som ikke ble kjøpt direkte, har en periode på 30 dager der enheten kan fjernes fra registrering, tilsyn og MDM. Den midlertidige perioden på 30 dager starter når enheten aktiveres.

Hvis iOS-, iPadOS- og tvOS-enheter har absolutt ingen internettilkobling og kobles til en Mac-vert med en internettilkobling mens enhetene konfigureres, kan organisasjoner aktivere dem ved hjelp av Apple Configurator 2. Administratorer kan gjenopprette, aktivere og klargjøre enheter med nødvendig konfigurasjon, inkludert apper, profiler og dokumenter uten å måtte koble til Wi-Fi-nettverk eller mobilnettverk. Denne funksjonen tillater ikke en administrator å overstyre noen av de eksisterende Aktiveringslås-kravene som vanligvis er påkrevd under aktivering uten tilkobling.

Skjermtid-sikkerhet

Skjermtid er en funksjon i iOS 12 eller nyere, iPadOS og macOS 10.15 eller nyere og i enkelte funksjoner i watchOS 6 eller nyere som gir brukerne oversikt og kontroll over deres egen app- og nettbruk, eller deres barns app- og nettbruk. Selv om Skjermtid ikke er en ny funksjon for systemsikkerhet, er det viktig å forstå hvordan Skjermtid beskytter sikkerheten og personvernet til dataene som samles inn og deles mellom enheter.

Det finnes to typer brukere i Skjermtid: voksne og barn.

Tabellen under beskriver hovedfunksjonene for Skjermtid.

Funksjon	Støttet operativsystem
Se bruksdata	iOS iPadOS macOS
Håndhev ytterligere restriksjoner	iOS iPadOS macOS watchOS
Angi nettbruksgrenser	iOS iPadOS macOS

Funksjon	Støttet operativsystem
Angi appgrenser	iOS iPadOS macOS watchOS
Konfigurere skjermfri tid	iOS iPadOS macOS watchOS

For brukere som administrerer egen enhetsbruk, kan Skjermtid-kontroller og bruksdata synkroniseres på tvers av enheter knyttet til samme iCloud-konto ved hjelp av gjennomgående CloudKit-kryptering. Dette krever at brukerens konto har aktivert tofaktorautentisering (synkronisering er på som standard). Skjermtid erstatter Restriksjoner-funksjonen fra tidligere versjoner av iOS og iPadOS og Foreldrekontroll-funksjonen fra tidligere versjoner av macOS.

I iOS 13 eller nyere, iPadOS 13.1 eller nyere og macOS 10.15 eller nyere vil Skjermtid-brukere og barn som administreres, automatisk dele bruken på tvers av enheter hvis iCloud-kontoen har aktivert tofaktorautentisering. Når en bruker tømmer Safari-loggen eller sletter en app, fjernes tilhørende bruksdata fra enheten og alle synkroniserte enheter.

Foreldre og Skjermtid

Foreldre kan også bruke Skjermtid på iOS-, iPadOS- og macOS-enheter til å forstå og styre barnas bruk. Hvis forelderen er en familieorganisator (i iCloud-familiedeling), kan de se bruksdata og administrere Skjermtid-innstillinger for barna. Barn informeres når foreldrene slår på Skjermtid, og de kan også følge med på sin egen bruk. Når foreldre slår på Skjermtid for barna, angir foreldrene en kode, slik at barna ikke kan utføre endringer. Når de når myndighetsalder (alderen varierer avhengig av land eller område), kan barna slå av denne overvåkningen.

Bruksdata og konfigurasjonsinnstillinger overføres mellom forelderens og barnets enheter ved hjelp av en forbindelse med gjennomgående kryptert IDS-protokoll (Apple Identity Service). Krypterte data kan oppbevares i en begrenset periode på IDS-tjenere til de leses av mottakerenheten (for eksempel så snart iPhone, iPad eller iPod touch slås på, hvis enheten var slått av). Disse dataene kan ikke leses av Apple.

Skjermtid-analyse

Hvis brukeren slår på Del iPhone- og Watch-analyse, samles kun følgende anonymiserte data inn, slik at Apple får en bedre forståelse av hvordan Skjermtid brukes:

- ble Skjermtid slått på under oppsettassistent eller senere i Innstillinger
- endringer i kategoribruk etter at det er definert en grense for den (innen 90 dager)
- er Skjermtid slått på
- er Skjermfri tid aktivert
- antall ganger «Be om mer» ble brukt

- antall appgrenser
- antall ganger brukere viste bruken i innstillinger for Skjermtid, per brukertype og per visningstype (lokal, ekstern, widget)
- antall ganger brukere ignorerer en grense, per brukertype
- antall ganger brukere sletter en grense, per brukertype

Ingen spesifikke app- eller nettbruksdata samles inn av Apple. Når en bruker ser en liste over apper i Skjermtid-bruksinformasjon, hentes appsymbolene direkte fra App Store, som ikke beholder noen data fra disse forespørslene.

Ordliste

Address Space Layout Randomization (ASLR) En teknikk som brukes av operativsystemer for å gjøre det mye vanskeligere å misbruke en programvarefeil. Den sørger for at adresser og plasseringer i minnet er uforutsigbare, slik at misbrukskode ikke kan hardkode disse verdiene.

AES (Advanced Encryption Standard (avansert krypteringsstandard)) En populær global krypteringsstandard som brukes til å kryptere data for å beskytte dem.

AES kryptografimotor En dedikert maskinvarekomponent som implementerer AES.

AES-XTS En modus av AES definert i IEEE 1619-2007 som er ment å fungere til å kryptere lagringsmedier.

APFS (Apple File System) Standard filsystem for iOS, iPadOS, tvOS, watchOS og Macer med macOS 10.13 eller nyere. APFS har sterk kryptering, plassdeling, øyeblikksbilder, rask endring av katalogstørrelser og forbedrede grunnleggende filsystemfunksjoner.

Apple Business Manager Apple Business Manager er en enkel, nettbasert portal for IT-administratorer som gjør det raskt og effektivt å rulle ut Apple-enheter som organisasjonen har kjøpt direkte fra Apple eller fra en deltakende Apple-autorisert forhandler eller operatør. De kan rulle ut enheter automatisk i MDM-løsningen uten å måtte fysisk berøre eller klargjøre enhetene før brukerne får dem.

Apple Identity Service (IDS) Apples katalog for offentlige iMessage-nøkler, APNs-adresser og telefonnumre og e-postadresser som brukes til å slå opp nøklene og enhetsadressene.

Apple School Manager Apple School Manager er en enkel, nettbasert portal for IT-administratorer som gjør det raskt og effektivt å rulle ut Apple-enheter som organisasjonen har kjøpt direkte fra Apple eller fra en deltakende Apple-autorisert forhandler eller operatør. De kan rulle ut enheter automatisk i MDM-løsningen uten å måtte fysisk berøre eller klargjøre enhetene før brukerne får dem.

Apples pushvarslingstjeneste (APNs) En verdensomspennende tjeneste fra Apple som leverer pushvarslinger til Apple-enheter.

Boot Camp Boot Camp støtter installasjon av Microsoft Windows på støttede Macer.

Boot Progress Register (BPR) Et sett med System on Chip-maskinvareflagg (SoC-maskinvareflagg) som programvare kan bruke til å spore oppstartsmodusene enheten er i, for eksempel Device Firmware Update-modus (DFU-modus) og gjenopprettingsmodus. Når et Boot Progress Register-flagg er satt, kan det ikke fjernes. Dette gjør det mulig for senere programvare å få en pålitelig indikator om systemets tilstand.

CKRecord En ordbok med nøkkelverdipar som inneholder data arkivert til eller hentet fra CloudKit.

Databeskyttelse Mekanismer for fil- og nøkkelkjedebeskyttelse for støttede Apple-enheter. Begrepet kan også brukes om API-er som benyttes av apper til å beskytte filer og nøkkelringobjekter.

Datahvelv En mekanisme, som håndheves av kjernen, for å beskytte mot uautorisert tilgang til data uavhengig av om appen som sender forespørselen, selv bruker sandkasseteknologi.

DFU-modus (Device Firmware Upgrade) En modus der oppstart-ROM-koden til en enhet venter på å bli gjenopprettet via USB. Skjermen er svart når enheten er i DFU-modus, men når den kobles til en datamaskin som har iTunes eller Finder, vises denne teksten: «iTunes (eller Finder) har oppdaget en (iPad, iPhone eller iPod touch) i gjenopprettingsmodus. Brukeren må gjenopprette denne (iPad, iPhone eller iPod touch) før den kan brukes sammen med iTunes (eller Finder).»

Direkte minnetilgang (DMA) En funksjon som gir maskinvareundersystemer tilgang til hovedminnet uavhengig av prosessoren.

Effaceable Storage Et eget område med NAND-lagringsplass, som brukes til å lagre krypteringsnøkler, og som man kan få direkte tilgang til og som kan slettes på en sikker måte. Det gir riktignok ikke beskyttelse hvis en utenforstående har fått tak i den fysiske enheten, men nøkler som oppbevares i Effaceable Storage kan brukes som en del av et nøkkelhierarki for å legge til rette for rask sletting og «forward security».

Elliptic Curve Diffie-Hellman Exchange (ECDHE) Elliptic Curve Diffie-Hellman Exchange med kortvarige nøkler. ECDHE gjør det mulig for to parter å enes om en hemmelig nøkkel på en måte som hindrer at nøkkelen oppdages av en tyvtitter som ser meldingene mellom de to partene.

Elliptic Curve Digital Signature Algorithm (ECDSA) Elliptic Curve Digital Signature Algorithm (ECDSA) er en digitalsignaturalgoritme basert på elliptisk kurve-kryptografi.

eSPI Enhanced Serial Peripheral Interface-databuss for synkron seriekommunikasjon.

Exclusive Chip Identification (ECID) En 64-bit-ID som er unik for prosessoren i iOS- og iPadOS-enheter. Hvis en samtale besvares på én enhet, avsluttes ringingen på iCloud-sammenkoblede enheter i nærheten med en kort beskjed via Bluetooth Low Energy (BLE) 4.0. Annonseringsbytene krypteres ved hjelp av samme metode som Handoff-annonseringer. Den brukes som en del av tilpasningsprosessen og anses ikke for å være hemmelig.

filsystemnøkkel Nøkkelen som krypterer metadataene til de enkelte filene, inkludert klassenøkkelen. Den oppbevares i Effaceable Storage for å legge til rette for rask sletting i større grad enn for konfidensialitet.

Gjenopprettingsmodus En modus brukes til å gjenopprette mange Apple-enheter hvis den ikke gjenkjenner brukerens enhet, slik at brukeren kan installere operativsystemet igjen.

godkjenning av systemprogramvare En prosess som kombinerer kryptografiske nøkler innebygd i maskinvaren med en nettjeneste for å sikre at kun godkjent programvare fra Apple, som er riktig for støttede enheter, leveres og installeres på oppgraderingstidspunktet.

gruppe-ID (GID) Samme som UID, men felles for alle prosessorer i klassen.

HMAC En hash-basert meldingsautentiseringskode basert på en kryptografisk hash-funksjon.

iBoot Kode som laster XNU, som en del av den sikre oppstartssekvensen. Avhengig av System on Chip-generasjonen (SoC-generasjonen), kan iBoot lastes av Low-Level Bootloader eller direkte av oppstart-ROM.

Input/Output Memory Management Unit (IOMMU) En minneadministreringsenhet for inndata og utdata. Et undersystem i en integrert brikke som kontrollerer tilgangen til adresseområdet fra andre inn-/utdataenheter og eksterne enheter.

integrering Prosessen der en brukers kode gjøres om til en krypteringsnøkkel og styrkes med enhetens UID. Denne prosessen bidrar til å sikre at et brute-force-angrep må utføres på en gitt enhet, noe som gjør at omfanget begrenses, og at angrepet ikke kan utføres parallelt. Integreringsalgoritmen er PBKDF2, og den bruker AES-nøkkel sammen med enhets-UID-en som PRF (pseudorandom function) for hver iterasjon.

integrert krets (IC) Kalles også for *mikrobrikke*.

Joint Test Action Group (JTAG) Standardverktøy for feilsøking av maskinvare som brukes av programmerere og brikkeutviklere.

klargjøringsprofil En egenskapslistefil (.plist-fil) signert av Apple som inneholder et sett med entiteter og rettigheter som gjør det mulig å installere og teste apper på en iOS-enhet. En klargjøringsprofil for utvikling inneholder en liste over enhetene som en utvikler har valgt for ad hoc-distribusjon, og en klargjøringsprofil for distribusjon inneholder app-ID-en til en bedriftsutviklet app.

Kodeavledet nøkkel (PDK) Krypteringsnøkkelen som er avledet fra integreringen av brukerens passord med den langvarige SKP-nøkkelen og UID-en til Secure Enclave.

Komponent for sikker lagring Brikken er designet med uforanderlig RO-kode, en maskinvarebasert generator for tilfeldige tall, kryptografimotorer og gjenkjenning av fysisk manipulasjon. I støttede enheter kobles Secure Enclave sammen med en komponent for sikker lagring for lagring av anti-repetisjonsnonce-verdien. For å lese og oppdatere nonce-verdier bruker Secure Enclave og brikken for sikker lagring en sikker protokoll som bidrar til å sørge for eksklusiv tilgang til nonce-verdiene. Det finnes flere generasjoner av denne teknologien, alle med forskjellig sikkerhetsnivå.

Low-Level Bootloader (LLB) På Macer med en to-trinns oppstartsarkitektur, inneholder LLB koden som aktiveres av oppstart-ROM, som så laster iBoot, som en del av den sikre oppstartssekvensen.

maskinvaresikkerhetsmodul (Hardware security module, HSM) En manipulasjonsbestandig spesialmaskin som verner og administrerer digitale nøkler.

MDM (Mobile Device Management) En tjeneste som lar en administrator fjernadministrere registrerte enheter. Når en enhet er registrert, kan brukeren bruke MDM-tjenesten over nettverket til å konfigurere innstilling og utføre andre oppgaver på enheten uten brukermedvirkning.

medienøkkel En del av hierarkiet av krypteringsnøkler som bidrar til å gi sikker og umiddelbar sletting. I iOS, iPadOS, tvOS og watchOS pakker medienøkkelen metadataene på datavolumet (og uten dem blir tilgang til alle filspesifikke nøkler umulig, og alle filer som er beskyttet med Databeskyttelse, blir dermed utilgjengelige). I macOS pakker medienøkkelen nøkkelmaterialet, alle metadata og data på volumet som er beskyttet med FileVault. I begge tilfeller blir krypterte data utilgjengelige når medienøkkelen slettes.

minnekontroller Undersystemet i System on Chip (SoC) som kontrollerer grensesnittet mellom SoC og hovedminnet.

NAND Ikke-flyktig flashminne.

nøkkeletui En datastruktur som brukes til å lagre en samling klassenøkler. De ulike typene (bruker, enhet, system, sikkerhetskopiering, deponering og iCloud-sikkerhetskopiering) har samme format.

En topptekst som inneholder: Versjon (satt til fire i iOS 12 eller nyere), type (system, sikkerhetskopiering, deponering eller iCloud-sikkerhetskopiering), Nøkkeletui-UUID, en HMAC hvis nøkkeletuiet er signert og metoden som brukes for å pakke inn klassenøklene: integrering i UID eller PBKDF2, sammen med salt- og iterasjonsnummer.

En liste over klassenøkler: Nøkkel-UUID; Klasse (hvilken databeskyttelsesklasse for fil eller nøkkelring), innpakningstype (kun UID-avledet nøkkel; UID-avledet nøkkel og kodeavledet nøkkel), innpakket klassenøkkel og en offentlig nøkkel for asymmetriske klasser.

nøkkelinnpakning Kryptering av én nøkkel med en annen nøkkel. iOS og iPadOS bruker NIST AES-nøkkelinnpakning i henhold til RFC 3394.

nøkkelring Infrastrukturen og et API-sett som brukes av operativsystemer fra Apple og tredjepartsapper til å lagre og gjenfinne passord, nøkler og andre sensitive akkreditiver.

Oppstart-ROM Den første koden som kjøres av enhetsprosessoren når enheten startes opp for første gang. Den er en integrert del av prosessoren og kan ikke endres verken av Apple eller av utenforstående.

opptegning av mønsterlinjer En matematisk framstilling av retningen og bredden på linjene som trekkes ut fra en del av et fingeravtrykk.

per-fil-nøkkel Nøkkelen som brukes av databeskyttelse til å kryptere en fil på filsystemet. Per-fil-nøkkelen pakkes med en klassenøkkel og lagres i filens metadata.

programvare-seed-biter Dedikerte bits i Secure Enclave AES-motoren som legges til UID når nøkler genereres fra UID. Hver programvare-seed-bit har en korresponderende lås-bit. Secure Enclave oppstart-ROM og operativsystem kan uavhengig endre verdien av hver programvare-seed-bit så lenge den tilsvarende lås-biten ikke er satt. Når lås-biten er satt, kan verken programvare-seed-biten eller lås-biten endres. Programvare-seed-biter og deres låser nullstilles når Secure Enclave-starter på nytt.

Sealed Key Protection (SKP) En teknologi i Databeskyttelse som beskytter, eller *forsegler*, krypteringsnøkler med målinger av programvaren på systemet og nøklene som kun er tilgjengelige i maskinvaren (for eksempel UID-en i Secure Enclave).

sepOS Secure Enclave-firmwaren, basert på en Apple-tilpasset versjon av L4-mikrokjernen.

Sikkerhetsbelønning fra Apple En belønning gitt av Apple til personer som rapporterer en sårbarhet som påvirker de nyeste operativsystemene og, der det er relevant, den nyeste maskinvaren.

SSD-kontroller Et maskinvareundersystem som administrerer lagringsmediene (solid-state drive).

System Coprocessor Integrity Protection (SCIP) En mekanisme som Apple bruker, som er utviklet for å hindre modifisering av koprosessorfirmware.

System on Chip (SoC) En integrert krets (IC) der flere komponenter kombineres på én brikke. Applikasjonsprosessoren, Secure Enclave og andre koprosessorer er komponenter i SoC-en.

UEFI-firmware Unified Extensible Firmware Interface, en erstatningsteknologi for BIOS for å koble firmware en datamaskins operativsystem.

Uniform Resource Identifier (URI) En tegnrekke som identifiserer en nettbasert ressurs.

unik ID (UID) En 256-bit AES-nøkkel som brennes inn i prosessoren under produksjonen. Den kan ikke leses av firmware eller programvare og brukes kun av prosessorens maskinvarebaserte AES-motor. For å få tak i den faktiske nøkkelen, må en utenforstående utføre et svært avansert og kostbart fysisk angrep mot prosessoren. UID-en er ikke knyttet til noen annen ID på enheten, inkludert, men ikke begrenset til UDID-en.

xART eXtended Anti-Replay Technology, eller et sett med tjenester som sørger for kryptert, autentisert og varig lagring for Secure Enclave med anti-repetisjonsegenskaper basert på den fysiske lagringsarkitekturen. Se komponent for sikker lagring.

XNU Kjernen som er hjertet i operativsystemene til Apple. Det antas at den er godkjent, og den iverksetter sikkerhetstiltak som for eksempel kodesignering, sandkasseteknologi, rettighetskontroll og Address Space Layout Randomization (ASLR).

Endringslogg for dokumentet

Dato	Oppsummering
Mai 2021	Oppdatert for: • iOS 14.5 • iPadOS 14.5 • macOS 11.3 • tvOS 14.5 • watchOS 7.4 Emner som er lagt til: • Magic Keyboard med Touch ID. • Sikker intensjon og tilkoblinger til Secure Enclave. • Automatisk opplåsing og Apple Watch. • CustomOS Image4-listesignatur (coih). Emner som er redigert: • Lagt til to nye Ekspressmodustransaksjoner i Ekspress-kort med reservestrøm. • Redigert Oppsummering av Secure Enclave-funksjonen. • Innhold om programvareoppdatering lagt til i Secure Multi-Boot (smb3). • Ekstra innhold i Sealed Key Protection (SKP).

Dato	Oppsummering
Februar 2021	Oppdatert for: • iOS 14.3 • iPadOS 14.3 • macOS 11.1 • tvOS 14.3 • watchOS 7.2 Emner som er lagt til: • Minnesikker iBoot-implementering • Oppstartsprosess for Macer med Apple Silicon • Oppstartsmoduser for Macer med Apple Silicon • Kontroll av sikkerhetsregelsett for Startdisk for Macer med Apple Silicon • Oppretting og administrering av LocalPolicy-signer-ingsnøkkel • Innhold i en LocalPolicy-fil for en Mac med Apple Silicon • Sikkerhet for signert systemvolum i macOS • Apple SRD-enhet (Security Research Device) • Passordovervåking • IPv6-sikkerhet • Bilnøkler-sikkerhet i iOS Emner som er oppdatert: • Secure Enclave • Maskinvarefrakobling av mikrofoner • recoveryOS og diagnostikkmiljøer for Intel-baserte Macer • Beskyttelser for direkte minnetilgang for Macer • Kjerneutvidelser i macOS • System Integrity Protection • Systemsikkerhet for watchOS • Administrering av FileVault i macOS • Apptilgang til arkiverte passord • Anbefalinger for passordsikkerhet • Apple Cash-sikkerhet i iOS, iPadOS og watchOS • Sikker bruk av Spør bedriften med Meldinger-appen • Wi-Fi-personvern • Aktiveringslås-sikkerhet • Apple Configurator 2-sikkerhet
April 2020	Oppdatert for: • iOS 13.4 • iPadOS 13.4 • macOS 10.15.4 • tvOS 13.4 • watchOS 6.2 Oppdateringer: • Frakobling av mikrofonen på iPad lagt til i Maskinvarefra-kobling av mikrofoner. • Datahvelv lagt til i Beskytte apptilgang til brukerdata. • Oppdateringer av Administrering av FileVault i macOS og Kommandolinjeverktøy. • Tillegg om Verktøy for fjerning av skadelig programvare i Beskyttelse mot skadelig programvare i macOS. • Oppdateringer av Delt iPad-sikkerhet i iPadOS.

Dato	Oppsummering
Desember 2019	Har kombinert Sikkerhetshåndbok for iOS, macOS Security Overview og Apple T2 Security Chip Overview Oppdatert for: • iOS 13.3 • iPadOS 13.3 • macOS 10.15.2 • tvOS 13.3 • watchOS 6.1.1 Kontroller for personvern, Siri og Siri-forslag samt Intelligent sporingsbeskyttelse i Safari er fjernet. Se https://www.apple.com/no/privacy/ for oppdatert informasjon om disse funksjonene.
Mai 2019	Oppdatert for iOS 12.3 • Støtte for TLS 1.3 • Revidert beskrivelse av AirDrop-sikkerhet • DFU-modus og gjenopprettingsmodus • Kodekrav for tilbehørstilkoblinger
November 2018	Oppdatert for iOS 12.1 • Gruppesamtaler i FaceTime
September 2018	Oppdatert for iOS 12 Secure Enclave • Beskyttelse av operativsystemintegritet • Ekspresskort med reservestrøm • DFU-modus og gjenopprettingsmodus • HomeKit TV-fjernkontrolltilbehør • Kontaktløse kort • Studentbevis • Siri-forslag • Snarveier i Siri • Snarveier-app • Administrering av brukerpassord • Skjermtid • Sikkerhetssertifiseringer og -programmer
Juli 2018	Oppdatert for iOS 11.4 • Biometriregler • HomeKit • Apple Pay • Spør bedriften • Meldinger i iCloud • Apple Business Manager
Desember 2017	Oppdatert for iOS 11.2 • Apple Pay Cash

Dato	Oppsummering
Oktober 2017	Oppdatert for iOS 11.1 • Sikkerhetssertifiseringer og -programmer • Touch ID / Face ID • Delte notater • Gjennomgående kryptering i CloudKit • TLS-oppdatering • Apple Pay, betaling med Apple Pay på nettet • Siri-forslag • Delt iPad
Juli 2017	Oppdatert for iOS 10.3 • Secure Enclave • Beskyttelse av fildata • Nøkkeletuier • Sikkerhetssertifiseringer og -programmer • SiriKit • HealthKit • Nettverkssikkerhet • Bluetooth • Delt iPad • Mistet-modus • Aktiveringslås • Kontroller for personvern
Mars 2017	Oppdatert for iOS 10 Systemsikkerhet • Databeskyttelsesklasser • Sikkerhetssertifiseringer og -programmer • HomeKit, ReplayKit, SiriKit • Apple Watch • Wi-Fi, VPN • Single Sign On • Apple Pay, betaling med Apple Pay på nettet • Klargjøring av kredittkort, debetkort og forhånds-betalte kort • Safari-forslag
Mai 2016	Oppdatert for iOS 9.3 • Administrert Apple-ID • Tofaktorautentisering for Apple-ID • Nøkkeletuier • Sikkerhetssertifiseringer • Mistet-modus, Aktiveringslås • Sikre notater • Apple School Manager • Delt iPad

Dato	Oppsummering
September 2015	Oppdatert for iOS 9 Aktiveringslås for Apple Watch • Koderegler • API-støtte for Touch ID • Databeskyttelse på A8 bruker AES-XTS • Nøkkeletuier for programvareoppdateringer i bakgrunnen • Sertifiseringsoppdateringer • Godkjenningsmodell for bedriftsapp • Databeskyttelse for Safari-bokmerker • App Transport Security • VPN-spesifikasjoner • iCloud-fjerntilgang for HomeKit • Lojalitetskort i Apple Pay, kortutstederapp for Apple Pay • Spotlight-indeksering på enheten • Sammenkoblingsmodellen i iOS • Apple Configurator 2 • Restriksjoner

Apple
One Apple Park Way
Cupertino, CA 95014
USA
apple.com

H028-00406