# Apple och säkerhetsteknik

Maj 2024

# Innehåll

# Introduktion till Apple och säkerhetsteknik

Apple skapar sina plattformar med säkerheten i fokus. Genom att utgå från erfarenheten av att skapa världens mest avancerade mobiloperativsystem har Apple skapat säkerhetsarkitekturer som hanterar de unika kraven för mobil, klocka, dator och hem.

Varje Apple-enhet är uppbyggd av *maskinvara*, *programvara* och *tjänster* som har utformats så att de fungerar tillsammans med maximal säkerhet och en transparent användarupplevelse för att uppnå målet som är att personlig information ska förbli privat. Exempelvis driver Apple-utformade kretsar och säkerhetsmaskinvaror kritiska säkerhetsfunktioner. Programvaruskydd används till att skydda operativsystemet och tredjepartsappar. Slutligen har tjänster en mekanism för säkra och snabba programuppdateringar, driver ett säkrare appekosystem och skyddar kommunikation och betalning. Resultatet blir att Apple-enheter skyddar inte bara enheten och de data som lagras på den, utan hela ekosystemet inklusive allt som användarna gör lokalt, i nätverk och med viktiga internettjänster.

Precis som vi utformar våra produkter för att vara enkla, intuitiva och praktiska så utformar vi dem för att vara säkra. Viktiga säkerhetsfunktioner, som maskinvarubaserad enhetskryptering, kan inte avaktiveras av misstag. Andra funktioner, som Face ID och Touch ID, ger en bättre användarupplevelse eftersom det blir enklare och mer intuitivt att skydda enheten. Och eftersom många av funktionerna är aktiverade som förval behöver varken användare eller IT-avdelningar utföra några omfattande konfigurationer.

I den här dokumentationen finns information om hur säkerhetsteknik och säkerhetsfunktioner används i Apple-plattformarna. Den hjälper också organisationer att kombinera säkerhetstekniken i Apple-plattformen med egna policyer och rutiner som uppfyller deras säkerhetsbehov.

Innehållet är uppdelat i följande ämnesområden:

- **Maskinvarusäkerhet och biometri:** De kretsar och den maskinvara som utgör grunden för säkerheten i Apple-enheter, inklusive Apple Silicon, Secure Enclave, kryptografiska motorer, Face ID och Touch ID.
- **Systemsäkerhet:** De integrerade maskin- och programvarufunktionerna som står för säker start, uppdateringar och aktiv drift av Apple-operativsystem.
- **Kryptering och dataskydd:** Den arkitektur och design som skyddar användarnas data om enheten tappas bort eller blir stulen eller om någon obehörig person eller process försöker att använda eller ändra den.
- **Appsäkerhet:** Den programvara och de tjänster som tillhandahåller ett säkert appekosystem som gör det möjligt att köra appar säkert och utan att riskera plattformens integritet.

- **Tjänstesäkerhet:** Apples tjänster för identifiering, lösenordshantering, betalningar, kommunikation och för att hitta förlorade enheter.
- **Nätverkssäkerhet:** Nätverksprotokoll enligt branschstandarder som ger säker autentisering och kryptering av data vid överföring.
- **Säkerhet och kit för utvecklare:** ”Ramverkskit” för säker och integritetsskyddad hantering av hem och hälsa, liksom en utökning av Apples funktioner för enheter och tjänster till tredjepartsappar.
- **Säker enhetshantering:** Metoder som gör det möjligt att hantera Apple-enheter, hjälper till att förhindra obehörig användning och aktivera fjärradering om en enhet försvinner eller blir stulen.

# Vårt engagemang för säkerhet

Apple arbetar för sina kunders säkerhet genom att använda ledande integritets- och säkerhetsteknik som har utformats för att skydda personlig information samt heltäckande lösningar för att skydda affärsdata i företagsmiljöer. Apple belönar forskare för det arbete de utför för att hitta sårbarheter genom programmet Apple Security Bounty. Information om programmet och olika belöningskategorier finns på https://security.apple.com/se/bounty/.

Vi har ett särskilt säkerhetsteam som deltar i arbetet med alla Apple-produkter. Teamet tillhandhåller testning och övervakning av produktsäkerheten, både under utveckling och efter lansering. Apple-teamet tillhandahåller också säkerhetsverktyg och utbildning samt söker aktivt efter hot och rapporter om nya säkerhetsproblem. Apple är medlem i FIRST – Forum of Incident Response and Security Teams.

Apple fortsätter att tänja på gränserna för vad som är möjligt inom säkerhet och Integritet. De använder anpassade Apple-kretsar för alla produkter, som Apple Watch, iPhone och iPad samt M-seriekretsarna i Mac-datorer, som inte bara driver effektiva beräkningar utan även säkerhet. Exempelvis utgör Apple-kretsar grunden för säker start, Face ID och Touch ID samt dataskydd. Dessutom bidrar säkerhetsfunktioner som drivs av Apple-kretsar, som kärnintegritetsskydd, pekarautentiseringskoder och snabba behörighetsbegränsningar, till att förhindra vanliga typer av obehörigt utnyttjande. På det sättet blir åverkan från skadlig kod som på något vis lyckas köras kraftigt minskad.

För att få ut det mesta av de omfattande inbyggda säkerhetsfunktionerna i våra plattformar uppmuntras organisationer att granska sina IT- och säkerhetspolicyer och se till att använda sig av alla de lager av säkerhet som dessa plattformar erbjuder.

Mer information om hur du rapporterar problem till Apple och prenumererar på säkerhetsmeddelanden finns i Rapportera en säkerhets- eller integritetsrisk.

**Apple anser att integritet är en grundläggande mänsklig rättighet och har många inbyggda inställningar och alternativ som gör att användarna kan bestämma hur och när appar använder deras information samt vilken information som används. Du kan läsa mer om Apples förhållningssätt till integritet, integritetsreglage på Apple-enheter och Apples integritetspolicy på https://www.apple.com/se/privacy.**

*Obs!* Om inget annat sägs täcker den här dokumentationen följande operativsystemversioner: iOS 17.3, iPadOS 17.3, macOS 14.3, tvOS 17.3 och watchOS 10.3.

# Maskinvarusäkerhet och biometri

## Maskinvarusäkerhet i översikt

För att programvara ska vara säker måste den finnas på maskinvara med inbyggd säkerhet. Därför har Apples enheter med iOS, iPadOS, macOS, tvOS och watchOS skapats med inbyggda säkerhetsfunktioner i kretsarna. Bland funktionerna finns en processor som strömförsörjer systemsäkerhetsfunktioner, och även ytterligare kretsar som är dedikerade till säkerhetsfunktioner. Säkerhetsfokuserad maskinvara följer principen att stöda begränsade och diskret definierade funktioner för att minska angreppsytan. Sådana komponenter innehåller ett startminne (Boot ROM) som bildar en maskinvarubaserad betrodd rot, dedikerade AES-motorer för effektiv och säker kryptering och avkryptering samt en Secure Enclave. *Secure Enclave* är en komponent i Apples SoC (System on Chip) som finns i alla nyare iPhone-, iPad-, Apple Watch-, Apple TV- och HomePod-enheter samt i alla Mac-datorer med Apple Silicon och de med Apples T2-säkerhetskrets. Själva Secure Enclave följer samma designprincip som SoC och innehåller ett eget diskret startminne och en egen AES-motor. Secure Enclave tillhandahåller också grunden för den säkra generering och lagring av nycklar som krävs för kryptering av data vid vila, och den skyddar och utvärderar biometriska data för Face ID och Touch ID.

Lagringskryptering måste vara snabb och effektiv. Samtidigt kan den inte exponera de data (eller *nyckelmaterial*) den använder till att upprätta kryptografiska nyckelrelationer. AES-maskinvarumotorn löser det här problemet genom att utföra snabb in-line-kryptering och -avkryptering *när filer skrivs till eller läses*. En speciell kanal från Secure Enclave tillhandahåller nödvändigt nyckelmaterial till AES-motorn utan att exponera informationen mot approcessorn (eller processorn) eller operativsystemet som helhet. Detta säkerställer att Apples dataskydds- och FileVault-tekniker skyddar användarnas filer utan att exponera långlivade krypteringsnycklar.

Apple har utformat säker start för att skydda de lägsta nivåerna av programvara från att manipuleras, och för att endast tillåta att betrodd operativsystemprogramvara från Apple körs vid start. Säker start börjar med statisk kod som kallas *Boot ROM*. Den skapas när Apples SoC tillverkas och är en *betrodd rot för maskinvaran*. På Mac-datorer med en T2-krets startar förtroendet för den säkra macOS-startsekvensen med själva T2. (Både T2-kretsen och Secure Enclave kör dessutom sina egna säkra startprocesser med egen säker Boot ROM – detta är en exakt motsvarighet till hur kretsarna i A-serien samt M1- och M2-familjerna startar säkert.)

Secure Enclave bearbetar också ansikts- och fingeravtrycksdata från Face ID- och Touch ID-sensorerna i Apple-enheter. Detta ger säker autentisering samtidigt som integriteten och säkerheten för användarens biometriska data upprätthålls. Det innebär också att användarna kan dra nytta av säkerheten som långa och mer komplexa lösenkoder och lösenord ger, oftast tillsammans med snabb och smidig autentisering för åtkomst eller vid köp.

# Säkerhet och Apples SoC:er

Apple-utformade kretsar utgör nu en gemensam arkitektur i alla Apple-produkter och driver Mac-datorer, iPhone, iPad, Apple TV och Apple Watch. I över ett decennium har Apples främsta designteam byggt och utvecklat Apples SoC:er (System on Chip) och arbetat med Apple-kretsar. Resultatet är en skalbar arkitektur som kan anpassas för alla enheter och som är branschledande när det gäller säkerhetsfunktioner. Den här gemensamma grunden för säkerhetsfunktioner är bara möjlig från ett företag som utformar sina egna kretsar för att fungera med sin egen programvara.

Apple-kretsar har utvecklats och tillverkats specifikt för att göra systemsäkerhetsfunktionerna som beskrivs nedan möjliga.

| Funktion | A10 | A11, S3 | A12, A13, A14<br>S4–S9 | A15, A16, A17 | M1, M2, M3 |
|---|---|---|---|---|---|
| Kärnintegritetsskydd | ✓ | ✓ | ✓ | ✓ | ✓ |
| Begränsningar för snabb behörighet | ✗ | ✓ | ✓ | ✓ | ✓ |
| Integritetsskydd för systemcoprocessor | ✗ | ✗ | ✓ | ✓ | ✓ |
| Pekarautentiseringskoder | ✗ | ✗ | ✓ | ✓ | ✓ |
| Page Protection Layer | ✗ | ✓ | ✓ | ✗ | ✗<br>Se Anmärkning 1 nedan. |
| Secure Page Table Monitor | ✗ | ✗ | ✗ | ✓<br>Se Anmärkning 2 nedan. | ✗ |

*Anmärkning 1:* PPL (Page Protection Layer) kräver att plattformen kör *endast* signerad och betrodd kod – detta är en säkerhetsmodell som inte gäller i macOS.

*Anmärkning 2:* Secure Page Table Monitor (SPTM) stöds i A15, A16 och A17 och ersätter Page Protection Layer på plattformar som stöds.

Apple-utformade kretsar gör även särskilt dataskyddsfunktionerna nedan möjliga.

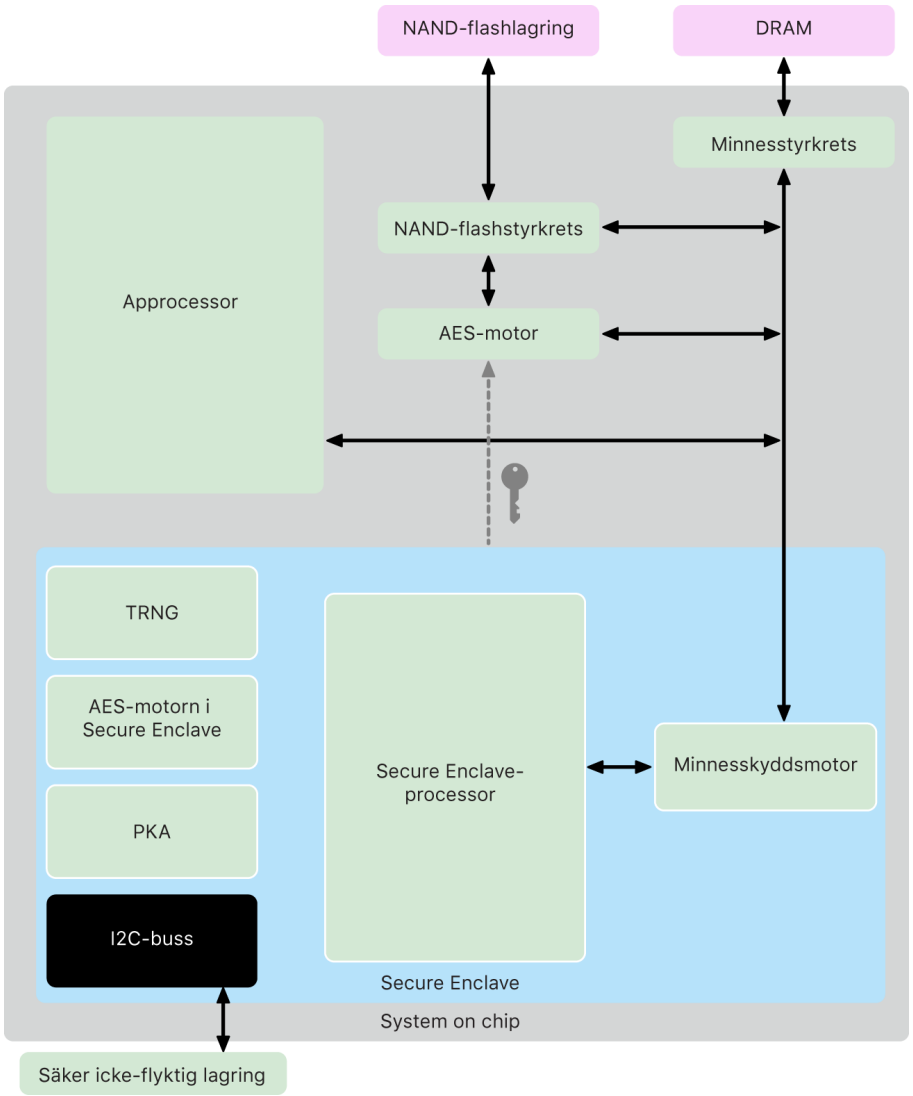
| Funktion | A10, A11<br>S3 | A12–A17<br>S4–S9<br>M1, M2, M3 |
|---|---|---|
| SKP (Sealed Key Protection) | ✓ | ✓ |
| recovery OS – Alla dataskyddsklasser skyddas | ✓ | ✓ |
| Alternativa starter för DFU-läge, diagnos och uppdateringar - Dataskydd för klass A, B och C | ✗ | ✓ |

# Secure Enclave

Secure Enclave är ett dedikerat, säkert undersystem i de senaste versionerna av iPhone, iPad, Mac, Apple TV, Apple Watch och HomePod.

## Översikt

Secure Enclave är ett dedikerat säkert delsystem som är integrerat i Apples SoC:er (Systems on Chip). Secure Enclave är isolerad från huvudprocessorn för att tillhandahålla ett extra säkerhetslager och har utformats för att skydda känsliga användardata även när approcessorns kärna har manipulerats. Den följer samma designprinciper som SoC gör – ett startminne för att upprätta en betrodd rot för maskinvaran, en AES-motor för effektiva och säkra krypteringsåtgärder samt skyddat minne. Även om Secure Enclave inte inkluderar lagring har den en mekanism för att lagra information säkert i ett kopplat lagringsutrymme som är separerat från NAND-flashminnet som används av approcessorn och operativsystemet.



Secure Enclave är en maskinvarufunktion i de flesta versionerna av iPhone, iPad, Mac, Apple TV, Apple Watch och HomePod, nämligen:

- iPhone 5s eller senare
- iPad Air eller senare
- Mac-datorer med Apple Silicon
- MacBook Pro-datorer med Touch Bar (2016 och 2017) som har Apple T1-krets
- Intel-baserade Mac-datorer med Apple T2-säkerhetskrets
- Apple TV HD eller senare
- Apple Watch Series 1 eller senare
- HomePod och HomePod mini

## Secure Enclave-processor

Secure Enclave-processorn står för den huvudsakliga beräkningskraften i Secure Enclave. För starkast möjliga isolering är Secure Enclave-processorn dedikerad för användning endast av Secure Enclave. Detta gör det svårare att genomföra sidokanalattacker som är beroende av att sabotageprogram delar samma kärna för körning som målprogramvaran under en attack.

Secure Enclave-processorn kör en Apple-anpassad version av L4-mikrokärnan. Secure Enclave-processorn är utformad för att fungera effektivt vid lägre klockfrekvenser, vilket ger ytterligare skydd mot klock- och strömangrepp. Secure Enclave-processorn inkluderar, från och med A11 och S4, en minnesskyddad motor och krypterat minne med anti-replay-funktion, säker start, en dedikerad slumptalsgenerator och en egen AES-motor.

## MPE (Memory Protection Engine)

Secure Enclave körs från en dedikerad region i enhetens DRAM-minne. Flera lager av skydd isolerar det Secure Enclave-skyddade minnet från approcessorn.

När enheten startar genererar Secure Enclaves Boot ROM en slumpmässig och tillfällig minneskyddsnyckel för MPE. Varje gång Secure Enclave skriver till sin dedikerade minnesregion krypterar MPE minnesblocket med AES i datorn som är i XEX-läge (xor-encrypt-xor) och beräknar en CMAC-autentiseringstagg (Cipher-based Message Authentication Code) för minnet. MPE lagrar autentiseringstaggen tillsammans med det krypterade minnet. När Secure Enclave läser minnet verifierar MPE autentiseringstaggen. Om autentiseringstaggen matchar avkrypterar MPE minnesblocket. Om taggen inte matchar signalerar MPE ett fel till Secure Enclave. Efter ett minnesautentiseringsfel slutar Secure Enclave att acceptera förfrågningar tills systemet startas om.

Med utgångspunkt i Apple A11 och S4 SoC ger MPE replay-skydd för Secure Enclave-minnet. För att förhindra replay av säkerhetskritiska data lagrar MPE ett unikt engångstal, ett *anti-replay-värde*, för minnesblocket tillsammans med autentiseringstaggen. Anti-replay-värdet används som en ytterligare förvrängning för CMAC-autentiseringstaggen. Anti-replay-värdena för alla minnesblock skyddas av ett integritetsträd grundat i dedikerat SRAM i Secure Enclave. Vid skrivning *uppdaterar* MPE anti-replay-värdet och varje nivå på integritetsträdet upp till SRAM. Vid läsning *verifierar* MPE anti-replay-värdet och varje nivå på integritetsträdet upp till SRAM. Anti-replay-värden som inte matchar hanteras på ett likande sätt som icke-matchande autentiseringstaggar.

På Apples A14, M1 och senare SoC:er har MPE (Memory Protection Engine) stöd för två tillfälliga minnesskyddsnycklar. Den första används till data som är privata för Secure Enclave och den andra används till data som delas med Secure Neural Engine.

MPE körs inbäddat och transparent mot Secure Enclave. Secure Enclave läser och skriver till minne som om det var vanligt okrypterat DRAM, medan någon som observerar utanför Secure Enclave bara ser den krypterade och autentiserade versionen av minnet. Resultatet är ett starkt minnesskydd utan att ge avkall på prestanda eller programvarukomplexitet.

## Secure Enclave-Boot ROM

Secure Enclave innehåller en dedikerad Secure Enclave Boot ROM. Precis som approcessor-Boot ROM består Secure Enclaves Boot ROM av statisk kod som utgör det betrodda och maskinvarubaserade säkerhetssystemet för Secure Enclave.

Vid systemstart tilldelar iBoot en dedikerad minnesregion till Secure Enclave. Innan minnet används initierar Secure Enclaves Boot ROM MPE för att tillhandahålla kryptografiskt skydd åt det Secure Enclave-skyddade minnet.

Approcessorn skickar sedan sepOS-avbilden till Secure Enclaves Boot ROM. Efter att ha kopierat sepOS-avbilden till det Secure Enclave-skyddade minnet kontrollerar Secure Enclaves Boot ROM den kryptografiska hashen och signaturen för avbilden för att verifiera att sepOS är auktoriserat att köras på enheten. Om sepOS-avbilden är korrekt signerad för körning på enheten överför Secure Enclaves Boot ROM styrningen till sepOS. Om signaturen är ogiltig ska Secure Enclaves Boot ROM förhindra ytterligare användning av Secure Enclave fram tills nästa gång kretsen återställs.

På Apples A10 och senare SoC:er låser Secure Enclaves Boot ROM ett hashvärde för sepOS i ett register som är dedikerat för just det syftet. PKA (Public Key Accelerator) använder den här hashen för operativsystembundna nycklar (OS-bundna) nycklar.

## Startövervakaren i Secure Enclave

På Apples A13 och senare SoC:er har Secure Enclave en startövervakare som säkerställer starkare integritet för hashen i startat sepOS.

Vid systemstart förhindrar Secure Enclave-processorns SCIP-konfiguration (System Coprocessor Integrity Protection) att Secure Enclave kör någon annan kod än Secure Enclaves Boot ROM. Startövervakaren förhindrar att Secure Enclave direkt ändrar SCIP-konfigurationen. För att göra inläst sepOS körbart skickar Secure Enclaves Boot ROM en begäran till startövervakaren med adressen och storleken på inläst sepOS. När startövervakaren tar emot begäran återställer den Secure Enclave-processorn, hashar inläst sepOS, uppdaterar SCIP-inställningarna till att tillåta körning av inläst sepOS och startar sedan körningen med den nyinlästa koden. När systemet fortsätter att starta används samma process varje gång ny kod görs körbar. Varje gång uppdaterar startövervakaren en körningshash för startprocessen. Startövervakaren inkluderar också kritiska säkerhetsparametrar i körningshashen.

När starten är klar slutför startövervakaren körningshashen och skickar den till PKA för att användas med operativsystemsbundna nycklar. Den här processen är utformad så att bindning av operativsystemsnycklar inte kan förbigås även om det finns en sårbarhet i Secure Enclaves Boot ROM.

## True Random Number Generator

True Random Number Generator (TRNG) används till att generera säkra slumpdata. Secure Enclave använder TRNG varje gång den genererar en slumpmässig kryptografisk nyckel, slumpmässig nyckelseed eller annan entropi. TRNG baseras på flera ringoscillatorer som efterbehandlas med CTR_DRBG (en algoritm baserad på blockkoder i Counter Mode).

## Kryptografisk rotnyckel

Secure Enclave innehåller en kryptografisk UID-rotnyckel (Unique ID). UID:t är unikt för varje enskild enhet och är inte relaterat till någon annan identifierare på enheten.

Ett slumpmässigt genererat UID byggs in i SoC under tillverkningen. Från och med A9 SoC genereras UID av Secure Enclaves TRNG vid tillverkningen och skrivs till säkringarna med en programvaruprocess som helt och hållet körs i Secure Enclave. Den här processen skyddar UID från att bli synligt utanför enheten under tillverkningen och det är därför inte tillgängligt för åtkomst eller lagring av vare sig Apple eller några av Apples leverantörer.

sepOS använder UID:t till att skydda enhetsspecifika hemligheter. Detta UID gör att data kan knytas till en specifik enhet kryptografiskt. Nyckelhierarkin som skyddar filsystemet innehåller till exempel detta UID, vilket innebär att det inte går att komma åt filerna om den interna SSD-lagringen flyttas fysiskt från en enhet till en annan. Andra skyddade enhetsspecifika hemligheter inkluderar Face ID- eller Touch ID-data. På en Mac får endast fullständigt intern lagring kopplad till AES-motorn den här nivån av kryptering. Exempelvis krypteras varken externa lagringsenheter som ansluts via USB- eller PCIe-baserade lagringsenheter som installeras i 2019 års Mac Pro på det här sättet.

Secure Enclave har också ett enhetsgrupp-ID (GID), vilket är gemensamt för alla enheter som använder en given SoC (exempelvis delar alla enheter som använder Apple A15 SoC samma GID).

UID och GID är inte tillgängliga via JTAG (Joint Test Action Group) eller något annat felsökningsgränssnitt.

## AES-motorn i Secure Enclave

AES-motorn i Secure Enclave är ett maskinvarublock som används till att utföra symmetrisk kryptografi baserad på AES-kod. AES-motorn är utformad för att stå emot informationsläckage genom användning av timing och SPA (Static Power Analysis). Från och med A9:s SoC inkluderar AES-motorn också DPA-motåtgärder (Dynamic Power Analysis).

AES-motorn har stöd för maskinvaru- och programvarunycklar. Maskinvarunycklar härleds från Secure Enclave-UID eller -GID. Dessa nycklar stannar kvar i AES-motorn och är inte synliga ens för sepOS-programvara. Även om programvara kan efterfråga krypterings- och avkrypteringsåtgärder med maskinvarunycklar kan de inte extrahera nycklarna.

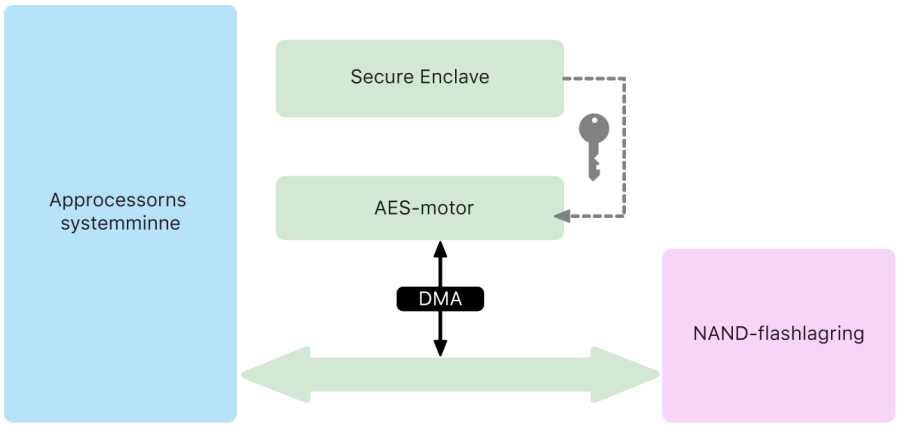
På Apple A10 och nyare SoC inkluderar AES-motorn låsbara startbitar som diversifierar nycklar härledda från UID eller GID. Detta gör det möjligt att villkora dataåtkomst efter enhetens driftläge. Låsbara startbitar kan exempelvis användas till att neka åtkomst till lösenordskyddade data vid start från DFU-läge (Device Firmware Update). Mer information finns i Lösenkoder och lösenord.

## AES-motor

Alla Apple-enheter med Secure Enclave har också en dedikerad AES256-krypteringsmotor (AES-motorn) som är inbyggd i DMA-sökvägen (Direct Memory Access) mellan NAND-flashminnet (icke-flyktigt) och systemets huvudminne, vilket gör filkrypteringen mycket effektiv. På A9- och senare A-serieprocessorer ligger undersystemet för flashminne på en isolerad buss som tillåts åtkomst endast till minne som innehåller användardata via DMA-krypteringsmotorn.

Vid tidpunkten för start genererar sepOS en tillfällig paketeringsnyckel med TRNG. Secure Enclave överför den här nyckeln till AES-motorn via dedikerade ledningar som är utformade för att förhindra åtkomst från någon annan programvara utanför Secure Enclave. sepOS kan sedan använda den tillfälliga paketeringsnyckeln till att paketera filnycklar för användning av approcessorns filsystemsdrivrutin. När filsystemsdrivrutinen läser eller skriver till en fil skickar drivrutinen den paketerade nyckeln till AES-motorn som packar upp nyckeln. AES-motorn exponerar aldrig den uppackade nyckeln för programvara.

*Obs!* AES-motorn är en separat komponent i förhållande till både Secure Enclave och Secure Enclaves AES-motorn, men dess drift är nära knuten till Secure Enclave enligt vad som visas nedan.




## PKA (Public Key Accelerator)

PKA är ett maskinvarublock som används till att utföra asymmetriska krypteringsåtgärder. PKA har stöd för signerings- och krypteringsalgoritmer av typerna RSA och ECC (elliptisk kurvkryptografi). PKA är utformat för att stå emot informationsläckage med hjälp av timing och sidokanalattacker som SPA och DPA.

PKA har stöd för programvaru- och maskinvarunycklar. Maskinvarunycklar härleds från Secure Enclave-UID eller -GID. Dessa nycklar stannar kvar i PKA och är inte synliga ens för sepOS-programvara.

Från och med på A13 SoC har PKA-krypteringsimplementeringar visat sig vara matematiskt korrekta med formella verifieringstekniker.

På Apples A10 och senare SoC:er har PKA stöd för operativsystemsbundna nycklar, också kallat SKP (Sealed Key Protection). De här nycklarna genereras via en kombination av enhetens UID och hashvärdet för det sepOS som körs på enheten. Hashvärdet tillhandahålls av Secure Enclaves Boot ROM, eller av startövervakaren i Secure Enclave på Apples A13 och senare SoC:er. Nycklarna används även till att verifiera sepOS-versionen vid förfrågningar till vissa Apple-tjänster, och till att förbättra skyddet av lösenkodsskyddade data, genom att hjälpa till att förhindra åtkomst till nyckelmaterial om kritiska ändringar görs i systemet utan att ha auktoriserats av användaren.

## Säker icke-flyktig lagring

Secure Enclave är utrustad med en dedikerad enhet för icke-flyktig lagring. Den säkra icke-flyktiga lagringsenheten är ansluten till Secure Enclave via en dedikerad I2C-buss så att den endast är åtkomlig för Secure Enclave. Alla krypteringsnycklar för användardata är baserade i entropi som lagras i den icke-flyktiga Secure Enclave-lagringen.

På enheter som har en A12, S4 och senare SoC parkopplas Secure Enclave med en integrerad krets för säker lagring av entropi, SSC (Secure Storage Component). SSC är i sin tur utformat med statisk ROM-kod, en maskinvarubaserad slumptalsgenerator, en kryptografisk nyckel som är unik för varje enhet, kryptografimotorer och detektering av fysisk manipulering. Secure Enclave och SSC kommunicerar via ett krypterat och autentiserat protokoll som tillhandahåller exklusiv åtkomst till entropin.

Enheter som lanserats första gången hösten 2020 eller senare är utrustade med andra generationens SSC. Med den andra generationens SSC tillkommer låsboxar för räknare. Varje låsbox för räknare lagrar ett 128-bitars salt, en 128-bitars lösenkodsverifierare, en 8-bitars räknare och ett 8-bitars maxförsöksvärde. Åtkomst till låsboxar för räknare sker via ett krypterat och autentiserat protokoll.

Låsboxar för räknare innehåller den entropi som behövs för att låsa upp lösenordskyddade användardata. För att komma åt användardata måste dess parkopplade Secure Enclave härleda korrekt entropivärde för lösenkoden från användarens lösenkod och UID för Secure Enclave. Det går inte att ta reda på användarens lösenkod genom att använda upplåsningsförsök som skickas från en annan källa än dess parkopplade Secure Enclave. Om gränsen för lösenordsförsök överskrids (t.ex. 10 försök på iPhone) raderas lösenordsskyddade data fullständigt av SSC.

För att skapa en låsbox för räknare skickar Secure Enclave lösenkodens entropivärde och det maximala värdet för antalet försök till SSC. SSC genererar saltvärdet genom att använda sin slumptalsgenerator. Den härleder sedan ett verifierarvärde för lösenkoden och ett entropivärde för låsboxen från den tillhandahållna lösenkodsentropin, den unika kryptografiska nyckeln för SSC samt saltvärdet. SSC initierar låsboxen för räknare med talet 0, det tillhandahållna maxvärdet för antal försök, det härledda verifierarvärdet för lösenkod och saltvärdet. SSC returnerar sedan det genererade entropivärdet för låsboxen till Secure Enclave.

När låsboxens entropivärde senare ska hämtas från en låsbox för räknare skickar Secure Enclave lösenkodsentropin till SSC. SSC räknar först upp räknaren för låsboxen. Om värdet för den uppräknade räknaren överskrider maxvärdet för antal försök raderar SSC låsboxen för räknare helt och hållet. Om antalet försök inte har uppnått det högst tillåtna värdet försöker SSC härleda verifierarvärdet för lösenkoden och entropivärdet för låsboxen med samma algoritm som används för att skapa låsboxen för räknare. Om det härledda verifierarvärdet för lösenkoden matchar det lagrade verifierarvärdet för lösenkoden returnerar SSC entropivärdet för låsboxen till Secure Enclave och nollställer räknaren.

De nycklar som används till att komma åt lösenordskyddade data har sin grund i den entropi som lagras i låsboxar för räknare. Mer information finns i Dataskydd i översikt.

Den säkra icke-flyktiga lagringen används för alla anti-replay-tjänster i Secure Enclave. Anti-replay-tjänster i Secure Enclave används till att återkalla data över aktiviteter som markerar anti-replay-gränser, bland annat följande:

- Lösenkodsändring
- Aktivering eller avaktivering av Face ID eller Touch ID
- Tillägg eller borttagning av ett Face ID-ansikte eller ett Touch ID-fingeravtryck
- Nollställning av Face ID eller Touch ID
- Tillägg eller borttagning av ett Apple Pay-kort
- Radering av allt innehåll och alla inställningar

På arkitekturer som inte har någon SSC används EEPROM (Electrically Erasable Programmable Read-Only Memory) för att tillhandahålla säkra lagringstjänster åt Secure Enclave. Precis som SSC är EEPROM kopplat till och endast åtkomligt från Secure Enclave, men det innehåller inte några dedikerade funktioner för maskinvarusäkerhet och garanterar inte heller exklusiv åtkomst till entropi (utöver de fysiska kopplingsegenskaperna) eller någon låsboxfunktion för räknare.

## Secure Neural Engine

På enheter med Face ID (inte Touch ID) omvandlar Secure Neural Engine 2D-bilder och djupkartor till en matematisk bild av en användares ansikte.

På SoC-modellerna A11 till A13 är Secure Neural Engine integrerad i Secure Enclave. Secure Neural Engine använder DMA (Direct Memory Access) för högre prestanda. En IOMMU (Input-output Memory Management Unit) som sepOS-kärnan styr begränsar den direkta åtkomsten till auktoriserade minnesregioner.

Från och med A14, M1 och senare har Secure Neural Engine implementerats som ett säkert läge i approcessorns Neural Engine. En dedikerad styrenhet för maskinvarusäkerhet växlar mellan approcessor- och Secure Enclave-åtgärder och nollställer Neural Engine-statusen vid varje övergång så att Face ID-data hålls säkra. En dedikerad motor använder minneskryptering, -autentisering och -åtkomstkontroll. Den använder samtidigt en separat kryptografisk nyckel och ett separat minnesintervall till att begränsa Secure Neural Engine till auktoriserade minnesregioner.

## Ström- och klockövervakare

All elektronik är utformad för att fungera inom ett begränsat spännings- och frekvensområde. Produkterna kan sluta att fungera om de används utanför området, och då kan det hända att säkerhetskontrollerna förbigås. För att säkerställa att spänningen och frekvensen är inom ett säkert intervall har Secure Enclave övervakningskretsar. Övervakningskretsarna är utformade så att de kan fungera i ett mycket större driftområde än resten av Secure Enclave. Om övervakarna upptäcker en ogiltig driftspunkt stoppas klockorna i Secure Enclave automatiskt och startas inte förrän vid nästa SoC-nollställning.

## Funktionssammanfattning för Secure Enclave

*Obs!* A12-, A13-, S4- och S5-produkter som lanserats första gången hösten 2020 har andra generationens SSC, medan tidigare produkter som är baserade på dessa SoC:er har första generationen SSC.

| SoC | MPE (Memory Protection Engine) | Säker lagring | AES-motor | PKA |
|---|---|---|---|---|
| A8 | Kryptering och autentisering | EEPROM | Ja | Nej |
| A9 | Kryptering och autentisering | EEPROM | DPA-skydd | Ja |
| A10 | Kryptering och autentisering | EEPROM | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| A11 | Kryptering, autentisering och förhindrande av replay | EEPROM | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| A12 (Apple-enheter som lanserats före hösten 2020) | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 1 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| A12 (Apple-enheter som lanserats efter hösten 2020) | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 2 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| A13 (Apple-enheter som lanserats före hösten 2020) | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 1 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar och startövervakare |
| A13 (Apple-enheter som lanserats efter hösten 2020) | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 2 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar och startövervakare |
| A14–A17 | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 2 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar och startövervakare |
| S3 | Kryptering och autentisering | EEPROM | DPA-skydd och låsbara startbitar | Ja |
| S4 | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 1 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| S5 (Apple-enheter som lanserats före hösten 2020) | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 1 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| S5 (Apple-enheter som lanserats efter hösten 2020) | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 2 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |

| SoC | MPE (Memory Protection Engine) | Säker lagring | AES-motor | PKA |
|---|---|---|---|---|
| S6–S9 | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 2 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| T2 | Kryptering och autentisering | EEPROM | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar |
| M1, M2, M3 | Kryptering, autentisering och förhindrande av replay | Secure Storage Component generation 2 | DPA-skydd och låsbara startbitar | Operativsystembundna nycklar och startövervakare |

# Face ID och Touch ID

## Face ID- och Touch ID-säkerhet

Lösenkoder och lösenord är avgörande för att Apple-enheter ska vara säkra. Samtidigt behöver användare smidig åtkomst till sina enheter, ofta fler än hundra gånger om dagen. Biometriska autentiseringar är ett sätt att behålla säkerheten hos starka lösenkoder, eller till och med skapa ännu starkare lösenkoder eller lösenord eftersom de inte behöver anges manuellt, samtidigt som de gör att det snabbt och smidigt går att låsa upp enheten med ett fingeravtryck eller ögonkast. Face ID och Touch ID ersätter inte en lösenkod eller ett lösenord, men de gör det oftast enklare och snabbare att låsa upp enheten.

Apples biometriska säkerhetsarkitektur bygger på en strikt uppdelning av ansvarsområden mellan den biometriska sensorn och Secure Enclave samt en säker anslutning mellan dem. Sensorn tar den biometriska bilden och överför den säkert till Secure Enclave. Under registreringen behandlar, krypterar och lagrar Secure Enclave motsvarande malldata för Face ID och Touch ID. Under matchningen jämför Secure Enclave inkommande data från den biometriska sensorn med de lagrade mallarna för att avgöra om den ska låsa upp enheten eller svara att en matchning är giltig (för Apple Pay, i appar och andra tillfällen då Face ID och Touch ID används). Arkitekturen har stöd för enheter som har både en sensor och Secure Enclave (som iPhone, iPad och en hel del Mac-system). Den kan också fysiskt dela sensorn i en kringutrustning som sedan parkopplas säkert med Secure Enclave i en Mac med Apple Silicon.

### Face ID-säkerhet

Face ID låser tryggt och säkert upp Apple-enheter som stöds med ett ögonkast. Det är en intuitiv och säker autentiseringsmetod som är möjlig tack vare TrueDepth-kamerasystemet som drar nytta av avancerad teknik för att kartlägga de geometriska punkterna i användarens ansikte. Face ID använder neuronnät till att bedöma uppmärksamhet, matchning och skydda mot bedrägeriförsök så att en användare kan låsa upp telefonen genom att titta på den även om användaren har ett munskydd när den använder enheter som stöds. Face ID anpassas automatiskt efter förändringar i utseendet och skyddar användarens biometriska data så att de förblir privata.

Face ID är utformat för att bekräfta användarens uppmärksamhet, tillhandahålla robust autentisering med få falska matchningar och begränsa digitala och fysiska bedrägeriförsök.

TrueDepth-kameran letar automatiskt efter användarens ansikte när han eller hon väcker en Apple-enhet som har Face ID (genom att höja den eller trycka på skärmen), liksom när enheten försöker autentisera användaren för att visa en inkommande notis eller när en app som stöds begär Face ID-autentisering. När ett ansikte upptäcks bekräftar Face ID uppmärksamheten och avsikten att låsa upp genom att upptäcka att användarens ögon är öppna och att uppmärksamheten är riktad mot enheten. Uppmärksamhetskontrollen för Face ID avaktiveras när VoiceOver aktiveras och kan vid behov även avaktiveras separat. Upptäckt av uppmärksamhet krävs alltid när Face ID används med ett munskydd.

När TrueDepth-kameran har bekräftat att det finns ett uppmärksamt ansikte projicerar den och läser av tusentals infraröda prickar för att skapa en djupkarta av ansiktet tillsammans med en infraröd 2D-bild. Dessa data används till att skapa en sekvens med 2D-bilder och djupkartor som signeras digitalt och skickas till Secure Enclave. För att avvärja både digitala och fysiska bedrägeriförsök slumpar TrueDepth-kameran ordningen på 2D-bilder och sparade djupkartor samt projicerar ett enhetsspecifikt slumpmässigt mönster. En del av Secure Neural Engine – skyddad i Secure Enclave – omvandlar dessa data till en matematisk representation och jämför sedan denna representation med de ansiktsdata som har registrerats. Dessa registrerade ansiktsdata utgör i sig självt en matematisk representation av användarens ansikte som har registrerats ur olika vinklar.

## Touch ID-säkerhet

Touch ID är ett system för fingeravtryckskläsning som ger snabb och säker tillgång till Apple-enheter som stöds. Tekniken läser av fingeravtryck ur alla vinklar och lär sig mer om användarens fingeravtryck över tid. Sensorn kartlägger kontinuerligt fingeravtrycket i och med att den upptäcker nya, överlappande noder vid varje användning.

Apple-enheter med en Touch ID-sensor kan låsas upp med ett fingeravtryck. Touch ID ersätter inte behovet av en enhetslösenkod eller ett användarlösenord, utan det krävs fortfarande när enheten startas, startas om eller när användaren loggar ut (på en Mac). I vissa appar kan Touch ID också användas istället för en enhets lösenkod eller användarlösenord – till exempel för att låsa upp lösenordsskyddade anteckningar i Anteckningar, för att låsa upp nyckelringsskyddade webbplatser och för att låsa upp applösenord som stöds. I vissa fall krävs dock alltid en enhetslösenkod eller ett användarlösenord (t.ex. vid byte av en befintlig enhetslösenkod eller ett användarlösenord eller vid borttagning av registrerade fingeravtryck och när nya fingeravtryck ska läggas till).

När fingeravtryckssensorn känner av ett finger utlöses en avancerad avläsningsmatris som skannar av fingret och skickar bilden till Secure Enclave. Vilken kanal som används till att skydda anslutningen varierar. Det beror på om Touch ID-sensorn är inbyggd i enheten med Secure Enclave eller finns i en separat kringutrustning.

Medan det inskannade fingeravtrycket vektoriseras för analys sparas det tillfälligt i krypterat minne inuti Secure Enclave och raderas sedan. Vid analysen används så kallad subdermal ridge flow angle mapping, vilket är en reducerande process där de detaljerade fingerdata som skulle krävas för att återskapa användarens fingeravtryck kastas. Under registreringen lagras kartan över noder som skapas i ett krypterat format som endast kan läsas av Secure Enclave som en mall så att den kan jämföras med framtida matchningar. Den innehåller ingen identitetsinformation. Dessa data lämnar aldrig enheten. De skickas inte till Apple och ingår inte i säkerhetskopior av enheten.

### Inbyggd Touch ID-kanalsäkerhet

Kommunikationen mellan Secure Enclave och den inbyggda Touch ID-sensorn sker via en SPI-buss (Serial Peripheral Interface). Processorn vidarebefordrar data till Secure Enclave, men kan inte läsa dem. De krypteras och autentiseras med en sessionsnyckel som förhandlas med hjälp av enhetens delade nyckel för varje Touch ID-sensor och dess motsvarande Secure Enclave. Den delade nyckeln är stark, slumpmässig och unik för varje Touch ID-sensor. Vid utbytet av sessionsnycklar används AES-nyckelpaketering där båda sidor tillhandahåller en slumpmässig nyckel som upprättar sessionsnyckeln och använder transportkryptering som ger både autentisering och konfidentialitet (med AES-CCM).

## Magic Keyboard med Touch ID

Magic Keyboard med Touch ID (och Magic Keyboard med Touch ID och numeriskt tangentbord) tillhandahåller en Touch ID-sensor i ett externt tangentbord som kan användas med alla Mac-datorer som har Apple Silicon. Magic Keyboard med Touch ID fungerar som den biometriska sensorn. Det lagrar inga biometriska mallar, utför ingen biometrisk matchning och tvingar inte igenom några säkerhetspolicyer (t.ex. att ett lösenord måste anges när ingen upplåsning har gjorts under 48 timmar). Touch ID-sensorn i Magic Keyboard med Touch ID måste parkopplas säkert med Secure Enclave i datorn innan den kan användas. Secure Enclave utför sedan registrerings- och matchningsåtgärderna och tvingar igenom säkerhetspolicyer på samma sätt som för inbyggda Touch ID-sensorer. Apple utför parkopplingen i fabriken för Magic Keyboard-tangentbord med Touch ID som levereras med Mac-datorer. Användare kan också parkoppla om det behövs. Ett Magic Keyboard med Touch ID kan bara parkopplas säkert med en dator åt gången, men en dator kan upprätthålla säkra parkopplingar med upp till fem olika Magic Keyboard-tangentbord med Touch ID.

Magic Keyboard med Touch ID och inbyggda Touch ID-sensorer är kompatibla. Om ett finger som har registrerats med en inbyggd Touch ID-sensor avläses på ett Magic Keyboard med Touch ID behandlar Secure Enclave i datorn matchningen och tvärtom.

Tangentbordet har en maskinvarubaserad PKA-blockering (Public Key Accelerator) så att datorns Secure Enclave och Magic Keyboard med Touch ID ska kunna parkopplas säkert och sedan kommunicera. Den tillhandahåller attestering och maskinvarubaserade nycklar utför de nödvändiga krypteringsprocesserna.

### Säker parkoppling

Ett Magic Keyboard med Touch ID måste parkopplas säkert med datorn innan det kan användas till Touch ID-åtgärder. För att parkoppla utbyter Secure Enclave i datorn och PKA-blocket i Magic Keyboard med Touch ID publika nycklar som finns hos den betrodda Apple-certifikatutfärdare. De använder attesteringsnycklar i maskinvaran och tillfälliga ECDH till att säkert attestera sin identitet. På datorn skyddas dessa data av Secure Enclave och på Magic Keyboard med Touch ID av PKA-blocket. När en säker parkoppling har gjorts krypteras alla Touch ID-data som kommuniceras mellan datorn och Magic Keyboard med Touch ID med AES-GCM med en 256-bitars nyckellängd och tillfälliga ECDH-nycklar med en NIST P-256-kurva som baseras på de lagrade identiteterna. För mer information om hur du använder tangentbordet i trådlöst läge, se Bluetooth-säkerhet.

### Säker avsikt att parkoppla

Första gången användaren ska utföra vissa Touch ID-åtgärder, t.ex. registrera ett nytt fingeravtryck, måste den fysiskt bekräfta sin avsikt att använda Magic Keyboard med Touch ID med datorn. Avsikten bekräftas fysiskt genom att trycka två gånger på datorns strömbrytare när det anges i användargränssnittet eller genom att använda ett fingeravtryck som tidigare har registrerats för datorn. Mer information finns i Säker avsikt och säkra anslutningar till Secure Enclave.

Apple Pay-transaktioner kan godkännas med en Touch ID-matchning. De kan också godkännas genom att ange macOS-användarens lösenord och trycka två gånger på Touch ID-knappen på Magic Keyboard med Touch ID. Med den sista metoden kan användaren bekräfta avsikten fysiskt även utan en Touch ID-matchning.

### Kanalsäkerhet för Magic Keyboard med Touch ID

För att säkerställa en säker kommunikationskanal mellan Touch ID-sensorn i Magic Keyboard med Touch ID och Secure Enclave i den parkopplade datorn krävs följande:

- Den säkra parkopplingen mellan PKA-blocket för Magic Keyboard med Touch ID och Secure Enclave enligt beskrivningen ovan
- En säker kanal mellan Magic Keyboard med Touch ID-sensorn och dess PKA-block

Den säkra kanalen mellan Magic Keyboard med Touch ID-sensorn och dess PKA-block skapas i fabriken med en unik nyckel som delas mellan båda. (Det är samma teknik som används till att skapa den säkra kanalen mellan Secure Enclave i datorn och dess inbyggda sensor på Mac-datorer med inbyggd Touch ID.)

## Face ID, Touch ID, lösenkoder och lösenord

För att kunna använda Face ID eller Touch ID måste användarna ställa in sina enheter så att det krävs en lösenkod eller ett lösenord för att låsa upp dem. När Face ID eller Touch ID känner igen en matchning låser den upp enheten utan att fråga efter lösenkoden eller lösenordet. Det här gör det mindre besvärligt att använda en lång och komplex lösenkod eller ett komplext lösenord eftersom användaren inte behöver ange lösenkoden eller lösenordet lika ofta. Face ID och Touch ID ersätter inte en lösenkod eller ett lösenord. De ger istället smidig tillgång till enheten inom genomtänkta gränser och tidsbegränsningar. Det här är viktigt eftersom en stark lösenkod eller ett starkt lösenord utgör grunden för hur en användares iPhone-, iPad-, Mac- eller Apple Watch-enhet kryptografiskt skyddar den användarens data.

### När en lösenkod eller ett lösenord krävs på en enhet

Användare kan när som helst använda lösenkod eller lösenord istället för Face ID eller Touch ID, men i några fall tillåts inte biometri. Följande säkerhetskritiska åtgärder kräver alltid lösenkod eller lösenord:

- Uppdatering av programvara
- Radering av enheten
- Visning eller ändring av lösenkodsinställningar
- Installation av konfigurationsprofiler
- Upplåsning av panelen Integritet och säkerhet i Systeminställningar (macOS 13 eller senare) på Mac

- Upplåsning av panelen Säkerhet och integritet i Systeminställningar (macOS 12 eller tidigare) på Mac
- Upplåsning av panelen Användare och grupper i Systeminställningar (macOS 13 eller senare) på Mac (om FileVault är aktiverat)
- Upplåsning av panelen Användare och grupper i Systeminställningar (macOS 12 eller tidigare) på Mac (om FileVault är aktiverat)

En lösenkod eller ett lösenord krävs också om enheten befinner sig i något av följande lägen:

- Om användaren precis har slagit på eller startat om enheten.
- Om användaren har loggat ut från sitt Mac-konto (eller inte har loggat in än).
- Om användaren inte har låst upp sin enhet under mer än 48 timmar.
- Om användaren inte har använt lösenkod eller lösenord till att låsa upp sin enhet under 156 timmar (6,5 dagar) och användaren inte har låst upp enheten med biometri under 4 timmar.
- Om enheten har mottagit ett fjärrlåsningskommando.
- När användaren har lämnat avstängning/Nödsamtal SOS genom att hålla in någon av volymknapparna och vilo-/väckningsknappen samtidigt under två sekunder och sedan tryckt på Avbryt.
- Efter fem misslyckade biometriska matchningsförsök (enheten kan dock erbjuda möjligheten att ange en lösenkod eller ett lösenord istället för att använda biometri efter ett mindre antal misslyckanden).

När Face ID med munskydd aktiveras på en iPhone är det tillgängligt i 6,5 timmar efter en av de följande användaråtgärderna:

- Lyckat Face ID-matchningsförsök (med eller utan munskydd)
- Validering av enhetslösenkod
- Enhetsupplåsning med Apple Watch

När någon av de här åtgärderna utförs förlängs perioden med ytterligare 6,5 timmar.

När Face ID eller Touch ID är aktiverat på en iPhone eller iPad låses enheten direkt när du trycker på vilo-/väckningsknappen och varje gång enheten försätts i viloläge. Face ID och Touch ID kräver en lyckad matchning, eller att du anger lösenkoden, vid varje väckning.

Sannolikheten för att en slumpmässig person i befolkningen kan låsa upp en användares iPhone eller iPad är mindre än 1 på 1 000 000 med Face ID, inklusive när Face ID med munskydd är aktiverat. För en användares iPhone, iPad, Mac-modeller med Touch ID och modeller som är parkopplade med ett Magic Keyboard är sannolikheten mindre än 1 på 50 000. Sannolikheten ökar om du har flera registrerade fingeravtryck (upp till 1 på 10 000 med fem fingeravtryck) eller utseenden (upp till 1 på 500 000 med två utseenden). Som ett extra skydd tillåter både Face ID och Touch ID bara fem misslyckade matchningsförsök innan en lösenkod eller ett lösenord krävs för att få tillgång till användarens enhet eller konto. Med Face ID är sannolikheten för en falsk matchning högre för:

- Tvillingar och syskon som liknar användaren
- Barn under 13 år (eftersom deras distinkta ansiktsdrag troligtvis inte är fullt utvecklade)

Sannolikheten ökar ytterligare i dessa fall när Face ID med munskydd används. Om en användare är orolig för falsk matchning rekommenderar Apple användning av en lösenkod för autentisering.

## Säkerhet för ansiktsmatchning

Ansiktsmatchningen utförs i Secure Enclave med hjälp av neuronnät som har specialtränats för det ändamålet. Apple utvecklade neuronnäten för ansiktsmatchning med hjälp av över en miljard bilder, inklusive infraröda bilder och djupbilder som har samlats in i studier som genomförts med deltagarnas informerade samtycke. Apple arbetade sedan med deltagare från hela världen för att få med en representativ samling personer med tanke på kön, ålder, etnicitet och andra faktorer. Studierna utökades vid behov för att få en hög tillförlitlighet gällande olika typer av användare. Face ID är utformat för att fungera med hattar, halsdukar, glasögon, kontaktlinser och många slags solglasögon. Face ID stöder även upplåsning med munskydd på iPhone från och med iPhone 12 och iOS 15.4 eller senare. Dessutom fungerar det inomhus, utomhus och till och med i totalt mörker. Ytterligare ett neuronnät är tränat för att upptäcka och avvärja försök att låsa upp enheten med hjälp av bilder eller masker. Face ID-data, inklusive matematiska representationer av en användares ansikte, krypteras och är bara tillgängliga för Secure Enclave. Dessa data lämnar aldrig enheten. De skickas inte till Apple och ingår inte i säkerhetskopior av enheten. Följande Face ID-data sparas och krypteras, endast för Secure Enclave, vid normal användning:

- De matematiska representationerna av en användares ansikte som beräknats vid registreringen.
- De matematiska representationerna av en användares ansikte som beräknats under vissa upplåsningsförsök om Face ID bedömer att de kan användas till att förbättra kommande matchningar.

Ansiktsbilder som har tagits vid normal användning sparas inte, utan kasseras omedelbart när den matematiska representationen har beräknats för antingen registrering i Face ID eller för jämförelse med registrerade Face ID-data.

### Förbättra Face ID-matchningar

Face ID förbättrar matchningsprestanda och håller jämna steg med de naturliga förändringarna i ett ansikte och utseende genom att utöka de lagrade matematiska representationerna över tid. Efter en matchning kan Face ID använda den nyligen beräknade matematiska representationen (om kvaliteten är tillräckligt hög) för ett begränsat antal ytterligare matchningar innan dessa data kasseras. Om Face ID däremot inte känner igen ett ansikte, men kvaliteten på matchningen överstiger ett visst tröskelvärde och användaren omedelbart följer upp genom att ange sin lösenkod, gör Face ID ytterligare en avläsning och utökar dess registrerade Face ID-data med den nyligen beräknade matematiska representationen. Dessa nya Face ID-data kasseras om användaren slutar matcha mot dem eller efter ett begränsat antal matchningar. Nya data kasseras även när alternativet för att nollställa Face ID väljs. Tack vare de här utökningsprocesserna kan Face ID hålla jämna steg med stora förändringar av en användares skäggväxt eller sminkning, samtidigt som falska matchningar minimeras.

# Användningsområden för Face ID och Touch ID

## Låsa upp en enhet eller ett användarkonto

Om Face ID eller Touch ID är avstängt när en enhet eller ett konto låses raderas nycklarna som förvaras i Secure Enclave för den högsta dataskyddsklassen. Filerna och nyckelringsobjekten i den klassen blir inte tillgängliga förrän användaren låser upp enheten eller kontot genom att ange sin lösenkod eller sitt lösenord.

När Face ID eller Touch ID är aktiverat kastas inte nycklarna när enheten eller kontot låses. Istället paketeras de och förvaras tillsammans med en nyckel som ges till Face ID- eller Touch ID-undersystemet inuti Secure Enclave. När en användare försöker låsa upp enheten eller kontot, och den upptäcker en matchning, tillhandahåller den nyckeln för att packa upp dataskyddsnycklarna och enheten låses upp. Den här processen ger extra skydd eftersom den kräver att undersystemen för dataskydd och Face ID eller Touch ID samarbetar för att låsa upp enheten.

När enheten startar om går de nycklar som krävs för att låsa upp enheten eller kontot med Face ID eller Touch ID förlorade. De kastas av Secure Enclave efter att något villkor som kräver att en lösenkod eller ett lösenord anges blir uppfyllt.

## Säkra inköp med Apple Pay

Användaren kan också använda Face ID och Touch ID med Apple Pay för att smidigt och säkert betala för inköp i butiker, appar och på webben:

- *Med Face ID i butiker:* När användaren vill auktorisera en butiksbetalning med Face ID måste han eller hon först bekräfta sin avsikt att betala genom att dubbelklicka på sidoknappen. Dubbelklicket bekräftar användarens avsikt med en fysisk gest som är direkt kopplad till Secure Enclave och motstår förfalskning via en skadlig process. Sedan auktoriserar användaren med Face ID innan han eller hon placerar enheten i närheten av den kontaktlösa betalningsläsaren. Det går att välja en annan Apple Pay-betalningsmetod som kräver autentisering efter Face ID-autentiseringen, men användaren behöver inte dubbelklicka på sidoknappen igen.
- *Med Face ID i appar och på webben:* När användaren vill auktorisera en betalning i en app eller på webben måste han eller hon först bekräfta sin avsikt att betala genom att dubbelklicka på sidoknappen och sedan autentisera betalningen med Face ID. Om Apple Pay-transaktionen inte slutförs inom 60 sekunder efter att användaren har dubbelklickat på sidoknappen måste användaren bekräfta sin avsikt att betala genom att dubbelklicka igen.
- *Med Touch ID:* För Touch ID bekräftar användaren sin avsikt att betala med gesten som aktiverar Touch ID-sensorn i kombination med matchningen av användarens fingeravtryck.

## Använda API:er tillhandahållna av systemet

Appar från tredje part kan dra nytta av systemomfattande API:er för att be användare att autentisera med Face ID eller Touch ID eller en lösenkod eller ett lösenord. Appar som redan stöder Touch ID stöder automatiskt Face ID utan några ändringar. När Face ID eller Touch ID används får appen bara ett meddelande om huruvida autentiseringen lyckades eller inte. Den får inte tillgång till Face ID, Touch ID eller till de data som är kopplade till den registrerade användaren.

### Skydda nyckelringsobjekt

Nyckelringsobjekt kan också skyddas med Face ID eller Touch ID så att de endast släpps av Secure Enclave när användaren gör en matchning eller anger enhetslösenkoden eller kontolösenordet. Apputvecklare har API:er som bekräftar att användaren har angett en lösenkod eller ett lösenord innan de kräver Face ID, Touch ID eller en lösenkod eller ett lösenord för att låsa upp nyckelringsobjekt. Apputvecklare kan göra följande:

- Kräva att API-anrop om autentisering inte faller tillbaka till ett applösenord eller enhetens lösenkod. Skicka en förfrågan om ifall en användare är registrerad och då tillåta att Face ID eller Touch ID används som en andra faktor i appar som kräver hög säkerhet.
- Generera och använda ECC-nycklar (elliptisk kurvkryptografi) i Secure Enclave som kan skyddas med Face ID eller Touch ID. Åtgärder med de här nycklarna utförs alltid i Secure Enclave efter att Secure Enclave har auktoriserat användningen.

### Göra och godkänna inköp

Användare kan också konfigurera Face ID eller Touch ID för användning vid inköp på iTunes Store, App Store och Apple Books med mera så att användaren slipper ange sitt Apple-ID-lösenord. När inköp görs verifierar Secure Enclave att en biometrisk autentisering inträffade och släpper sedan ECC-nycklar som används till att signera butikens begäran.

## Säker avsikt och säkra anslutningar till Secure Enclave

Säker avsikt är ett sätt att bekräfta en användares avsikt utan interaktion med operativsystemet eller approcessorn. Anslutningen är en fysisk länk, från en fysisk knapp till Secure Enclave, som finns i följande enheter:

- iPhone X och senare
- Apple Watch Series 1 eller senare
- iPad Pro (alla modeller)
- iPad Air (2020)
- Mac-datorer med Apple Silicon

Med den här länken kan användare bekräfta sin avsikt att slutföra en åtgärd på ett sätt som inte ens programvara som körs med rotbehörigheter eller i kärnan kan imitera.

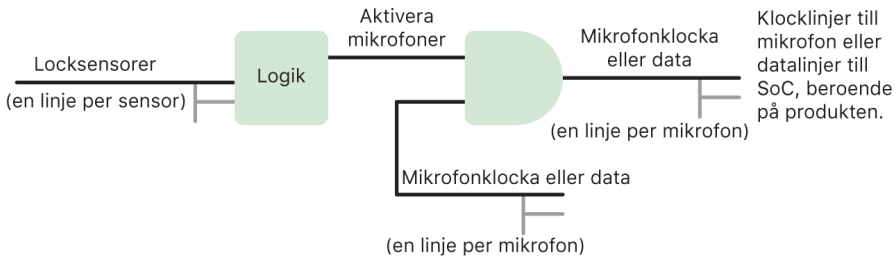
Funktionen används till att bekräfta användarens avsikt under Apple Pay-transaktioner och till att slutföra parkopplingen mellan Magic Keyboard med Touch ID och en Mac med Apple Silicon. Två snabba tryck på rätt knapp (för Face ID) eller en fingeravtrycksavläsning (för Touch ID) när det krävs i användargränssnittet bekräftar användarens avsikt. Mer information finns i Säkra inköp med Apple Pay. En liknande mekanism, som baseras på Secure Enclave och den fasta T2-programvaran, kan användas på MacBook-modeller med Apple T2-säkerhetskrets och utan Touch Bar.

# Maskinvarubortkoppling av mikrofonen

Alla bärbara Apple Silicon-baserade Mac-datorer och alla bärbara Intel-baserade Mac-datorer har en funktion för maskinvarubortkoppling som avaktiverar mikrofonen när locket stängs. På alla 13-tums MacBook Pro- och MacBook Air-datorer med T2-krets, alla bärbara MacBook-datorer med T2-krets från 2019 eller senare samt alla bärbara MacBook-datorer med Apple Silicon utförs den här frånkopplingen endast i maskinvaran. Bortkopplingen förhindrar att någon programvara – inte ens programvara med rot- eller kärnbehörighet i macOS, och inte heller programvaran i T2-kretsen eller annan fast programvara – kan aktivera mikrofonen när locket är stängt. (Kameran kopplas inte bort i maskinvaran eftersom sökaren ändå inte kan se något när locket är stängt.)

iPad-modeller från och med början av 2020 har även funktionen för maskinvarubortkoppling av mikrofonen. När ett MFi-anpassat fodral (inklusive de som säljs av Apple) används med iPad och användaren stänger fodralet kopplas mikrofonen bort i maskinvaran. Det förhindrar att ljuddata från mikrofonen blir tillgängliga för programvara – även för programvara med rot- eller kärnbehörighet i iPadOS och för fast programvara oavsett enhet.

Skydden i den här delen implementeras direkt med maskinvarulogik i enlighet med följande kretsdiagram:




I varje produkt där mikrofonen kopplas bort i maskinvaran känner en eller flera sensorer av att locket eller fodralet rent fysiskt stängs med hjälp av någon fysisk egenskap (t.ex. en hall-effektsensor eller en vinkelsensor på gångjärn) vid händelsen. För sensorer där kalibrering krävs ställs parametrar in under tillverkningen av enheten, och kalibreringsprocessen inkluderar en icke-reversibel maskinvarulåsning som hindrar eventuella senare ändringar av känsliga parametrar i sensorn. Dessa sensorer sänder ut en direkt maskinvarusignal som går genom en enkel uppsättning icke-omprogrammeringsbar maskinvarulogik. Denna logik tillhandahåller debounce, hysteres och/eller en fördröjning på upp till 500 ms innan mikrofonen avaktiveras. Beroende på produkt kan signalen implementeras antingen genom avaktivering av ledningarna som transporterar data mellan mikrofonen och SoC eller genom avaktivering av en av inmatningsledningarna till mikrofonmodulen som tillåter den att vara aktiv, t.ex. klockledningen eller motsvarande effektiv styrning.

# Expresskort med strömsparläge

Om iOS inte är igång eftersom iPhone måste laddas kan det fortfarande finnas tillräckligt mycket ström i batteriet för att genomföra transaktioner med expresskort. Vissa iPhone-modeller stöder automatiskt den här funktionen med:

- Ett betal- eller resekort som är valt som expressresekort
- Åtkomstkort med Expressläge aktiverat

När sidoknappen trycks in visar batterisymbolen att batterinivån är låg och text meddelar att det finns expresskort som kan användas. NFC-styrenheten genomför transaktioner med expresskort under likadana förhållanden som när iOS är igång, med undantag för att transaktionerna endast bekräftas med haptiska notiser (ingen visuell notis visas). På andra generationens iPhone SE kan det dröja några sekunder innan slutförda transaktioner visas på skärmen. Den här funktionen är inte tillgänglig när användaren väljer att stänga av enheten på vanligt sätt.

# Systemsäkerhet

## Systemsäkerhet i översikt

Systemsäkerheten bygger på de unika egenskaperna hos Apples maskinvara och ansvarar för att styra åtkomsten till systemresurser i Apple-enheter utan att kompromissa med användarvänligheten. Systemsäkerhet omfattar startprocessen, programuppdateringar och skydd av datorsystemresurser som processor, minne, hårddisk, programvaror och lagrade data.

De senaste versionerna av Apple-operativsystem är de säkraste. En viktig del av Apple-säkerheten är *säker start* som skyddar systemet från sabotageprogram vid start. En säker startprocess påbörjas i kretsar och bygger sedan en tillförlitlighetskedja genom programvaran. Varje steg är utformat på ett sätt som säkerställer att nästa steg fungerar korrekt innan ansvaret överlämnas. Den här säkerhetsmodellen stöder inte bara den förvalda startprocessen på Apple-enheter, utan även de olika lägen som finns för återställning och snabb uppdatering på Apple-enheter. Delkomponenter som Secure Enclave utför också en egen säker start för att ytterligare säkerställa att de endast startar känd, ofarlig kod från Apple. Uppdateringssystemet är utformat för att förhindra nedgraderingsangrepp så att enheter inte kan ändras tillbaka till en äldre version av operativsystemet (som angriparen känner till svagheterna i och kan hacka) i syfte att stjäla användardata.

Apple-enheter innehåller även skydd både vid start och vid användning så att integriteten kan upprätthållas under drift. Apple-utformade kretsar i iPhone, iPad, en Mac med Apple Silicon, Apple Watch, Apple TV och HomePod bildar en gemensam arkitektur för skydd av operativsystemets integritet. macOS har även en utökad och konfigurerbar uppsättning skyddsfunktioner som stöd för dess olika beräkningsmodeller samt funktioner som stöds på alla Mac-maskinvaruplattformar.

# Säker start

## Startprocess för iPhone- och iPad-enheter

Varje steg i startprocessen innehåller komponenter som är kryptografiskt signerade av Apple för att möjliggöra integritetskontroller så att startprocessen inte går vidare förrän varje tillförlitlighetssteg har verifierats. Bland dessa komponenter finns bootloader-komponenterna, kärnan, tillägg till kärnan och den fasta programvaran för mobilbasbandet. Den här säkra startsekvensen är utformad för att verifiera att den mest grundläggande programvaran inte har manipulerats.

När en iPhone- och iPad-enhet slås på kör approcessorn omedelbart kod från ett skrivskyddat startminne som kallas Boot ROM. Den här statiska koden, som kallas *betrodd rot för maskinvaran*, skapas när kretsen tillverkas och är implicit betrodd. Startminneskoden innehåller den publika nyckeln för Apples rotcertifikatutfärdare (CA) som används till att verifiera att iBoot-bootloadern är signerad av Apple innan den läses in. Det här är det första steget i tillförlitlighetskedjan där varje steg kontrollerar att nästa är signerat av Apple. När iBoot är klar med sina uppgifter verifierar och kör den iOS- eller iPadOS-kärnan. För enheter med en A9- eller tidigare A-serieprocessor läses ytterligare ett LLB-steg (Low-Level Bootloader) in och verifieras av startminnet (Boot ROM), och det läser i sin tur in och verifierar iBoot.

Om följande steg inte kan läsas in eller verifieras hanteras detta på olika sätt beroende på maskinvara:

- *Boot ROM kan inte läsa in LLB (äldre enheter):* DFU-läge (Device Firmware Upgrade)
- *LLB eller iBoot:* Återställningsläge

I båda fallen måste enheten vara ansluten till Finder (i macOS 10.15 eller senare) eller iTunes (macOS 10.14 eller tidigare) via USB och återställas till fabriksinställningarna.

BPR (Boot Progress Register) används av Secure Enclave för att begränsa tillgången till användardata i olika lägen och uppdateras innan enheten försätts i följande lägen:

- *DFU-läge:* Ställs in av Boot ROM på enheter med en Apple A12- eller senare SoC.
- *Återhämtningsläge:* Ställs in av iBoot på enheter med Apple A10-, S2- eller senare SoC.

På enheter med möjlighet till mobilanslutning utför ett mobilbasbands undersystem ytterligare en säker start med signerad programvara och nycklar som verifieras av basbandsprocessorn.

Secure Enclave utför också en säker startprocess som kontrollerar att dess egen programvara (sepOS) är verifierad och signerad av Apple.

## Minnessäker iBoot-implementering

I iOS 14 och iPadOS 14 eller senare har Apple ändrat den C-kompilatorverktygskedja som används till att bygga iBoot-bootloadern för att förbättra dess säkerhet. Den ändrade verktygskedjan implementerar kod som ska förhindra minnes- och typsäkerhetsproblem som är kända för C-program. Den förhindrar till exempel de flesta svagheterna i följande klasser:

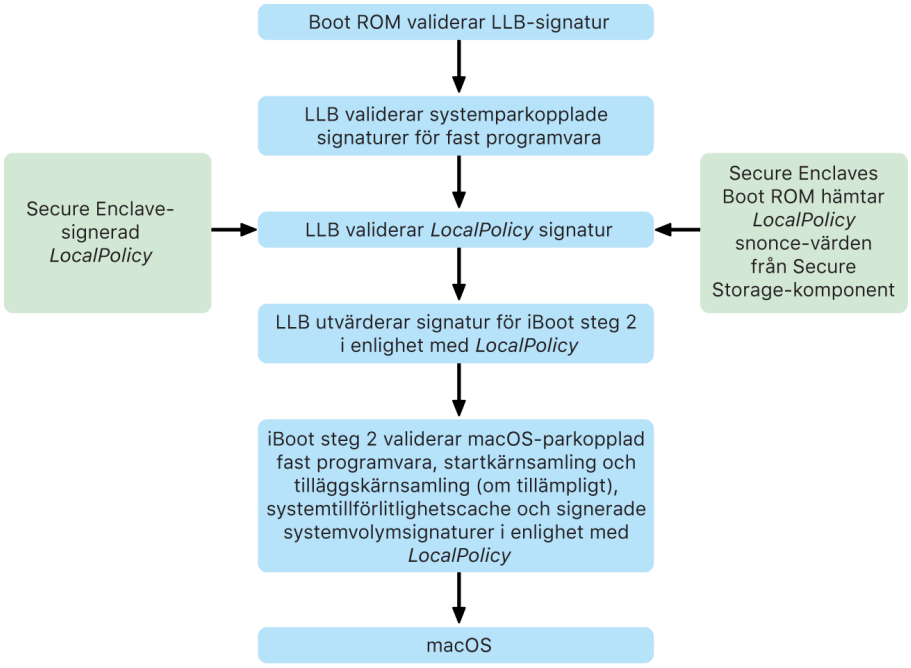
- Buffertöverflöden, genom att säkerställa att alla pekare bär gränsinformation som är verifierad vid åtkomst till minne
- Heap exploitation, genom att separera heap-data från dess metadata och korrekt detektera felhändelser som dubbelfribuggar
- Type confusion, genom att säkerställa att alla pekare bär runtime-type-information som är verifierad under pekar-cast-åtgärder
- Type confusion orsakad av UAF-fel (use-after-free), genom segregering av alla dynamiska minnestilldelningar efter statisk typ

Tekniken är tillgänglig på iPhone med A13 Bionic-kretsen eller senare och på iPad med A14 Bionic-krets eller senare.

## Mac-datorer med Apple Silicon

### Startprocessen för Mac-datorer med Apple Silicon

När en Mac med Apple Silicon slås på utförs en startprocess som liknar den för iPhone och iPad.




Kretsen kör kod från Boot ROM i det första steget i tillförlitlighetskedjan. Vid säker macOS-start på en Mac med Apple Silicon verifieras inte bara själva operativsystemets kod, utan även de säkerhetspolicyer och till och med de kärntillägg (stöds även om det inte rekommenderas) som konfigurerats av auktoriserade användare.

När LLB (Low Level Bootloader) startas verifierar det signaturerna och läser in systemparkopplad fast programvara för interna SoC-kärnor som styrenheterna för lagring, visning, systemhantering och Thunderbolt LLB ansvarar också för inläsning av LocalPolicy, vilket är en fil som signerats av Secure Enclave-processorn. LocalPolicy-filen beskriver den konfiguration som användaren har valt för säkerhetspolicyer vid systemstart och körning. LocalPolicy har samma datastrukturformat som alla andra startobjekt, men det signeras lokalt av en privat nyckel som endast är tillgänglig inom en viss dators Secure Enclave istället för att signeras av en central Apple-server (som programuppdateringar).

För att förhindra replay av tidigare LocalPolicy-filer måste LLB söka upp ett anti-replay-värde från den SSC (Secure Storage Component) som är kopplad till Secure Enclave. Den gör den genom att använda Secure Enclaves Boot ROM och säkerställer att anti-replay-värdet i LocalPolicy matchar anti-replay-värdet i SSC. Detta förhindrar att en gammal LocalPolicy-fil – som kan ha konfigurerats för en lägre säkerhet – återanvänds i systemet efter att säkerheten har uppgraderats. Resultatet blir att säker start på Mac-datorer med Apple Silicon inte bara utgör ett skydd mot nedgradering av operativsystemsversionen, utan även mot nedgraderingar av säkerhetspolicyer.

LocalPolicy-filen bekräftar om operativsystemet har konfigurerats för full, minskad eller tillåtande säkerhet.

- *Full säkerhet:* Systemet beter sig som iOS och iPadOS och tillåter endast start av programvara som har identifierats som den senast tillgängliga vid installationstillfället.
- *Minskad säkerhet:* LLB instrueras att lita på ”globala” signaturer som hör samman med operativsystemet. Detta tillåter systemet att köra äldre versioner av macOS. Eftersom äldre versioner av macOS oundvikligen har svagheter som inte är åtgärdade beskrivs detta säkerhetsläge som *Minskad* säkerhet. Detta är också den policynivå som krävs för att stöda start av kärntillägg.
- *Tillåtande säkerhet:* Systemet beter sig som Minskad säkerhet på så vis att det använder global signaturverifiering för iBoot och vidare, men det instruerar också iBoot att acceptera att vissa startobjekt är signerade av Secure Enclave med samma nyckel som används till att signera LocalPolicy. Med den här policynivån kan användare bygga, signera och starta egna anpassade XNU-kärnor.

Om LocalPolicy indikerar till LLB att det valda operativsystemet körs med full säkerhet utvärderar LLB den anpassade signaturen för iBoot. Om det körs med minskad eller tillåtande säkerhet utvärderas den globala signaturen. Eventuella fel vid signaturverifiering leder till att systemet startar i recoveryOS så att reparationsalternativ visas.

När LLB lämnar över till iBoot läser den in macOS-parkopplad fast programvara, t.ex. den för Secure Neural Engine, processorn som alltid är på och annan fast programvara. iBoot analyserar också information om LocalPolicy-filen som LLB har lämnat över. Om LocalPolicy indikerar att det ska finnas en AuxKC (Auxiliary Kernel Collection) söker iBoot efter den i filsystemet, verifierar att den är signerad av Secure Enclave med samma nyckel som LocalPolicy samt verifierar att dess hash matchar en hash som är lagrad i LocalPolicy. Om AuxKC verifieras placerar iBoot den i ett minne med startkärnsamlingen innan hela minnesregionen med startkärnsamlingen och AuxKC låses med SCIP (System Coprocessor Integrity Protection). Om policyn indikerar att en AuxKC ska finnas, men den inte kan hittas, fortsätter systemet att starta i macOS utan den. iBoot ansvarar också för att verifiera rothash för den signerade systemvolymen (SSV) för att kontrollera att filsystemet som kärnan kommer att länka in är fullständigt integritetsverifierat.

## Startlägen för Mac-datorer med Apple Silicon

Mac-datorer med Apple Silicon har de startlägen som beskrivs nedan.

| Läge | Tangentkombination | Beskrivning |
|---|---|---|
| macOS | Tryck på strömbrytaren när datorn är avstängd och **släpp** den sedan. | 1. Boot ROM lämnar över till LLB.<br>2. LLB läser in systemparkopplad fast programvara och LocalPolicy för det macOS som är valt.<br>3. LLB låser en indikation i BPR (Boot Progress Register) om att den startar i macOS och lämnar över till iBoot.<br>4. iBoot läser in den macOS-parkopplade fasta programvaran, den statiska tillförlitlighetscachen, enhetsträdet och startkärnsamlingen.<br>5. Om LocalPolicy tillåter det läser iBoot in AuxKC (Auxiliary Kernel Collection) med kärntillägg från tredje part.<br>6. Om LocalPolicy inte har avaktiverat det verifierar iBoot rotsignaturhashen för den signerade systemvolymen (SSV). |
| Parkopplat recoveryOS | Tryck på strömbrytaren när enheten är avstängd och **håll den intryckt**. | 1. Boot ROM lämnar över till LLB.<br>2. LLB läser in systemparkopplad fast programvara och LocalPolicy för recoveryOS.<br>3. LLB låser en indikation i BPR (Boot Progress Register) om att den startar i parkopplat recoveryOS och lämnar över till iBoot för parkopplat recoveryOS.<br>4. iBoot läser in den macOS-parkopplade fasta programvaran, tillförlitlighetscachen, enhetsträdet och startkärnsamlingen.<br>5. Om start i parkopplat recoveryOS misslyckas görs ett försök att starta i fallback recoveryOS. |
| Fallback recoveryOS | **Tryck på strömbrytaren två gånger och håll den intryckt** när enheten är avstängd. | 1. Boot ROM lämnar över till LLB.<br>2. LLB läser in systemparkopplad fast programvara och LocalPolicy för recoveryOS.<br>3. LLB låser en indikation i BPR (Boot Progress Register) om att den startar i parkopplat recoveryOS och lämnar över till iBoot för recoveryOS.<br>4. iBoot läser in den macOS-parkopplade fasta programvaran, tillförlitlighetscachen, enhetsträdet och startkärnsamlingen. |
| Säkert läge | Starta i recoveryOS enligt anvisningarna ovan. Håll sedan ned **skifttangenten** och markera startvolymen. | 1. Den startas i recoveryOS enligt beskrivningen ovan.<br>2. När skifttangenten hålls ned samtidigt som en volym markeras godkänner BootPicker-appen att den macOS-versionen startas som vanligt. En `nvram`-variabel som instruerar iBoot att inte läsa in AuxKC vid nästa start anges också.<br>3. Systemet startar om och startar med den valda volymen, men iBoot läser inte in AuxKC. |

## Begränsningar för parkopplat recoveryOS

I macOS 12.0.1 eller senare installerar varje ny macOS-installation även en parkopplad version av recoveryOS i den motsvarande APFS-volymgruppen. Den här metoden är välbekant för användare med Intel-baserade Mac-datorer, men på en Mac med Apple Silicon bidrar den med ytterligare garantier för säkerhet och kompatibilitet. Eftersom alla macOS-installationer nu har ett dedikerat parkopplat recoveryOS säkerställer det att endast detta dedikerade parkopplade recoveryOS kan utföra åtgärder som minskar säkerheten. Det bidrar till att skydda installationer av nyare macOS-versioner så att de inte kan manipuleras från äldre macOS-versioner och tvärtom.

Parkopplingsbegränsningarna genomdrivs enligt följande:

- Alla installationer av macOS 11 parkopplas med recoveryOS. Ifall en macOS 11-installation är det förvalda startsystemet startar du recoveryOS genom att hålla ned strömbrytaren när du startar en Mac med Apple Silicon. recoveryOS kan nedgradera säkerhetsinställningarna för valfria macOS 11-installationer, men inga installationer av macOS 12.0.1.
- Ifall en macOS 12.0.1-installation eller senare är det förvalda startsystemet startar du dess parkopplade recoveryOS genom att hålla ned strömbrytaren när du startar datorn. Ett parkopplat recoveryOS kan nedgradera säkerhetsinställningarna för den parkopplade macOS-installationen, men inga andra macOS-installationer.

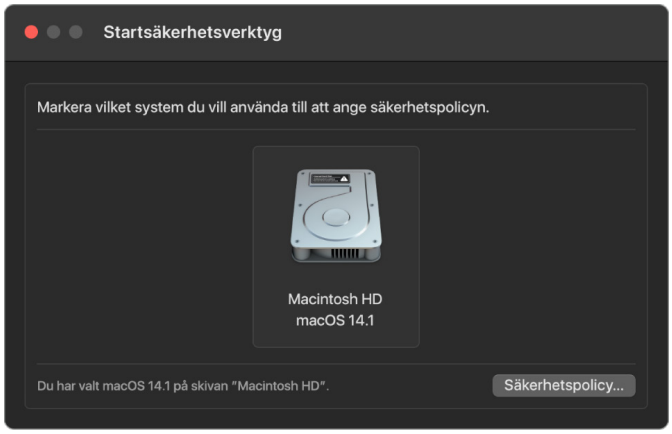
Om du vill starta ett parkopplat recoveryOS för valfri macOS-installation måste den installationen vara vald som förval i Allmänt > Startskiva i Systeminställningar (macOS 13 eller senare), Startskiva i Systeminställningar (macOS 12 eller tidigare) eller genom att starta ett valfritt recoveryOS och hålla ned alternativtangenten medan du väljer en volym.

*Obs!* Fallback recoveryOS kan inte utföra nedgraderingar för macOS-installationer.

## Kontroll av säkerhetspolicyn för Startskiva för Mac-datorer med Apple Silicon

### Översikt

Till skillnad från säkerhetspolicyer på Intel-baserade Mac-datorer kan Mac-datorer med Apple Silicon ha olika säkerhetspolicyer för varje installerat operativsystem. Detta innebär att det finns stöd för flera installerade macOS-instanser med olika versioner och säkerhetspolicyer på samma Mac. Därför har en *operativsystemsväljare* lagts till i Startsäkerhetsverktyg.

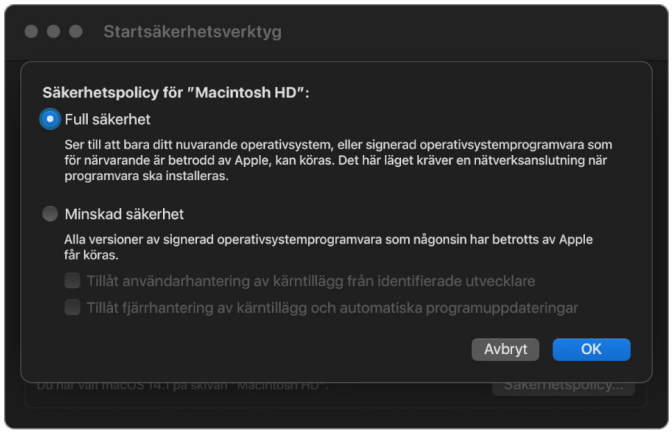


På Mac-datorer med Apple Silicon indikerar Systemsäkerhetsverktyg den allmänna, användarkonfigurerade säkerhetsstatusen för macOS, exempelvis start av ett kärntillägg eller konfigurationen av systemintegritetsskydd (SIP). Om en ändring av en säkerhetsinställning leder till kraftig försämring av säkerheten, eller gör systemet enklare att manipulera, måste användaren starta i recoveryOS genom att hålla in strömbrytaren (så att signalen inte kan utlösas av ett sabotageprogram utan endast av en människa med fysisk åtkomst till datorn) när ändringen ska utföras. Därför kräver Mac-datorer med Apple Silicon inte (och stöder inte) lösenord för fast programvara – alla kritiska ändringar skyddas redan av användarauktorisering. Mer information om systemintegritetsskyddet finns i Systemintegritetsskydd.

Full säkerhet och minskad säkerhet kan ställas in genom att använda Startsäkerhetsverktyg från recoveryOS. Tillåtande säkerhet är däremot endast tillgängligt via kommandoradsverktyg för användare som accepterar risken med att göra sin Mac-dator mycket mindre säker.

### Policyn Full säkerhet

Förvalet är Full säkerhet och det beter sig som iOS och iPadOS. Istället för att använda den globala signaturen som följer med programvaran när programvara har hämtats och är klar för installation kontaktar macOS samma Apple-signeringsserver som används för iOS och iPadOS och begär en ny, unik signatur. En unik signatur skapas när ett ECID (Exclusive Chip Identification) – ett unikt ID som i det här fallet är specifikt för Apple-processorn – ingår som en del av en signeringsbegäran. Den signatur som levereras av signeringsservern är sedan unik och kan endast användas av just den aktuella Apple-processorn. När policyn Full säkerhet används säkerställer Boot ROM och LLB att en levererad signatur inte bara är signerad av Apple, utan också signerad för den aktuella Mac-datorn, vilket i praktiken binder den versionen av macOS till just den datorn.

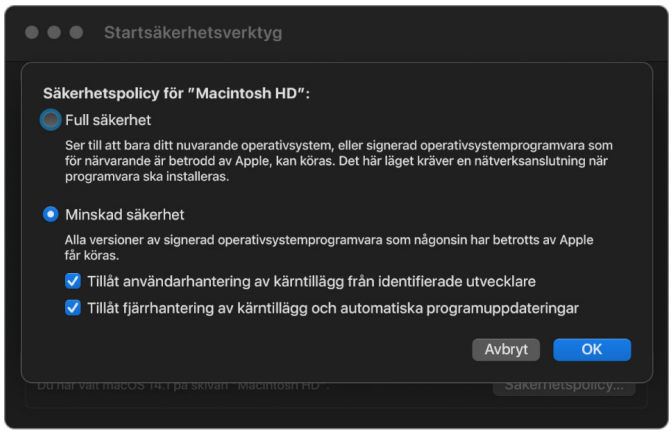



Att använda en onlinesigneringsserver ger också bättre skydd mot nedgraderingsangrepp än typiska metoder för global signering. I ett globalt signeringssystem kan säkerhetsepoken ha uppdaterats flera gånger, men ett system som aldrig har sett den senaste fasta programvaran kommer inte att känna till detta. En dator som t.ex. tror att den befinner sig i säkerhetsepok 1 kommer att acceptera programvara från säkerhetsepok 2, även om den senaste säkerhetsepoken är 5. Med ett onlinesigneringssystem av den typ som används för Apple-kretsar kan signeringsservern vägra att skapa signaturer för programvara som inte tillhör den senaste säkerhetsepoken.

Om angripare upptäcker ett säkerhetshål efter att en säkerhetsepok har ändrats kan de dessutom inte helt enkelt hämta den sårbara programvaran från en tidigare epok från system A och använda det på system B för att kunna angripa det. Om den sårbara programvaran från en äldre epok är personligt anpassad till system A kan denna inte överföras och heller inte användas för ett angrepp på system B. Alla dessa mekanismer ger tillsammans en bättre garanti för att angripare inte aktivt ska kunna installera sårbar programvara på en Mac för att komma runt det skydd som den senaste programvaran tillhandahåller. En användare som har tillgång till användarnamn och lösenord för en administratör på datorn kan dock alltid välja den säkerhetspolicy som fungerar bäst för den aktuella användningssituationen.

### Policyn Minskad säkerhet

Minskad säkerhet liknar beteendet för medelsäkerhet på Intel-baserade Mac-datorer med T2-krets där en leverantör (i det här fallet Apple) genererar en digital signatur för koden för att säkerställa att den kom från leverantören. Den här designen förhindrar att angripare matar in osignerad kod. Apple kallar den här signaturen för en ”global” signatur eftersom den kan användas på valfri Mac och under obegränsad tid för en Mac där policyn Minskad säkerhet har ställts in. Minskad säkerhet tillhandahåller i sig självt inte något skydd mot nedgraderingsangrepp (även om obehöriga operativsystemsändringar kan leda till att användardata blir otillgänglig). Mer information finns i Kärntillägg på Mac-datorer med Apple Silicon.

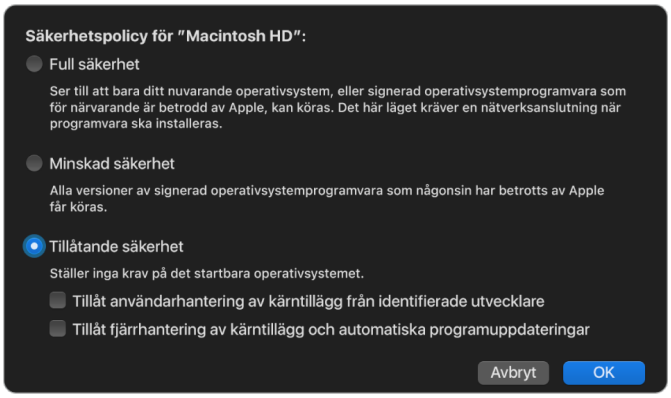



Utöver att göra det möjligt för användare att köra äldre versioner av macOS krävs minskad säkerhet för andra åtgärder som utsätter användarens systemsäkerhet för risker, exempelvis att introducera kärntillägg från tredje part. Kärntillägg har samma privilegier som kärnan, och därför kan eventuella svagheter i kärntillägg från tredje part leda till att hela operativsystemet äventyras. Det är därför utvecklare starkt rekommenderas att övergå till systemtillägg innan stöd för kärntillägg tas bort från macOS för framtida Mac-datorer med Apple Silicon. Även om kärntillägg från tredje part aktiveras kan de inte läsas in i kärnan på begäran. Istället har kärntilläggen slagits samman i en AuxKC (Auxiliary Kernel Collection) – vars hashvärde lagras i LocalPolicy – och kräver därför en omstart. Mer information om AuxKC-generering finns i Säker utökning av kärnan i macOS.

### Policyn Tillåtande säkerhet

Tillåtande säkerhet kan användas av användare som accepterar risken att försätta sina Mac-datorer i ett mycket mer oskyddat läge. Det här läget skiljer sig från läget Ingen säkerhet på Intel-baserade Mac-datorer med T2-krets. Med tillåtande säkerhet utförs signaturverifiering fortfarande längs hela den säkra startkedjan, men att ställa in policyn till Tillåtande signalerar till iBoot att det ska acceptera lokala startobjekt som är Secure Enclave-signerade, exempelvis en användargenererad startkärnsamling byggd från en anpassad XNU-kärna. På så vis tillhandahåller tillåtande säkerhet också en arkitekturmiljö för körning av en slumpmässig ”fullständigt obetrodd operativsystemkärna”. När en anpassad startkärnsamling eller ett helt betrott operativsystem läses in i systemet slutar vissa avkrypteringsnycklar att vara tillgängliga. Det är utformat för att förhindra att helt obetrodda operativsystem får tillgång till data från betrodda operativsystem.

**Viktigt:** Apple tillhandahåller inte och stöder inte anpassade XNU-kärnor.




Tillåtande säkerhet skiljer sig också från Ingen säkerhet på Intel-baserade Mac-datorer med T2-krets på ett annat sätt: Det är en förutsättning för vissa säkerhetsnedgraderingar som förut har kunnat styras oberoende. Framförallt måste användare som avaktiverar systemintegritetsskydd (SIP) på Mac-datorer med Apple Silicon godkänna att de försätter systemet i Tillåtande säkerhet. Detta krävs eftersom avaktivering av systemintegritetsskyddet alltid har försatt systemet i ett läge som gör det mycket enklare att manipulera kärnan. Framförallt innebär en avaktivering av systemintegritetsskyddet på Mac-datorer med Apple Silicon att kravet på signering av extra kärntillägg vid AuxKC-generering avaktiveras, vilket gör det möjligt för slumpmässiga kärntillägg att läsas in i kärnans minne. En annan förbättring av systemintegritetsskyddet som har skett på Mac-datorer med Apple Silicon är att policylagringen har flyttats ut från NVRAM och in i LocalPolicy. Därför kräver avaktivering av systemintegritetsskyddet numera autentisering av en användare som har tillgång till signeringsnyckeln för LocalPolicy, och att det sker via recoveryOS som nås genom att användaren håller in strömbrytaren. Det gör det betydligt svårare för en angripare som endast attackerar programvara, och till och med en fysiskt närvarande angripare, att avaktivera SIP.

Det går inte att nedgradera till Tillåtande säkerhet via appen Startsäkerhetsverktyg. Användare kan nedgradera endast genom att köra kommandoradsverktyg via Terminal i recoveryOS, exempelvis `csrutil` (för att avaktivera SIP). När en nedgradering har skett visas det i Startsäkerhetsverktyg, och en användare kan därför enkelt ställa in ett starkare säkerhetsläge.

*Obs!* Mac-datorer med Apple Silicon kräver inte och stöder inte heller en specifik mediestartpolicy eftersom alla starter rent tekniskt utförs lokalt. Om en användare väljer att starta från ett externt medium måste den operativsystemsversionen först anpassas genom användning av en autentiserad omstart från recoveryOS. Denna omstart skapar en LocalPolicy-fil på den interna enheten. Filen används till att utföra en betrodd start från det operativsystem som lagras på det externa mediet. Detta innebär att konfigurationen av start från ett externt medium alltid uttryckligen aktiveras per operativsystem och redan kräver användarauktorisering, så ingen extra säkerhetskonfiguration krävs.

## Skapa och hantera signeringsnyckeln för LocalPolicy

### Skapande

När macOS installeras första gången i fabriken, eller när en kopplad raderingsinstallation utförs, kör Mac-datorn kod från en tillfällig återställnings-RAM-skiva för att initiera standardläget. Under den här processen skapar återställningsmiljön ett nytt publikt och privat nyckelpar som förvaras i Secure Enclave. Den privata nyckeln kallas *Owner Identity Key (OIK)*. Om det redan finns en OIK förstörs den som en del av den här processen. Återställningsmiljön initierar också den nyckel som används för Aktiveringslås, kallad *User Identity Key (UIK)*. En del av den här processen som är unik för Mac-datorer med Apple Silicon är att när UIK-certifiering begärs för Aktiveringslås inkluderas en uppsättning begärda begränsningar som ska genomdrivas vid valideringstillfället på LocalPolicy. Om enheten inte kan få en UIK-certifiering för Aktiveringslås (t.ex. om enheten vid det aktuella tillfället är kopplad till ett Hitta min Mac-konto och är rapporterad som förlorad) kan den inte fortsätta att skapa en Local Policy. Om ett *ucrt-certifikat (User identity Certificate)* skapas för en enhet innehåller certifikatet serverskapade policybegränsningar och användarbegärda policybegränsningar i ett X.509 v3-tillägg.

När ett Aktiveringslås/`ucrt`-certifikat hämtats korrekt lagras det i en databas på serversidan och returneras även till enheten. När enheten har ett ucrt-certifikat skickas en certifieringsbegäran för den publika nyckeln som motsvarar OIK till *BAA*-servern (Basic Attestation Authority). BAA verifierar OIK-certifieringens begäran med den publika nyckeln från det ucrt som lagras i den databas som är tillgänglig för BAA. Om BAA inte kan verifiera certifieringen certifierar BAA den publika nyckeln och returnerar *OIC (Owner Identity Certificate)* som signerats av BAA och innehåller de begränsningar som är lagrade i ucrt. OIC skickas tillbaka till Secure Enclave. Därefter bifogar Secure Enclave OIC:n med Image4-filen varje gång den signerar en ny LocalPolicy. Förtroendet för BAA-rotcertifikatet är inbyggt i LLB, vilket leder till att LLB litar på OIC, vilket i sin tur leder till att det litar på den allmänna LocalPolicy-signaturen.

### RemotePolicy-begränsningar

Alla Image4-filer, inte bara Local Policy-filer, innehåller begränsningar för Image4-manifestutvärdering. Dessa begränsningar är kodade med speciella OID:er i bladcertifikatet. Image4-verifieringsbiblioteket kontrollerar det speciella certifikatbegränsnings-OID:t från ett certifikat under signaturutvärderingen och utvärderar sedan mekaniskt de begränsningar som anges i det. Begränsningarna är av typen:

- X måste finnas
- X får inte finnas
- X måste ha ett specifikt värde

Så för exempelvis ”personligt anpassade” signaturer innehåller certifikatbegränsningarna ”ECID måste finnas”, och för ”globala” signaturer innehåller de ”ECID får inte finnas”. Dessa begränsningar är utformade på ett sätt som säkerställer att alla Image4-filer som signerats av en given nyckel måste följa vissa krav för att undvika felaktigt signerad Image4-manifestgenerering.

När det gäller LocalPolicy-filer kallas dessa Image4-certifikatbegränsningar för *RemotePolicy*. Det kan finnas olika RemotePolicy-filer för olika startmiljöers LocalPolicy-filer. RemotePolicy används till att begränsa LocalPolicy-filen för recoveryOS så att recoveryOS alltid beter sig som om det startar med Full säkerhet. Detta ökar förtroendet för integriteten hos recoveryOS-startmiljön som en plats varifrån policy kan ändras. RemotePolicy begränsar LocalPolicy-filen så att den innehåller ECID för den dator som LocalPolicy genererades på och det specifika `rpnh`-värde (Remote Policy Nonce Hash) som lagras i SSC på den datorn. Detta `rpnh`-värde, och därmed RemotePolicy, ändras endast när åtgärder vidtas för Hitta min Mac och Aktiveringslås, exempelvis registrering, avregistrering, fjärrlås och fjärradering. Remote Policy-begränsningar bestäms och specificeras vid tidpunkten för UIK-certifieringen och signeras in i det utfärdade ucrt-certifikatet. Vissa Remote Policy-begränsningar, som ECID, ChipID och BoardID, bestäms av servern. Det ska förhindra att en enhet signerar LocalPolicy-filer åt en annan enhet. Andra Remote Policy-begränsningar kan specificeras av enheten för att förhindra säkerhetssnedgraderingar av Local Policy utan tillhandahållande av både den lokala autentisering som krävs för åtkomst till aktuell OIK och fjärrautentisering av kontot till vilket enheten är aktiveringslåst.

## Innehåll i en LocalPolicy-fil för Mac-datorer med Apple Silicon

LocalPolicy är en Image4-fil som är signerad av Secure Enclave. Image4 är ett ASN.1 DER-kodat datastrukturformat som används till att beskriva information om säkra startkedjeobjekt på Apple-plattformar. I en Image4-baserad säker startmodell begärs säkerhetspolicyer vid tidpunkten för programinstallation som initierats av en signeringsbegäran till en central Apple-signeringsserver. Om policyn var godtagbar returnerar signeringsservern en signerad Image4-fil med en mängd olika fyrteckenkodsekvenser (4-character-codes, 4CC). Dessa signerade Image4-filer och 4CC:er utvärderas vid start av programvara som Boot ROM eller LLB.

### Överlämnande av ägarskap mellan operativsystem

Tillgång till OIK (Owner Identity Key) kallas för ”ägarskap”. Ägarskap krävs för att tillåta användare att omsignera LocalPolicy när policy- eller programvaruändringar har gjorts. OIK skyddas med samma nyckelhierarki som beskrivs i SKP (Sealed Key Protection), där OIK skyddas av samma KEK (Key Encryption Key) som VEK (Volume Encryption Key). Det innebär att det normalt skyddas av både användarlösenord och mätvärden från operativsystemet och policyn. Det finns endast en enda OIK för alla operativsystem på datorn. Vid installation av ett andra operativsystem krävs därför uttryckligt godkännande från användarna i det första operativsystemet för att överlämna ägarskapet till användarna i det andra operativsystemet. Det finns dock ännu inga användare i det andra operativsystemet när installeraren körs från det första operativsystemet. Användare genereras normalt inte i operativsystem förrän operativsystemet startas och inställningsassistenten körs. Därför krävs två nya åtgärder vid installation av ett andra operativsystem på Mac-datorer med Apple Silicon:

- Skapa en LocalPolicy för det andra operativsystemet.
- Förbereda en ”installationsanvändare” för överlämnande av ägarskapet.

När en inställningsassistent och en målinstallation för en sekundär tom volym körs får användaren frågan om en användare från den aktuella volymen ska kopieras och skapas som den första användaren på den andra volymen. Om användaren svarar ja blir den ”installationsanvändare” som skapas i praktiken en KEK som härleds från den valda användarens lösenord och maskinvarunycklar, och som sedan används till att kryptera OIK när den överlämnas till det andra operativsystemet. Inställningsassistenten för det andra operativsystemet ber sedan om den användarens lösenord för att tillåta användaren tillgång till OIK i Secure Enclave för det nya operativsystemet. Om användare väljer att inte kopiera en användare skapas installationsanvändaren på samma sätt, men ett tomt lösenord används istället för en användares lösenord. Det här andra flödet finns för användning vid vissa systemadministrationsscenarier. Användare som vill installera med flera volymer och som vill utföra överlämnade av ägarskap så säkert som möjligt bör dock alltid välja att kopiera en användare från det första operativsystemet till det andra operativsystemet.

### LocalPolicy på Mac-datorer med Apple Silicon

För Mac-datorer med Apple Silicon har styrningen av lokala säkerhetspolicyer delegerats till en app som körs i Secure Enclave. Denna programvara kan utnyttja användarens behörighet och startläget för den primära processorn till att bestämma vem som kan ändra säkerhetspolicyn och från vilken startmiljö. Detta hjälper till att förhindra att sabotageprogram använder styrfunktionerna för säkerhetspolicyn mot användaren genom att nedgradera dem för att få fler behörigheter.

### Manifestegenskaper för LocalPolicy

LocalPolicy-filen innehåller en del arkitektoniska 4CC:er som finns i nästan alla Image4-filer, som ett kort- eller modell-ID (BORD) som indikerar en viss Apple-krets (CHIP) eller ECID (Exclusive Chip Identification). 4CC:erna nedan fokuserar dock endast på de säkerhetspolicyer som användare kan konfigurera.

*Obs!* Apple använder termen *Paired One True recoveryOS (1TR)* för att ange en start i parkopplat recoveryOS genom att strömbrytaren fysiskt hålls intryckt. Detta skiljer sig från en normal recoveryOS-start som sker genom användning av NVRAM, genom att snabbt trycka två gånger och hålla eller om fel uppstår under start. Att knappen trycks in fysiskt på ett särskilt sätt ökar förtroendet för att startmiljön inte kan nås av en programvaruangripare som har brutit sig in i macOS.

### LocalPolicy Nonce Hash (lpnh)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* Detta `lpnh`-värde används för anti-replay av LocalPolicy. Detta är en SHA384-hash av LocalPolicy Nonce (LPN) som lagras i SSC och är åtkomlig med Secure Enclaves Boot ROM eller Secure Enclave. Det råa anti-replay-värdet är aldrig synligt för approcessorn, endast för sepOS. En angripare som vill övertyga LLB om att en föregående LocalPolicy de har fångat in är giltig skulle behöva placera ett värde i SSC som hashas till samma `lpnh`-värde som hittas i den LocalPolicy som de vill återanvända. Normalt finns det en enda giltig LPN i systemet. Undantaget är under pågående programuppdatering när två LPN är giltiga samtidigt, detta för att tillåta möjligheten att återgå till start av den gamla programvaran om ett uppdateringsfel inträffar. När en LocalPolicy för något operativsystem ändras blir alla policyer omsignerade med det nya *lpnh*-värde som motsvarar den nya LPN som finns i SSC. Ändringen görs när användaren ändrar säkerhetsinställningar eller skapar nya operativsystem med en ny LocalPolicy för vardera system.

### Remote Policy Nonce Hash (rpnh)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* Detta `rpnh`-värde beter sig på samma sätt som `lpnh` men uppdateras endast när Remote Policy uppdateras, exempelvis när status ändras för Hitta-registreringen. Ändringen görs när användaren ändrar status för Hitta på datorn.

### recoveryOS Nonce Hash (ronh)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* Detta `ronh`-värde beter sig på samma sätt som lpnh men finns endast i LocalPolicy för systemets recoveryOS. Det uppdateras när systemets recoveryOS uppdateras, exempelvis vid programuppdateringar. Ett separat anti-replay-värde från `lpnh` och `rpnh` används så att befintliga operativsystem avaktiveras (genom att ta bort deras LPN och RPN från SSC) när en enhet avaktiveras via Hitta, samtidigt som systemets recoveryOS fortfarande går att starta. På så vis kan operativsystemet återaktiveras när systemägaren bevisar att han eller hon har kontroll över systemet genom att ange sitt iCloud-lösenord för Hitta-kontot. Ändringen görs när en användare uppdaterar systemets recoveryOS eller skapar nya operativsystem.

### Next Stage Image4 Manifest Hash (nsih)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* Detta *nsih*-fält representerar en SHA384-hash av Image4-manifestdatastrukturen som beskriver det startade macOS-systemet. macOS Image4-manifestet innehåller åtgärder för alla startobjekt som iBoot, den statiska tillförlitlighetscachen, enhetsträd, startkärnsamling och rothash för den signerade systemvolymen (SSV). När LLB instrueras att starta ett givet macOS är det meningen att det ska säkerställa att hashen för macOS Image4-manifestet som är bifogat till iBoot matchar informationen i fältet `nsih` i LocalPolicy. På så vis bekräftar `nsih` användarens avsikt gällande vilket operativsystem användaren har skapat en LocalPolicy för. Användarna ändrar implicit `nsih`-värdet när de utför en programuppdatering.

### Hash för Cryptex1 Image4-manifest (spih)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* `spih`-fältet representerar en SHA384-hash av Cryptex1 Image4-manifestdatastrukturen. Cryptex1 Image4-manifestet innehåller värden i dess cryptexar, deras filsystemssigill och deras associerade tillförlitlighetscache. När macOS startar säkerställer XNU-kärnan och Page Protection Layer att hashen av Cryptex1 Image4-manifestet matchar det som iBoot publicerade från `spih`-fältet i LocalPolicy. Användarna ändrar implicit `spih`-värdet när de installerar ett snabbt säkerhetssvar eller utför en programuppdatering. Hashen för Cryptex1 Image4-manifestet kan uppdateras oberoende av Next Stage Image4-manifesthashen.

### Cryptex1 Generation (stng)

- *Typ:* 64-bitars osignerat heltal
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* `stng`-fältet är ett räknarvärde som representerar när Cryptex1 Image4-manifesthashen senast uppdaterades i en LocalPolicy. Det tillhandahåller ett anti-replay-värde som ersätter `lpnh` under Page Protection Layers utvärdering av den lokala policyn för användning av Incoming Cryptex. Användarna ökar `stng`-värdet implicit när de installerar ett snabbt säkerhetssvar eller en programuppdatering.

### Auxiliary Kernel Collection (AuxKC) Policy Hash (auxp)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* macOS
- *Beskrivning:* Detta `auxp` är en SHA384-hash för den användarauktoriserade policyn för kärntillägg (user-authorized kext list, UAKL). Den används vid tidpunkten för AuxKC-generering för att säkerställa att endast användarauktoriserade kärntillägg inkluderas i AuxKC. `smb2` är ett krav för att ställa in detta fält. Användarna ändrar implicit `auxp`-värdet när de ändrar UAKL genom att godkänna ett kärntillägg från Integritet och säkerhet i Systeminställningar (macOS 13 eller senare) eller inställningspanelen Säkerhet och integritet i Systeminställningar (macOS 12 eller tidigare).

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

- *Typ:* OctetString (48)
- *Muterbara miljöer:* macOS
- *Beskrivning:* När systemet har verifierat att UAKL-hashen matchar med informationen i fältet `auxp` i LocalPolicy begär det att AuxKC ska signeras av den Secure Enclave-processorapp som ansvarar för LocalPolicy-signering. Därefter placeras en SHA384-hash för AuxKC Image4-manifestsignaturen i LocalPolicy för att undvika risken att tidigare signerade AuxKC:er mixas och matchas till ett operativsystem vid start. Om iBoot hittar fältet `auxi` i LocalPolicy försöker det läsa in AuxKC från lagringsutrymmet och validera dess signatur. Det verifierar även att hashen för det Image4-manifest som är bifogat till AuxKC matchar det värde som finns i fältet `auxi`. Om AuxKC av någon anledning inte läses in korrekt fortsätter systemet att starta utan det här startobjektet, och därför utan att några kärntillägg från tredje part blir inlästa. Fältet `auxp` är en förutsättning för att ställa in fältet auxi i LocalPolicy. Användarna ändrar implicit `auxi`-värdet när de ändrar UAKL genom att godkänna ett kärntillägg från Integritet och säkerhet i Systeminställningar (macOS 13 eller senare) eller inställningspanelen Säkerhet och integritet i Systeminställningar (macOS 12 eller tidigare).

### Auxiliary Kernel Collection (AuxKC) Receipt Hash (auxr)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* macOS
- *Beskrivning:* `auxr` är en SHA384-hash för AuxKC-kvittot som indikerar den exakta uppsättning kärntillägg som ingår i AuxKC. AuxKC-kvittot kan vara en delmängd av UAKL eftersom kärntillägg kan exkluderas från AuxKC, även om de auktoriserats av användaren, om de är kända för att användas för angrepp. Dessutom kan vissa kärntillägg som kan användas till att bryta gränsen mellan användare och kärna leda till minskad funktionalitet, exempelvis att det inte går att använda Apple Pay eller spela upp 4K- och HDR-innehåll. Användare som vill ha de här funktionerna väljer en mer restriktiv AuxKC-inkludering. Fältet `auxp` är en förutsättning för att ställa in fältet `auxr` i LocalPolicy. Användarna ändrar implicit `auxr`-värdet när de bygger en ny AuxKC från Integritet och säkerhet i Systeminställningar (macOS 13 eller senare) eller inställningspanelen Säkerhet och integritet i Systeminställningar (macOS 12 eller tidigare).

### Hash för CustomOS Image4-manifestet (coih)

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR
- *Beskrivning:* `coih` är en SHA384-hash i CustomOS Image4-manifestet. Nyttolasten för det manifestet används av iBoot (i stället för XNU-kärnan) till att överföra styrning. Användare ändrar implicit `coih`-värdet när de använder kommandoradsverktyget `kmutil configure-boot` i 1TR.

**APFS volume group UUID (vuid)**

- *Typ:* OctetString (16)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* `vuid` indikerar den volymgrupp kärnan ska använda som rot. Det här fältet är primärt upplysande och används inte för säkerhetsbegränsningar. Detta `vuid` ställs in implicit av användaren när en ny operativsystemsinstallation skapas.

**Key encryption key (KEK) Group UUID (kuid)**

- *Typ:* OctetString (16)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* `kuid` indikerar den volym som startas. KEK har typiskt använts för dataskydd. För varje LocalPolicy används den till att skydda LocalPolicy-signeringsnyckeln. `kuid` ställs in implicit av användaren när en ny operativsystemsinstallation skapas.

**Paired recoveryOS Trusted Boot Policy Measurement (prot)**

- *Typ:* OctetString (48)
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* Ett parkopplat recoveryOS-TBPM (Trusted Boot Policy Measurement) är en särskild iterativ SHA384-hashberäkning över Image4-manifestet för en LocalPolicy-fil, exklusive anti-replay-värden, för att ge ett konsekvent värde över tid (eftersom anti-replay-värden som `lpnh` uppdateras ofta). Fältet `prot`, som bara finns i vardera LocalPolicy-fil för macOS, tillhandahåller en parkoppling för att indikera den LocalPolicy för recoveryOS som motsvarar LocalPolicy för macOS.

**Has Secure Enclave Signed recoveryOS LocalPolicy (hrlp)**

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* `hrlp`-värdet indikerar om `prot`-värdet ovan är mätvärdet för en Secure Enclave-signerad LocalPolicy för recoveryOS eller inte. Om inte signeras LocalPolicy för recoveryOS av Apples onlinesigneringsserver som signerar sådant som exempelvis macOS Image4-filer.

**Local Operating System Version (love)**

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR, recoveryOS, macOS
- *Beskrivning:* `love` indikerar den OS-version som LocalPolicy är skapad för. Versionen hämtas från manifestet för nästa steg under skapandet av LocalPolicy och används till att genomdriva begränsningar av recoveryOS-parkoppling.

**Secure Multi-Boot (smb0)**

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR, recoveryOS
- *Beskrivning:* Om `smb0` är befintlig och sann tillåter LLB global signering av Image4-manifestet för nästa steg istället för att kräva en personligt anpassad signatur. Användarna kan ändra detta fält med Startsäkerhetsverktyg eller `bputil` för att nedgradera till Minskad säkerhet.

### Secure Multi-Boot (smb1)

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR
- *Beskrivning:* Om `smb1` är befintlig och sann tillåter iBoot att objekt som en anpassad startkärnsamling Secure Enclave-signeras med samma nyckel som LocalPolicy. Närvaron av `smb0` är ett krav för närvaron av `smb1`. Användarna kan ändra detta fält med kommandoradsverktyg som `csrutil` eller `bputil` för att nedgradera till Tillåtande säkerhet.

### Secure Multi-Boot (smb2)

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR
- *Beskrivning:* Om `smb2` är befintlig och sann tillåter iBoot att AuxKC Secure Enclave-signeras med samma nyckel som LocalPolicy. Närvaron av `smb0` är ett krav för närvaron av `smb2`. Användarna kan ändra detta fält med Startsäkerhetsverktyg eller `bputil` för att nedgradera till Minskad säkerhet och göra det möjligt att använda kärntillägg från tredje part.

### Secure Multi-Boot (smb3)

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR
- *Beskrivning:* Om `smb3` är befintlig och sann har en användare vid enheten valt att låta MDM (Mobile Device Management) styra systemet. Närvaron av det här fältet får den Secure Enclave-processorapp som styr LocalPolicy att acceptera MDM-autentisering istället för att kräva autentisering av en lokal användare. Användarna kan ändra det här fältet med Startsäkerhetsverktyg eller `bputil` för att möjliggöra hanterad styrning av kärntillägg från tredje part och programuppdateringar. (I macOS 11.2 och senare kan MDM också starta en uppdatering till den senaste macOS-versionen om säkerhetsläget är Full säkerhet.)

### Secure Multi-Boot (smb4)

- *Typ:* Booleskt
- *Muterbara miljöer:* macOS
- *Beskrivning:* Om `smb4` är befintlig och sann har enheten anslutit till MDM-styrning av operativsystemet via Apple School Manager, Apple Business Manager eller Apple Business Essentials. Närvaron av det här fältet får den Secure Enclave-app som styr LocalPolicy att acceptera MDM-autentisering istället för att kräva autentisering av en lokal användare. Detta fält ändras av MDM-lösningen när den upptäcker att enhetens serienummer finns i någon av de tre tjänsterna.

### Systemintegritetsskydd (sip0)

- *Typ:* 64-bitars osignerat heltal
- *Muterbara miljöer:* 1TR
- *Beskrivning:* `sip0` innehåller de befintliga SIP-policybitar (systemintegritetsskydd) som tidigare lagrats i NVRAM. Nya SIP-policybitar läggs till här (istället för att använda LocalPolicy-fält som nedan) om de endast används i macOS och inte används av LLB. Användarna kan ändra detta fält genom användning av `csrutil` från 1TR till att avaktivera SIP och nedgradera till tillåtande säkerhet.

### Systemintegritetsskydd (sip1)

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR
- *Beskrivning:* Om `sip1` är befintlig och sann tillåter iBoot fel vid verifiering av volymrothashen för SSV-volymen. Användarna kan ändra detta fält med `csrutil` eller `bputil` från 1TR.

### Systemintegritetsskydd (sip2)

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR
- *Beskrivning:* Om sip2 är befintlig och sann låser inte iBoot maskinvaruregistret *CTRR (Configurable Text Read-only Region)* som markerar kärnminnet som skrivskyddat. Användarna kan ändra detta fält med `csrutil` eller `bputil` från 1TR.

### Systemintegritetsskydd (sip3)

- *Typ:* Booleskt
- *Muterbara miljöer:* 1TR
- *Beskrivning:* Om `sip3` är befintlig och sann tvingar iBoot inte igenom användningen av dess inbyggda tillståndslista för NVRAM-variabeln i boot-args som annars skulle filtrera de alternativ som skickas till kärnan. Användarna kan ändra detta fält med `csrutil` eller `bputil` från 1TR.

### Certifikat och RemotePolicy

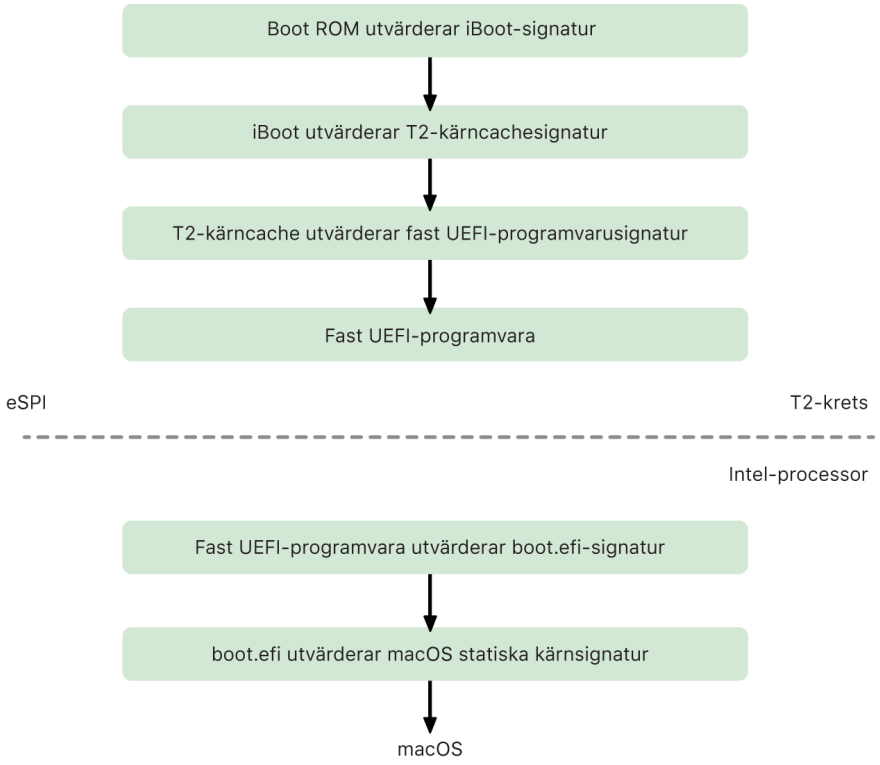
Enligt vad som beskrivs i Skapa och hantera signeringsnyckeln för LocalPolicy innehåller LocalPolicy Image4-filen också ett OIC (Owner Identity Certificate) och en inbäddad RemotePolicy.

# Intel-baserade Mac-datorer

## Startprocessen för Intel-baserade Mac-datorer

### Intel-baserade Mac-datorer med Apple T2-säkerhetskrets

När en Intel-baserad Mac med Apple T2-säkerhetskrets slås på utför kretsen en säker start från sin Boot ROM på samma sätt som iPhone, iPad och Mac-datorer med Apple Silicon. Det här verifierar iBoot-bootloadern och är det första steget i tillförlitlighetskedjan. iBoot kontrollerar kärnan och kärntilläggskoden i T2-kretsen som sedan kontrollerar den fasta Intel UEFI-programvaran. Den fasta UEFI-programvaran och tillhörande signatur är initialt endast tillgängliga för T2-kretsen.




Efter verifiering mappas avbilden för den fasta UEFI-programvaran till en del av T2-kretsens minne. Det här minnet görs tillgängligt för Intel-processorn via eSPI (enhanced Serial Peripheral Interface). Första gången Intel-processorn startas hämtar processorn den fasta UEFI-programvaran via eSPI:t från den integritetskontrollerade, minnesmappade kopian av den fasta programvaran som finns i T2-kretsen.

Utvärderingen av tillförlitlighetskedjan fortsätter på Intel-processorn där den fasta UEFI-programvaran utvärderar signaturen för boot.efi, vilket är macOS-bootloadern. De Intel-specifika signaturerna för säker start av macOS lagras i samma Image4-format som används för säker start av iOS, iPadOS och T2-kretsen, och den kod som tolkar Image4-filerna är samma hårda kod från den aktuella säkra iOS- och iPadOS-bootimplementeringen. Boot.efi verifierar i sin tur signaturen för en ny fil som heter immutablekernel. När säker start används representerar filen immutablekernel den fullständiga uppsättningen Apple-kärntillägg som krävs för att starta macOS. Policyn för säker start avslutas vid överlämningen till immutablekernel, och därefter börjar macOS-säkerhetspolicyer (som systemintegritetsskydd och signerade kärntillägg) att gälla.

Om några fel uppstår under den här processen går Mac-datorn in i återställningsläge, återställningsläge för Apple T2-säkerhetskretsen eller DFU-läge (Device Firmware Upgrade) för Apple T2-säkerhetskretsen.

### Microsoft Windows på Intel-baserade Mac-datorer med T2-krets

Som förval litar en Intel-baserad Mac som har stöd för säker start endast på innehåll som är signerat av Apple. För att förbättra säkerheten för Boot Camp-installationer har dock Apple även stöd för säker start av Windows. Den fasta UEFI (Unified Extensible Firmware Interface)-programvaran innehåller en kopia av certifikatet Microsoft Windows Production CA 2011 som används till att autentisera Microsoft-bootloaders.

*Obs!* Det finns för närvarande ingen tillförlitlighetsfunktion för Microsoft Corporation UEFI CA 2011, vilket skulle kunna tillåta verifiering av kod som är signerad av Microsoft-partners. Detta UEFI CA används ofta till att verifiera autenticiteten för bootloaders för andra operativsystem, exempelvis Linux-varianter.

Stöd för säker start av Windows är inte aktiverat som förval, utan det aktiveras istället med Boot Camp-assistent (BCA). När en användare kör BCA konfigureras macOS om till att lita på kod som är signerad av Microsoft som första part vid start. När BCA är slutfört, och om macOS inte godkänns vid Apple-tillförlitlighetsutvärderingen under säker start, försöker den fasta UEFI-programvaran utvärdera objektets tillförlitlighet i enlighet med formateringen för säker UEFI-start. Om tillförlitlighetsutvärderingen lyckas fortsätter Mac-datorn och Windows startas. Om den inte lyckas öppnas recoveryOS och användaren får ett meddelande om att tillförlitlighetsutvärderingen har misslyckats.

### Intel-baserade Mac-datorer utan T2-krets

Intel-baserade Mac-datorer utan T2-krets saknar stöd för säker start. Därför läser den fasta UEFI (Unified Extensible Firmware Interface)-programvaran in macOS-bootern (boot.efi) från filsystemet utan verifiering, och bootern läser in kärnan (prelinkedkernel) från filsystemet utan verifiering. För att skydda startsekvensens integritet bör användarna aktivera samtliga följande säkerhetsmekanismer:

- *Systemintegritetsskydd (SIP)* Aktiverat som förval. Skyddar bootern och kärnan mot skadlig skrift från inuti ett macOS som körs.
- *FileVault:* Det här kan aktiveras på två sätt: av användaren eller av en MDM-administratör. Detta skyddar mot en fysiskt närvarande angripare som försöker använda hårddiskläge till att skriva över bootern.
- *Lösenord för fast programvara:* Det här kan aktiveras på två sätt: av användaren eller av en MDM-administratör. Detta förhindrar att en fysiskt närvarande angripare startar ett alternativt startläge, som recoveryOS, enanvändarläge eller hårddiskläge, från vilka bootern kan skrivas över. Detta förhindrar också start från alternativa medier som en angripare kan använda till att försöka köra kod som skriver över bootern.

## Startlägen för Intel-baserade Mac-datorer med Apple T2-säkerhetskrets

Intel-baserade Mac-datorer med Apple T2-säkerhetskrets har flera olika startlägen som det går att välja mellan vid start genom att trycka på olika tangentkombinationer som kan identifieras av den fasta UEFI-programvaran eller bootern. Vissa startlägen, som enanvändarläge, fungerar inte förrän säkerhetspolicyn ändras till Ingen säkerhet i Startsäkerhetsverktyg.

| Läge | Tangentkombination | Beskrivning |
|---|---|---|
| macOS-start | Ingen | Den fasta UEFI-programvaran lämnar över till macOS-bootern (en UEFI-app) som lämnar över till macOS-kärnan. Vid standardstart av en Mac med FileVault aktiverat presenterar macOS-bootern gränssnittet för inloggningsfönstret där lösenordet för avkryptering av lagringsutrymmet anges. |
| Starthanteraren | Alternativ (⌥) | Den fasta UEFI-programvaran startar den inbyggda UEFI-appen som visar ett gränssnitt för val av startenhet för användaren. |
| Hårddiskläge | T | Den fasta UEFI-programvaran startar den inbyggda UEFI-appen som exponerar den interna lagringsenheten som en rå, blockbaserad lagringsenhet via FireWire, Thunderbolt eller USB eller valfri kombination av alla tre (beroende på Mac-modell). |
| Enanvändarläge | Kommando (⌘)-S | macOS-kärnan överför flaggan –s i launchd:s argumentvektor, varpå launchd skapar enanvändarskalet i appen Systemmeddelandens tty.<br>*Obs!* Om användaren avslutar kommandotolken fortsätter macOS starten till inloggningsfönstret. |
| recoveryOS | Kommando (⌘)-R | Den fasta UEFI-programvaran läser in ett minimalt macOS från en signerad skivavbildsfil (.dmg) på den interna lagringsenheten. |
| Internet recoveryOS | Alternativ (⌥)-kommando (⌘)-R | Den signerade skivavbilden hämtas från internet via HTTP. |
| Diagnostik | D | Den fasta UEFI-programvaran läser in en minimal UEFI-diagnosmiljö från en signerad skivavbildsfil på den interna lagringsenheten. |
| Internetdiagnostik | Alternativ (⌥)-D | Den signerade skivavbilden hämtas från internet via HTTP. |
| Windows-start | Ingen | Om Windows har installerats med Boot Camp lämnar den fasta UEFI-programvaran över till Windows-bootern som lämnar över till Windows-kärnan. |

## Startsäkerhetsverktyg på Mac-datorer med Apple T2-säkerhetskrets

### Översikt

På Intel-baserade Mac-datorer med en Apple T2-säkerhetskrets hanterar Startsäkerhetsverktyg flera olika säkerhetspolicyinställningar. Du kommer åt verktyget som genom att starta med recoveryOS och sedan välja Startsäkerhetsverktyg från menyn Verktyg. Verktyget skyddar säkerhetsinställningar som stöds från att enkelt manipuleras av en angripare.


Autentisering krävs för kritiska policyändringar även i återställningsläge. Första gången Startsäkerhetsverktyg öppnas ber det användaren att ange ett administratörslösenord från den primära macOS-installationen som är associerad med det recoveryOS som är startat för tillfället. Om det inte finns någon administratör måste du skapa en innan policyn kan ändras. T2-kretsen kräver att Mac-datorn har startats i recoveryOS och att en autentisering med en Secure Enclave-förknippad ID-handling har skett innan en sådan policyändring kan genomföras. Säkerhetspolicyändringar har två implicita krav. recoveryOS måste:

- Startas från en lagringsenhet direkt ansluten till T2-kretsen eftersom partitioner på andra enheter inte har Secure Enclave-förknippade ID-handlingar som är knutna till den interna lagringsenheten.
- Finnas på en APFS-baserad volym eftersom det endast finns stöd för att lagra autentiseringen i ID-handlingarna för återställning som skickas till Secure Enclave på APFS-förstartsvolymen på en enhet. HFS plus-formaterade volymer fungerar inte med säker start.

Den här policyn visas endast i Startsäkerhetsverktyg på Intel-baserade Mac-datorer med en T2-krets. Även om det i det flesta fall inte krävs några ändringar i policyn för säker start så är det användarna som slutligt bestämmer över inställningarna på sina enheter, och beroende på sina behov kan de välja att avaktivera eller nedgradera funktionen för säker start på datorn.

Ändringar av policyn för säker start som sker via den här appen används endast för utvärderingen av tillförlitlighetskedjan som verifieras på Intel-processorn. Alternativet Säker start för T2-kretsen är alltid aktiverat.

Policyn för säker start kan konfigureras till en av tre inställningar: Full säkerhet, Medelsäkerhet och Ingen säkerhet. Ingen säkerhet avaktiverar utvärderingen vid säker start på Intel-processorn helt och hållet och låter användarna starta vad de vill.

### Startpolicyn Full säkerhet

Full säkerhet är den förvalda startpolicyn, och den beter sig ungefär på samma sätt som iOS och iPadOS eller Full säkerhet på Mac-datorer med Apple Silicon. När programvaran har hämtats och är förberedd för installation anpassas den med en signatur som innehåller ett ECID (Exclusive Chip Identification) – ett unikt ID som i det här fallet är specifikt för T2-kretsen – som en del av en signeringsbegäran. Den signatur som levereras av signeringsservern är sedan unik och kan endast användas av just den aktuella T2-kretsen. Den fasta UEFI (Unified Extensible Firmware Interface)-programvaran ska säkerställa att en levererad signatur inte bara är signerad av Apple, utan att den är signerad för den aktuella Mac-datorn när policyn Full säkerhet används. I praktiken binder den just den versionen av macOS till just den datorn. Detta hjälper till att förhindra nedgraderingsangrepp på det sätt som beskrivs för Full säkerhet på Mac-datorer med Apple Silicon.

### Startpolicyn Medelsäkerhet

Startpolicyn Medelsäkerhet påminner om en traditionell säker UEFI-start där en leverantör (Apple i det här fallet) genererar en digital signatur för koden för att säkerställa att den kommer från leverantören. På så vis kan angripare förhindras från att infoga osignerad kod. Den här signaturen kallas för en ”global” signatur eftersom den kan användas på valfri Mac och under obegränsad tid för en Mac där policyn Medelsäkerhet har ställts in. Varken iOS, iPadOS eller själva T2-kretsen har stöd för globala signaturer. Den här inställningen försöker inte förhindra nedgraderingsangrepp.

### Policy för mediestart

Policyn för mediestart finns endast på Intel-baserade Mac-datorer med T2-krets och är fristående från policyn för säker start. Så även om en användare avaktiverar säker start ändrar detta inte det förvalda beteendet att förhindra start från allt utom den lagringsenhet som är direkt kopplad till T2-kretsen. (Policy för mediestart krävs inte på Mac-datorer med Apple Silicon. Mer information finns i Kontroll av säkerhetspolicyn för Startskiva.)

## Lösenord för fast programvara på Intel-baserade Mac-datorer

macOS på Intel-baserade Mac-datorer med Apple T2-säkerhetskrets har stöd för ett lösenord för fast programvara i syfte att förhindra oavsiktliga ändringar av inställningarna för fast programvara på en specifik dator. Lösenord för fast programvara är utformad för att förhindra att en användare väljer alternativa startlägen som start med recoveryOS, enanvändarläge, start från en obehörig volym eller start i hårddiskläge.

*Obs!* Lösenordet för fast programvara krävs inte på Mac-datorer med Apple Silicon eftersom den kritiska fasta programvarufunktionen det begränsar har flyttats till recoveryOS och (när FileVault är aktiverat) recoveryOS kräver användarauktorisering innan dess kritiska funktioner kan nås.

Du kommer åt det mest grundläggande läget för lösenord för fast programvara via Firmware-lösenordshanterare i recoveryOS på Intel-baserade Mac-datorer *utan* T2-krets och via Startsäkerhetsverktyg på Intel-baserade Mac-datorer *med* T2-krets. Avancerade alternativ (som möjligheten att begära lösenordet vid varje start) är tillgängliga via kommandoradsverktyget `firmwarepasswd` i macOS.

Det är framförallt viktigt att ställa in ett lösenord för fast programvara för att minska risken för angrepp på Intel-baserade Mac-datorer utan T2-krets från en fysiskt närvarande angripare. Lösenordet för fast programvara kan hjälpa till att hindra angripare från att starta i recoveryOS där de sedan annars skulle kunna avaktivera systemintegritetsskydd (SIP). Och genom att begränsa start från alternativa medier kan en angripare inte köra behörig kod från ett annat operativsystem i syfte att angripa perifera fasta programvaror.

Det finns en återställningsmekanism för lösenordet för fast programvara för att hjälpa användare som har glömt bort lösenordet. Användarna kan trycka ned en tangentkombination vid start så att en modellspecifik sträng visas och sedan ge den strängen till AppleCare. AppleCare använder en digital signatur på en resurs som signaturkontrolleras av URI (Uniform Resource Identifier). Om signaturen godkänns, och innehållet är avsett för den specifika Mac-datorn, tar den fasta UEFI-programvaran bort lösenordet för fast programvara.

För användare som inte vill att någon annan än de själva ska kunna ta bort lösenordet för fast programvara genom att använda programvara har alternativet `–disable–reset–capability` lagts till i kommandoradsverktyget `firmwarepasswd` i macOS 10.15. Ifall lösenordet glöms och behöver tas bort måste logikkortet bytas ut. Innan det här alternativet ställs in måste användaren godkänna att stå för kostnaden vid ett eventuellt logikkortsbyte. Organisationer som vill skydda sina Mac-datorer från externa angripare och från anställda måste ställa in ett lösenord för fast programvara på system som ägs av organisationen. Detta kan göras på enheten på följande sätt:

- Vid tidpunkten för tillhandahållandet genom att manuellt använda kommandoradsverktyget `firmwarepasswd`.
- Med hanteringsverktyg från tredje part som använder kommandoradsverktyget `firmwarepasswd`.
- Via MDM (Mobile Device Management).

## Miljöer för recoveryOS och diagnos för Intel-baserade Mac-datorer

### recoveryOS

recoveryOS är helt separat från huvud-macOS och hela innehållet lagras i en skivavbild med namnet BaseSystem.dmg. Det finns också en tillhörande BaseSystem.chunklist som används till att verifiera integriteten för BaseSystem.dmg. Chunklistan är en serie hashvärden för 10 MB-delar av BaseSystem.dmg. Den fasta UEFI (Unified Extensible Firmware Interface)-programvaran utvärderar chunklist-filens signatur och utvärderar sedan hashen en del i taget från BaseSystem.dmg. Det säkerställer att den matchar det signerade innehållet i chunklistan. Om något hashvärde inte matchar avbryts starten från det lokala recoveryOS, och den fasta UEFI-programvaran försöker starta via Internet recoveryOS istället.

Om verifieringen slutförs korrekt öppnar den fasta UEFI-programvaran BaseSystem.dmg som en RAM-skiva och startar sedan den boot.efi-fil som finns i den. Den fasta UEFI-programvaran behöver inte göra någon separat kontroll av boot.efi, och inte heller behöver boot.efi kontrollera kärnan, eftersom det fullständiga innehållet i operativsystemet (av vilket dessa element bara utgör en delmängd) redan har integritetskontrollerats.

### Apple Diagnostics

Processen för start av den lokala diagnosmiljön är i huvudsak samma som start av recoveryOS. Separata AppleDiagnostics.dmg- och AppleDiagnostics.chunklist-filer används, men de verifieras på samma sätt som BaseSystem-filerna. Istället för att öppna boot.efi öppnar den fasta UEFI-programvaran en fil inuti den skivavbild (dmg-fil) som heter diags.efi, vilken i sin tur ansvarar för att anropa en mängd andra UEFI-drivrutiner som kan skapa gränssnitt med och kontrollera fel i maskinvaran.

### Miljö för Internet recoveryOS och diagnos

Om ett fel har uppstått vid start av miljön för lokal återställning eller diagnos försöker den fasta UEFI-programvaran hämta avbilderna från internet istället. (En användare kan också specifikt begära att avbilderna ska hämtas från internet genom att trycka ned en speciell tangentkombination vid start.) Integritetsvalideringen av skivavbilder och chunklistor från OS Recovery Server utförs på samma sätt som med avbilder som hämtas från en lagringsenhet.

Trots att anslutningen till OS Recovery Server sker via HTTP så integritetskontrolleras fortfarande det fullständigt hämtade innehållet på det sätt som beskrivits tidigare, och det är därför skyddat mot manipulation om en angripare har tagit kontroll över nätverket. Om integritetsverifikationen misslyckas för en enskild del begärs den om från OS Recovery Server 11 gånger innan försöken avbryts och ett felmeddelande visas.

När lägen för internetåterställning och diagnostik lades till på Mac-datorer under 2011 beslutades det att det var bättre att använda enklare HTTP-transport och hantera autentisering med hjälp av chunklist-mekanismen, snarare än att implementera mer komplicerad HTTPS-funktionalitet i den fasta UEFI-programvaran och på så visa öka den fasta programvarans angreppsyta.

# Säkerhet för signerade systemvolymer

I macOS 10.15 introducerade Apple den skrivskyddade systemvolymen, vilket är en dedikerad, isolerad volym för systeminnehåll. Med macOS 11 eller senare tillkommer starka kryptografiska skydd för systeminnehåll på en *signerad systemvolym (SSV)*. SSV har en kärnmekanism som verifierar integriteten hos systeminnehållet vid körning och avvisar alla data – kod och icke-kod – som saknar giltig kryptografisk signatur från Apple. Från och med iOS 15 och iPadOS 15 har systemvolymen på en IPhone eller iPad även det kryptografiska skyddet hos en signerad systemvolym.

SSV hjälper inte bara till att förhindra manipulering av någon Apple-programvara som ingår i operativsystemet, utan det gör dessutom uppdateringen av macOS-programvara tillförlitligare och mycket säkrare. Och eftersom SSV använder APFS (Apple File System)-ögonblicksbilder kan den gamla systemversionen återskapas utan ominstallation om en uppdatering inte kan utföras.

Sedan introduktionen har APFS tillhandahållit integritet för filsystemets metadata genom användning av icke-kryptografiska kontrollsummor på den interna lagringsenheten. SSV stärker integritetsmekanismen genom att lägga till kryptografiska hashvärden och utökar den på så vis till att omfatta alla fildatabyte. Data från den interna lagringsenheten (inklusive filsystemmetadata) hashas kryptografiskt i lässökvägen, och hashvärdet jämförs sedan med ett förväntat värde i filsystemets metadata. Om värdena inte stämmer antar systemet att data har manipulerats och skickar inte tillbaka något till den programvara som skickat förfrågan.

Varje SSV-SHA256-hash lagras i huvudmetadataträdet för filsystemet, vilket i sig också hashas. Och eftersom varje nod i trädet rekursivt verifierar integriteten för hashvärdena för dess underordnade noder – ungefär som ett binärt hash-träd (Merkel-träd) – omfattar rotnodens hashvärde (kallas *sigill*) därför alla databyte i SSV, vilket innebär att den kryptografiska signaturen täcker hela systemvolymen.

Under installation och uppdatering av macOS beräknas sigillet om från filsystemsenheten och det värdet verifieras mot värdet som Apple har signerat. På Mac-datorer med Apple Silicon verifierar bootloadern sigillet innan styrningen överförs till kärnan. På Intel-baserade Mac-datorer med Apple T2-säkerhetskrets vidarebefordrar bootloadern mätvärdet och signaturen till kärnan som sedan verifierar sigillet direkt innan inlänkning av rotfilsystemet. I båda fallen avbryts startprocessen om verifieringen misslyckas och användaren blir uppmanad att installera om macOS. Den här processen upprepas vid varje start om inte användaren har valt ett lägre säkerhetsläge och dessutom separat har valt att avaktivera den signerade systemvolymen.

Under iOS- och iPadOS-programuppdateringar blir systemvolymen förberedd och beräknas om på ett liknande sätt. iOS- och iPadOS-bootloaders verifierar att sigillet är intakt och matchar ett Apple-signerat värde innan enheten får starta kärnan. Ifall en matchning inte stämmer vid start uppmanas användaren att uppdatera systemprogramvaran på enheten. Användare får inte avaktivera skyddet hos en signerad systemvolym i iOS och iPadOS.

## SSV och kodsignering

Kodsignering sker fortfarande och genomdrivs av kärnan. Den signerade systemvolymen tillhandahåller skydd varje gång byte läses från den interna lagringsenheten. Kodsignering tillhandahåller däremot skydd när Mach-objekt minnesmappas som körbara. Både SSV och kodsignering skyddar körbar kod på alla läs- och körsökvägar.

## SSV och FileVault

I macOS 11 eller senare tillhandahålls motsvarande skydd för systeminnehåll under vila av SSV-volymen, och därför behöver systemvolymen inte längre vara krypterad. Eventuella ändringar som görs i filsystemet under vila upptäcks av filsystemet när de läses.
Om användaren har slagit på FileVault är användarens innehåll på datavolymen fortfarande krypterat med en hemlighet som tillhandahålls av användaren.

Om användaren väljer att avaktivera SSV-volymen blir systemet sårbart för manipulering vid vila, och sådan manipulering kan göra det möjligt för angripare att extrahera krypterade användardata nästa gång systemet startar. Därför tillåter systemet inte att användaren avaktiverar SSV-volymen om FileVault är påslaget. Skydd under vila måste konsekvent vara aktiverat eller avaktiverat för båda volymerna.

I macOS 10.15 eller tidigare skyddar FileVault operativsystemprogramvara under vila genom att kryptera användar- och systeminnehåll med en nyckel som skyddades av en hemlighet tillhandahållen av användaren. Detta skydd innebär att angripare med fysisk tillgång till enheten inte kan komma åt eller effektivt ändra det filsystem som innehåller systemprogramvaran.

## SSV och Mac-datorer med Apple T2-säkerhetskrets

På Mac-datorer med Apple T2-säkerhetskrets skyddas endast själva macOS av SSV-volymen. Den programvara som körs på T2-kretsen och verifierar macOS skyddas av säker start.

# Säkra programuppdateringar

Säkerhet är en process. Det räcker inte att bara starta den operativsystemsversion som installerats på fabriken – det måste också finnas en mekanism för att snabbt och säkert hämta de senaste säkerhetsuppdateringarna. Apple släpper regelbundet programuppdateringar för att åtgärda nyfunna säkerhetsproblem. Användare med iPhone- och iPad-enheter får uppdateringsnotiser på enheterna. Mac-användare hittar tillgängliga uppdateringar i Systeminställningar. Uppdateringar levereras trådlöst för snabb installation av de senaste säkerhetsfixarna.

## Säkerhet vid uppdatering

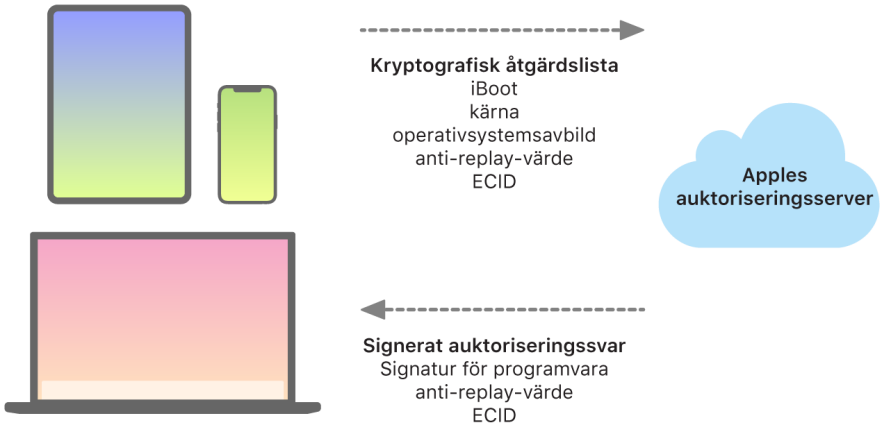
Uppdateringsprocessen använder samma maskinvarubaserade betrodda rot som används av säker start och som är utformad för att endast installera Apple-signerad kod. Uppdateringsprocessen använder också auktorisering av systemprogramvara för att se till att endast kopior av operativsystemversioner som aktivt signeras av Apple kan installeras på iPhone- och iPad-enheter, eller på Mac-datorer med inställningen Full säkerhet konfigurerad som policy för säker start i Startsäkerhetsverktyg. När de här säkerhetsprocesserna är aktiva kan Apple sluta signera äldre operativsystemsversioner med kända svagheter och minska risken för nedgraderingsangrepp.

För ökad säkerhet vid programuppdateringar hämtas och installeras en fullständig kopia av iOS eller iPadOS när en enhet som ska uppgraderas är fysiskt ansluten till en Mac. Vid trådlös programuppdatering hämtas dock inte hela operativsystemet, utan *endast de komponenter som krävs för att slutföra uppdateringen* vilket ger en effektivare nätverksanvändning. Dessutom kan programuppdateringar cachelagras på en Mac med macOS 10.13 eller senare där Innehållscachelagring är aktiverat så att iPhone- och iPad-enheterna inte behöver ansluta till Apples servrar för att komma åt nödvändiga uppdateringar. (De måste fortfarande kontakta Apples servrar för att slutföra uppdateringsprocessen.)

## Personligt anpassad uppdateringsprocess

Vid uppgraderingar och uppdateringar blir viss information tillgänglig för Apples auktoriseringsserver för installation. Via anslutningen skickas en lista med kryptografiska värden för de olika delarna av installationspaketet (exempelvis iBoot, kärnan och operativsystemsavbilden), ett slumpmässigt anti-replay-värde samt enhetens unika ID som kallas ECID (Exclusive Chip Identification).

Auktoriseringsservern jämför listan över åtgärder med de versioner vars installation är tillåten, och om de överensstämmer lägger den till enhetens ECID och signerar resultatet. Servern överför en komplett uppsättning signerade data till enheten under uppgraderingsprocessen. Att lägga till enhetens ECID är ett sätt att göra auktoriseringen unik för enheten. Genom att endast auktorisera och signera kända åtgärder försäkrar sig servern om att uppdateringen sker exakt som Apple har avsett.




Under startprocessens utvärdering av tillförlitlighetskedjan verifieras att signaturen kommer från Apple och att åtgärden för objektet som har lästs in från lagringsenheten, i kombination med enhetens ECID, matchar det som signaturen avser. På enheter som har stöd för personlig anpassning är det tänkt att dessa steg ska säkerställa att auktoriseringen gäller för en specifik enhet, och att ett äldre operativsystem eller en äldre version av fast programvara inte kan kopieras från en enhet till en annan. Anti-replay-värdet förhindrar att obehöriga sparar serverns svar och använder det till att manipulera en enhet eller modifiera systemprogramvaran.

Anpassningsprocessen är orsaken till varför en nätverksanslutning till Apple alltid krävs för uppdatering av enheter med Apple-utformade kretsar, inklusive Intel-baserade Mac-datorer med Apple T2-säkerhetskrets.

På enheter med Secure Enclave använder den maskinvaran också auktorisering av systemprogramvara till att kontrollera integriteten hos den egna programvaran och är utformad för att förhindra nedgraderingar.

# Operativsystemets integritet

Apples operativsystemprogramvara har utformats med fokus på säkerhet. Designen har en maskinvarubaserad betrodd rot, som används till att säkerställa säker start, och en säker och snabb programuppdateringsprocess. Apples operativsystem använder också inbyggda, funktionsspecifika kretsbaserade maskinvarufunktioner som bidrar till att förhindra obehörigt utnyttjande av systemet när det körs. Körningsfunktionerna skyddar den betrodda kodens integritet medan den körs. Kortfattat hjälper Apples operativsystemprogramvara till att begränsa angrepp och utnyttjandet av tekniker, vare sig de har sitt ursprung i en skadlig app, på webben eller kommit via någon annan kanal. De skydd som listas här är tillgängliga på enheter med Apple-utformade SoC:er. Här ingår bland annat iOS, iPadOS, tvOS, watchOS och nu även macOS på Mac-datorer med Apple Silicon.

| Funktion | A10 | A11, S3 | A12, A13, A14 S4–S9 | A15, A16, A17 | M1, M2, M3 |
|---|---|---|---|---|---|
| Kärnintegritetsskydd | ✓ | ✓ | ✓ | ✓ | ✓ |
| Begränsningar för snabb behörighet | ✗ | ✓ | ✓ | ✓ | ✓ |
| Integritetsskydd för systemcoprocessor | ✗ | ✗ | ✓ | ✓ | ✓ |
| Pekarautentiseringskoder | ✗ | ✗ | ✓ | ✓ | ✓ |
| Page Protection Layer | ✗ | ✓ | ✓ | ✓ | ✗ Se Anmärkning 1 nedan. |
| Secure Page Table Monitor | ✗ | ✗ | ✗ | ✓ Se Anmärkning 2 nedan. | ✗ |

*Anmärkning 1:* PPL (Page Protection Layer) kräver att plattformen kör *endast* signerad och betrodd kod – detta är en säkerhetsmodell som inte gäller i macOS.

*Anmärkning 2:* Secure Page Table Monitor (SPTM) stöds i A15, A16 och A17 och ersätter Page Protection Layer på plattformar som stöds.

## Kärnintegritetsskydd

När operativsystemets kärna slutför initiering aktiveras ett kärnintegritetsskydd (Kernel Integrity Protection, KIP) som förhindrar ändringar av kärn- och drivrutinskod. Minnesstyrenheten tillhandahåller en skyddad fysisk minnesregion som iBoot använder till att läsa in kärnan och kärntillägg. När starten är färdig nekar minnesstyrenheten skrivningar till den skyddade fysiska minnesregionen. Approcessorns minneshanteringsenhet (Memory Management Unit, MMU) är konfigurerad så att den förhindrar mappning av privilegierad kod från fysiskt minne utanför den skyddade minnesregionen och förhindrar skrivbara mappningar av fysiskt minne i kärnminnesregionen.

För att förhindra omkonfigurering låses maskinvaran som används till att aktivera KIP när startprocessen är klar så att den inte går att omkonfigurera.

## Begränsningar för snabb behörighet

Från och med Apple A11 Bionic och S3 SoC har ett nytt maskinvaruprimitiv introducerats. Detta primitiv, begränsningar för snabb behörighet, innehåller ett processorregister som snabbt begränsar behörigheterna per tråd. Med dessa begränsningar för snabb behörighet (eller APRR-register) kan operativsystem som stöds ta bort körbehörigheter från minnet – utan hantering av systemanrop och sidtabeller. De här registren begränsar ytterligare möjligheten för angrepp via webben, framförallt från kod som kompileras vid körning (just-in-time-kompilerat, JIT) eftersom minne i praktiken inte kan köras samtidigt som information läses och skrives till det.

## Integritetsskydd för systemcoprocessor

Den fasta coprocessorprogramvaran hanterar många kritiska systemuppgifter – t.ex. Secure Enclave, bildsensorprocessorn och rörelsecoprocessorn. Därför är dess säkerhet central för säkerheten för systemet som helhet. För att förhindra ändring av coprocessorns fasta programvara använder Apple en mekanism som kallas *integritetsskydd för systemcoprocessor (SCIP)*.

SCIP fungerar ungefär som kärnintegritetsskydd (KIP): Vid start läser iBoot in de enskilda coprocessorernas fasta programvara i en skyddad minnesregion som är reserverad och separat från KIP-regionen. iBoot konfigurerar varje coprocessors minnesenheter för att förhindra:

- Körbara mappningar utanför dess del av den skyddade minnesregionen
- Skrivbara mappningar i dess del av den skyddade minnesregionen

Vid start används även Secure Enclaves operativsystem för att konfigurera SCIP för Secure Enclave. Maskinvaran som används till att aktivera SCIP låses när startprocessen har slutförts. Det ska förhindra omkonfigurering.

## Pekarautentiseringskoder

Pekarautentiseringskoder (PAC) används som skydd mot utnyttjande av minnesfelsbuggar. Systemprogramvara och inbyggda appar använder PAC till att förhindra ändring av funktionspekare och returadresser (kodpekare). PAC använder fem hemliga 128-bitarsvärden till att signera kärninstruktioner och data och varje användarutrymmesprocess har sina egna B-nycklar. Objekt saltas och signeras enligt vad som anges nedan.

| Objekt | Nyckel | Salt |
|---|---|---|
| Funktionsreturadress | IB | Lagringsadress |
| Funktionspekare | IA | 0 |
| Blockanropsfunktion | IA | Lagringsadress |
| Objective-C-metodcache | IB | Lagringsadress + klass + selektor |
| C++ V-tabellposter | IA | Lagringsadress + hash (mangled method name) |
| Beräknad Goto-etikett | IA | Hash (funktionsnamn) |
| Kärntrådstatus | GA | • |
| Statusregister för användartråd | IA | Lagringsadress |
| C++ V-tabellpekare | DA | 0 |

Signaturvärdet lagras i de oanvända utfyllnadsbitarna överst i 64-bitarspekaren. Signaturen verifieras före användning och utfyllnaden återställs för att säkerställa en fungerande pekaradress. Om verifieringen misslyckas avbryts åtgärden. Den här verifieringen gör det svårare att utföra många angrepp, t.ex. ett ROP-angrepp (Return-Oriented Programming) som försöker lura enheten att köra befintlig kod i skadligt syfte genom att manipulera funktionsreturadresser som lagras i stacken.

## Page Protection Layer

Page Protection Layer (PPL) i iOS, iPadOS och watchOS ska förhindra att användarutrymmeskod ändras efter att verifieringen av kodsignaturen har slutförts. PPL bygger på KIP och APRR och hanterar förbigångar av sidtabellbehörigheten för att säkerställa att endast PPL kan ändra skyddade sidor som innehåller användarkod och sidtabeller. Systemet minskar angreppsytan kraftigt genom stöd för upprätthållande av kodens integritet genom hela systemet, till och med om kärnan har manipulerats. Detta skydd finns inte i macOS eftersom PPL endast omfattar system där all kod som körs måste vara signerad.

## Secure Page Table Monitor och Trusted Execution Monitor

Secure Page Table Monitor (SPTM) och Trusted Execution Monitor (TXM) är utformade så att de samverkar för att skydda sidtabeller för både användar- och kärnprocesser mot ändringar även när angripare har skrivbehörighet till kärnan och kan förbigå kontrollflödesskydd. SPTM gör detta genom att använda en högre behörighetsnivå än kärnan och använder TMX:en med lägre behörighet till att faktiskt genomdriva policyerna som styr kodkörning. Det här systemet är utformat så att en TMX-kompromiss inte innebär att SPTM automatiskt förbigås på grund av att deras behörigheter är separerade och styrningen av tillförlitlighet mellan dem. I SoC:erna A15, A16 och A17 ersätts PPL av SPTM (i kombination med TXM) som har en mindre angreppsyta och inte förlitar sig på tillförlitligheten hos kärnan ens under tidig start. SPTM bygger även på nya kretsprimitiv som är en utveckling av Fast Permission Restrictions som används av PPL.

# Aktivera dataanslutningar säkert

På iPhone- och iPad-enheter samt Mac-datorer, där ingen dataanslutning har upprättats nyligen, måste användare använda Face ID, Touch ID eller en lösenkod till att aktivera dataanslutningar via ett Thunderbolt-, USB-, Lightning-, Smart Connector-gränssnitt eller – i macOS 13.3 eller senare – SDXC (SD Extended Capacity)-kortgränssnitt. Det här begränsar möjligheter till angrepp från fysiskt anslutna enheter (exempelvis sabotageladdare) samtidigt som andra tillbehör kan användas inom rimliga tidsgränser. Om det är mer än en timme sedan iPhone eller iPad låstes, eller sedan ett tillbehörs dataanslutning kopplades från, kommer enheten inte att tillåta några nya dataanslutningar förrän enheten blir upplåst. Under denna 60-minutersperiod tillåts endast dataanslutningar från tillbehör som tidigare har varit anslutna till enheten när den varit olåst. De här tillbehören koms ihåg i 30 dagar efter den senaste gången de anslöts. Om ett okänt tillbehör försöker att upprätta en dataanslutning under denna period avaktiveras alla dataanslutningar för tillbehör via dessa anslutningar tills enheten blir upplåst igen. 60-minutersperioden:

- Ser till att användare som ofta ansluter enheten till en Mac eller PC, till tillbehör eller med en kabel till CarPlay inte behöver ange sin lösenkod varje gång de ansluter enheten.
- Är nödvändig eftersom ekosystemet för tillbehör saknar ett kryptografiskt tillförlitligt sätt att identifiera tillbehör innan en dataanslutning upprättas.

Dessutom kommer enheter att neka nya dataanslutningar omedelbart efter låsning om det är mer än tre dagar sedan en dataanslutning upprättades. Detta sker för att öka säkerheten för användare som sällan använder sådana tillbehör. De här dataanslutningarna avaktiveras även så fort enheten befinner sig i ett läge där den kräver en lösenkod för att återaktivera biometrisk autentisering.

Användaren kan välja att återaktivera alltid på-dataanslutningar i Inställningar (vid installation av vissa hjälpmedelsenheter sker detta automatiskt).

# Verifiera tillbehör för iPhone och iPad

Licensprogrammet Made for iPhone and iPad (MFi) ger granskade och godkända tillbehörstillverkare tillgång till iAP-protokollet (iPod Accessories Protocol) och nödvändiga maskinvarukomponenter.

När ett MFi-tillbehör kommunicerar med en iPhone eller iPad måste tillbehöret bevisa för Apple att det har granskats. (Tillbehöret ansluts till enheten via Thunderbolt, Lightning, Bluetooth eller för vissa enheter USB-C.) Som bevis på auktorisering skickar tillbehöret ett certifikat som Apple tillhandahållit till enheten och enheten verifierar detta. Enheten skickar då en utmaning som tillbehöret måste besvara med ett signerat svar. Den här processen hanteras helt av en anpassad integrerad krets (IC) som Apple tillhandahåller till godkända tillbehörstillverkare, och den är osynlig för tillbehöret självt.

Verifierade MFi-tillbehör kan begära tillgång till andra överföringsmetoder och funktioner – till exempel tillgång till digitala ljudströmmar via Thunderbolt-kabeln eller platsinformation som skickas via Bluetooth. En integrerad krets för autentisering är utformad för att säkerställa att endast godkända MFi-tillbehör ges full tillgång till enheten. Om ett tillbehör inte kan autentiseras begränsas dess tillgång till analogt ljud och ett litet urval av seriella (UART) reglage för ljuduppspelning.

AirPlay använder också integrerade autentiseringskretsar för att verifiera att mottagarna har godkänts av Apple. Strömmat AirPlay-ljud och CarPlay-video använder MFi-SAP (Secure Association Protocol) som krypterar kommunikationen mellan tillbehöret och enheten med hjälp av AES128 i CTR-läge (Counter). Tillfälliga nycklar utväxlas under ECDH-nyckelutbytet (Curve25519) och signeras med den integrerade autentiseringskretsens 1 024-bitars RSA-nyckel som en del av protokollet STS (Station-to-Station).

# BlastDoor för Meddelanden och IDS

iOS, iPadOS och macOS innehåller säkerhetsfunktionen *BlastDoor* som först introducerades i iOS 14 och relaterade släpp. Målet för BlastDoor är att skydda systemet genom att stänga in angripare och öka komplexiteten i deras ansträngningar att utnyttja Meddelanden och IDS (Apple Identity Service). BlastDoor isolerar, tolkar, kodningskonverterar och validerar obetrodda data som anländer via Meddelanden, IDS och andra vektorer i syfte att förhindra angrepp.

BlastDoor gör det genom att använda sandlådebegränsningar och minnessäker validering av utmatning, vilket skapar ett betydande hinder som angripare måste övervinna innan de når andra delar av operativsystemet. Funktionen är utformad så att den kraftigt förbättrar användarskyddet mot angrepp, i synnerhet mot nollklicksangrepp som inte kräver interaktion från användaren.

Slutligen behandlar Meddelanden trafik från ”kända avsändare” annorlunda än trafik från ”okända avsändare” genom att erbjuda olika funktionsuppsättningar till varje grupp och segmentera ”kända” och ”okända” data i distinkta BlastDoor-instanser.

# Säkerhet med Låst läge för Apple-enheter

Låst läge är ett frivilligt, extremt skydd som är utformat för det fåtal personer som på grund av vilka de är eller vad de gör kan utgöra personliga måltavlor för några av de mest sofistikerade digitala hoten som riktade finansiella spionprogram. De flesta användarna blir aldrig utsatta för attacker av den här typen.

När Låst läge är påslaget fungerar en enhet inte som vanligt. I syfte att minska den potentiella angreppsytan blir vissa appar, webbplatser och funktioner kraftigt begränsade av säkerhetsskäl och vissa upplevelser kanske inte alls är tillgängliga.

Låst läge är tillgängligt i iOS 16, iPadOS 16, macOS 13 och watchOS 10 eller senare. Ytterligare skydd är tillgängliga i iOS 17, iPadOS 17, macOS 14 och uppdateringar av watchOS 10.1 eller senare. För att dra nytta av ytterligare funktioner i Låst läge bör enheter uppdateras till det senaste operativsystemet. Mer information finns i Apple Support-artikeln Om låsningsläge.

Låst läge kompromissar för att öka säkerheten på bekostnad av funktionalitet, prestanda eller både och. De här kompromisserna påverkar:

- Bakgrundstjänster
- Anslutningar
- Enhetshantering
- FaceTime
- Game Center
- Mail
- Meddelanden
- Bilder
- Safari
- Systeminställningar
- WebKit

# Ytterligare funktioner för macOS-systemsäkerhet

## Ytterligare funktioner för macOS-systemsäkerhet

macOS körs på en bredare uppsättning maskinvaror (t.ex. Intel-baserade processorer, Intel-baserade processorer i kombination med Apple T2-säkerhetskretsen och Apple-kretsbaserade SoC:er) och har stöd för mängder av olika användningsområden och allmänna beräkningsåtgärder. Medan vissa användare endast använder de grundläggande förinstallerade apparna, eller de som är tillgängliga via App Store, är andra kärnhackare som behöver avaktivera i stort sett allt plattformsskydd så att de kan köra och testa sin kod med högsta möjliga betroddhet. De flesta ligger någonstans i mitten och många användare har kringutrustning och programvara som kräver olika åtkomstnivåer. Apple designade macOS-plattformen med integrering i åtanke för maskinvara, programvara och tjänster som tillhandahåller säkerhet i sig själva – en plattform som gör det enkelt att konfigurera, driftsätta och hantera systemet men samtidigt upprätthåller den konfigurerbarhet som användarna förväntar sig. macOS innehåller de viktiga säkerhetstekniker som IT-proffs behöver för att skydda företagsdata och integrera i säkra företagsnätverksmiljöer.

Följande funktioner stöder och bidrar till att skydda de olika behoven hos macOS-användare. De inkluderar:

- Säkerhet för signerade systemvolymer
- Systemintegritetsskydd
- Tillförlitlighetscacher
- Skydd för kringutrustning
- Stöd och skydd för Rosetta 2 (automatisk översättning) på Mac-datorer med Apple Silicon
- Stöd och skydd för DMA
- Stöd och skydd för kärntillägg
- Stöd och skydd för Option ROM
- Fast UEFI-programvara för Intel-baserade Mac-datorer

## Systemintegritetsskydd

macOS använder kärnbehörigheter till att begränsa skrivbarheten för kritiska systemfiler med en egenskap som kallas *systemintegritetsskydd (SIP)*. Det här funktionen är separat och utöver det maskinvarubaserade KIP som är tillgängligt på Mac-datorer med Apple Silicon och som skyddar mot ändringar av kärnan i minnet. En teknik för obligatoriska åtkomstkontroller används för att tillhandahålla detta och ett antal andra skydd på kärnnivå, inklusive sandlådor och datavalv.

### Obligatoriska åtkomstkontroller

macOS använder obligatoriska åtkomstkontroller som kallas MAC (Mandatory Access Controls). Det är policyer som anger säkerhetsbegränsningar som har skapats av utvecklaren och som inte kan förbigås. Det här sättet skiljer sig från DAC (Discretionary Access Controls) som tillåter användarna att förbigå säkerhetspolicyer.

MAC syns inte för användaren, men det är den underliggande tekniken som gör många viktiga funktioner möjliga, exempelvis sandlådor, föräldrakontroll, hanterade inställningar, tillägg och systemintegritetsskydd.

### Systemintegritetsskydd

*Systemintegritetsskydd* skrivskyddar komponenter på vissa kritiska filsystemplatser för att förhindra att skadlig kod ändrar dem. Systemintegritetsskydd är en datorspecifik inställning som är aktiverad som förval när användaren uppgraderar till OS X 10.11 eller senare. Om du avaktiverar den på en Intel-baserad Mac tas skyddet bort för alla partitioner på den fysiska lagringsenheten. macOS använder den här säkerhetspolicyn för alla processer som körs i systemet, oavsett om de körs i sandlåda eller med administratörsbehörighet.

## Tillförlitlighetscacher

Ett av objekten som ingår i kedjan vid säker start är den statiska tillförlitlighetscachen som är en betrodd registrering av alla Mach-O-binärfiler som används med den signerade systemvolymen. Varje Mach-O representeras av en kodkataloghash. För effektiv sökning sorteras dessa hashvärden innan de infogas i tillförlitlighetscachen. Kodkatalogen är resultatet av den signeringsoperation som utförs av `codesign(1)`. Tillämpning av tillförlitlighetscachen kräver att SIP hela tiden är aktiverat. För att tillämpningen av tillförlitlighetscachen ska kunna avaktiveras på Mac-datorer med Apple Silicon måste därför säker start konfigureras till Tillåtande säkerhet.

När en binärfil körs (som en del av startandet av en ny process eller vid mappning av körbar kod till en befintlig process) extraheras och hashas filens kodkatalog. Om det hashvärde som skapas finns i tillförlitlighetscachen får de mappningar av körbar kod som skapas för den binära filen plattformsbehörigheter – det innebär att de innehar samma behörigheter och möjlighet att köra utan ytterligare verifiering av signaturens autenticitet. Det här skiljer sig från Intel-baserade Mac-datorer där plattformsbehörigheter överförs till operativsystemsinnehåll av det Apple-certifikat som signerar binärfilerna. (Det här certifikatet begränsar inte vilka behörigheter binärfilen kan ha.)

Binärfiler som inte är knutna till plattformen (exempelvis attesterad tredjepartskod) måste ha giltiga certifikatkedjor för att köras, och alla behörigheter de kan ha, begränsas av den signeringsprofil som utfärdats till utvecklaren av Apple Developer Program.

Alla binärfiler som levereras med macOS är signerade med en *plattformsidentifierare*. På Mac-datorer med Apple Silicon används den här identifieraren till att indikera att dess kodkataloghash måste finnas i tillförlitlighetscachen för att kunna köras även om binärfilen är signerad av Apple. På Intel-baserade Mac-datorer används plattformsidentifieraren till att utföra riktad återkallning av binärfiler från äldre versioner av macOS. Den här riktade återkallningen förhindrar binärfilerna från att köras i nyare versioner.

Den statiska tillförlitlighetscachen låser fullständigt en uppsättning binärfiler till en viss version av macOS. Det här beteendet förhindrar att legitima Apple-signerade binärfiler från äldre operativsystem introduceras i nyare för att ge en angripare en fördel.

### Plattformskod levererad utanför operativsystemet

Apple levererar vissa binärfiler – t.ex. Xcode och uppsättningen av utvecklarverktyg – som inte är signerade med en plattformsidentifierare. De har ändå behörighet att köra med plattformsbehörighet på Mac-datorer med Apple Silicon och Mac-datorer med T2-krets. Eftersom den här plattformsprogramvaran levereras separat från macOS lyder den inte under det återkallningsbeteende som följer med den statiska tillförlitlighetscachen.

### Inläsningsbara tillförlitlighetscacher

Apple levererar vissa programvarupaket med *inläsningsbara tillförlitlighetscacher*. Dessa cacher har samma datastruktur som den statiska tillförlitlighetscachen. Men även om det bara finns en statisk tillförlitlighetscache – och dess innehåll alltid är garanterat inlåst i skrivskyddade intervall efter att den tidiga initieringen av kärnan har slutförts – läggs inläsningsbara tillförlitlighetscacher till i systemet vid körning.

Dessa tillförlitlighetscacher autentiseras antingen genom samma mekanism som autentiserar den fasta startprogramvaran (anpassning sker med Apples betrodda signeringsprocess) eller som globalt signerade objekt (vars signaturer inte binder dem till en specifik enhet).

Ett exempel på en anpassad tillförlitlighetscache är den cache som levereras med skivavbilden som används till att utföra fältdiagnostik på Mac-datorer med Apple Silicon. Den här tillförlitlighetscachen anpassas, tillsammans med skivavbilden, och läses in i den aktuella Mac-datorns kärna när den startas i diagnosläge. Tillförlitlighetscachen tillåter att programvaran i skivavbilden körs med plattformsbehörighet.

Ett exempel på en globalt signerad tillförlitlighetscache levereras med macOS-programuppdateringar. Den här tillförlitlighetscachen tillåter att en specifik uppsättning kod i programuppdateringar – *uppdateringskärnan* – körs med plattformsbehörighet. Uppdateringskärnan utför allt arbete för att förbereda programuppdateringen som värdsystemet saknar kapacitet att utföra på ett konsekvent sätt mellan olika versioner.

## Säkerhet och perifera processorer i Mac-datorer

Alla moderna datorsystem har många inbyggda perifera processorer som är dedikerade till uppgifter som nätverk, grafik, strömhantering med mera. Dessa perifera processorer har ofta ett enda syfte och är betydligt mindre kraftfulla än den primära processorn. Inbyggd perifer utrustning som saknar tillräcklig säkerhet blir ett enklare mål för angripare och kan utnyttjas till att åstadkomma en bestående infektion i operativsystemet. Efter att ha infekterat den fasta programvaran i en perifer processor kan en angripare angripa programvara i den primära processorn, eller direkt samla in känsliga data (en Ethernetenhet kan t.ex. se innehållet i paket som inte är krypterade).

Apple arbetar för att så långt som möjligt minska antalet perifera processorer som krävs och för att undvika design som kräver fast programvara. När separata processorer med egen fast programvara ändå krävs inriktas arbetet på att säkerställa att angripare inte kan utnyttja de processorerna. Detta kan ske genom verifiering av processorn på ett av två sätt:

- Genom att köra processorn så att den hämtar verifierad fast programvara från den primära processorn vid start
- Genom att låta den perifera processorn använda en egen säker startsekvens för att verifiera den egna fasta programvaran varje gång Mac-datorn startar

Apple samarbetar med leverantörer för att övervaka deras implementeringar och för att förbättra deras utformningar så att de inkluderar önskade egenskaper som:

- Säkerställning av en miniminivå för kryptografisk styrka
- Säkerställning av kraftfull återkallning av fast programvara med kända problem
- Avaktivering av felsökningsgränssnitt
- Signering av den fasta programvaran med kryptografiska nycklar som lagras i Apple-reglerade HSM-moduler (Hardware Security Modules)

Under senare år har Apple samarbetat med ett antal externa leverantörer så att de använder samma Image4-datastrukturer, verifieringskod och signeringsinfrastruktur som används av Apple Silicon.

När varken lagringsfri drift eller lagring i kombination med säker start är ett alternativ kräver designen att uppdateringar av fast programvara är kryptografiskt signerade och verifierade innan det bestående lagringsutrymmet kan uppdateras.

## Rosetta 2 på Mac-datorer med Apple Silicon

Mac-datorer med Apple Silicon kan köra kod som har kompilerats för instruktionsuppsättningen x86_64 genom att använda en översättningsmekanism som kallas *Rosetta 2*. Två typer av översättning erbjuds: just-in-time och ahead-of-time.

### Just-in-time-översättning

I översättningsfunktionen just-in-time (JIT) identifieras ett x86_64 Mach-objekt tidigt i avbildskörningssökvägen. När dessa avbilder hittas överför kärnan styrningen till en speciell Rosetta-översättningsstubb istället för till redigeraren för dynamiska länkar, `dyld(1)`. Översättningsstubben översätter sedan x86_64-sidor medan avbilden körs. Den här översättningen sker helt och hållet inom processen. Kärnan verifierar fortfarande kodhasherna för varje x86_64-sida mot de kodsignaturer som är bifogade till den binärfil som sidan hör till. Om ett hashvärde inte stämmer tvingar kärnan fram den rensningspolicy som är lämplig för den aktuella processen.

### Ahead-of-time-översättning

I översättningssökvägen ahead-of-time (AOT) läses x86_64-binärfiler från lagring vid de tidpunkter som systemet bedömer är optimala för kodens svarshastighet. De översatta artefakterna skrivs till lagring som en speciell typ av Mach-objektfil. Den filen liknar en körbar avbild, men är markerad för att indikera att det är den översatta produkten av en annan avbild.

I den här modellen härleder AOT-artefakten all sin identitetsinformation från den ursprungliga körbara x86_64-avbilden. För att tvinga igenom denna bindning signerar en priviligierad användarutrymmesentitet översättningsartefakten med en enhetsspecifik nyckel som hanteras av Secure Enclave. Denna nyckel släpps endast till den priviligierade användarutrymmesentiteten som identifieras som sådan genom en begränsad behörighet. Den kodkatalog som skapats för översättningsartefakten innehåller kodkataloghashen för den ursprungliga körbara x86_64-avbilden. Själva signaturen på översättningsartefakten kallas för *supplementsignatur*.

I likhet med AOT-funktionen startar JIT-funktionen med att kärnan överför styrningen till Rosetta-körningen istället för redigeraren för dynamiska länkar, `dyld(1)`. Men Rosetta-körningen skickar sedan en IPC-förfrågan till Rosetta-systemtjänsten som frågar om det finns någon AOT-översättning tillgänglig för den aktuella körbara avbilden. Om en sådan hittas tillhandahåller Rosetta-tjänsten ett handtag till den översättningen och den mappas in i processen och körs. Under körning driver kärnan igenom översättningsartefaktens kodkataloghasher som autentiserats av den signatur som lagras i den enhetsspecifika signeringsnyckeln. Den ursprungliga x86_64-avbildens kodkataloghasher är inte inblandade i den här processen.

Översatta artefakter lagras i ett datavalv som inte är tillgängligt vid körning av någon entitet med undantag för Rosetta-tjänsten. Rosetta-tjänsten hanterar åtkomsten till dess cache genom att distribuera skrivskyddade filbeskrivningar till enskilda översättningsartefakter. Detta begränsar åtkomsten till AOT-artefaktens cache. Tjänstens IPC och beroende fotavtryck hålls avsiktligt väldigt smalt för att begränsa angreppsytan.

Om kodkataloghashen för den ursprungliga x86_64-avbilden inte matchar med den som är inkodad i AOT-översättningsartefaktens signatur hanteras resultatet som en ogiltig kodsignatur och lämpliga tillämpningsåtgärder genomförs.

Om en fjärrprocess frågar kärnan efter behörigheter eller andra kodidentitetsegenskaper för en AOT-översatt körbar avbild returneras den ursprungliga x86_64-avbildens identitetsegenskaper till processen.

## Innehåll i statiska tillförlitlighetscacher

macOS 11 och senare levereras med Mach-"fat"-binärfiler som innehåller delar av x86_64- och arm64-datorkod. På Mac-datorer med Apple Silicon kan användaren välja att köra x86_64-delen av en binär systemfil genom Rosetta-funktionen, t.ex. för att läsa in en insticksfil som saknar inbyggd arm64-variant. För att stöda detta innehåller den statiska tillförlitlighetscachen som levereras med macOS normalt tre kodkataloghasher per Mach-objektfil.

- En kodkataloghash för arm64-delen
- En kodkataloghash för x86_64-delen
- En kodkataloghash för AOT-översättningen av x86_64-delen

Rosettas AOT-översättningsprocess är deterministisk på så vis att den reproducerar identiska utmatningar för varje given inmatning, oberoende av när översättningen utfördes eller på vilken enhet den utfördes.

Under macOS-bygget körs varje Mach-objektfil genom Rosettas AOT-översättningskörning som är kopplad till den version av macOS som byggs, och den kodkataloghash som skapas spelas in till tillförlitlighetscachen. Av effektivitetsskäl levereras de faktiska översatta produkterna inte med operativsystemet och omkonstitueras vid behov när användaren begär dem.

När en x86_64-avbild körs på Mac-datorer med Apple Silicon, och avbildens kodkataloghash finns i den statiska tillförlitlighetscachen, förväntas *också* kodkataloghashen för den AOT-artefakt som skapas finnas i den statiska tillförlitlighetscachen. Sådana produkter signeras inte av den enhetsspecifika nyckeln eftersom signeringsutfärdaren lagras i den säkra Apple-startkedjan.

## Osignerad x86_64-kod

Mac-datorer med Apple Silicon tillåter inte att inbyggd arm64-kod körs om giltig bifogad signatur saknas. Den här signaturen kan vara så enkel som en "ad hoc"-kodsignatur (jämför `codesign(1)`) som inte har någon faktisk identitet från den hemliga halvan av ett asymmetriskt nyckelpar (det är helt enkelt en icke-autentiserad mätning av binärfilen).

Av kompatibilitetsskäl kopplade till binärfiler har översatt x86_64-kod tillstånd att köras genom Rosetta utan någon signaturinformation alls. Ingen specifik identitet överförs till den koden genom den enhetsspecifika Secure Enclave-signeringsprocessen, och den utför körningar med exakt samma begränsningar som inbyggd osignerad kod utför körningar på Intel-baserade Mac-datorer.

# DMA-skydd i Mac-datorer

För att uppnå hög genomströmning i gränssnitt med höga hastigheter, som PCIe, FireWire, Thunderbolt och USB, måste datorer ha stöd för DMA (Direct Memory Access) från perifer utrustning. Det innebär att de måste kunna läsa och skriva till RAM-minnet utan kontinuerlig användning av processorn. Sedan 2012 har flera olika tekniker implementerats i Mac-datorer för att skydda mot angrepp via DMA, vilket har lett till den bästa och mest fullständiga uppsättningen DMA-skydd på någon typ av persondator.

## DMA-skydd för Mac-datorer med Apple Silicon

Apples SoC-enheter innehåller en IOMMU (Input/Output Memory Management Unit) för varje DMA-agent i systemet, inklusive PCIe- och Thunderbolt-portar. Eftersom varje IOMMU har en egen uppsättning tabeller för adressöversättning för att översätta DMA-förfrågningar kan kringutrustning som ansluts via PCIe eller Thunderbolt bara komma åt minne som uttryckligen har mappats för deras användning. Kringutrustning kan inte komma åt minne som tillhör andra delar av systemet – som kärnan eller fast programvara – eller minne som är tilldelat till annan kringutrustning. Om en IOMMU upptäcker ett försök av kringutrustning att komma åt minne som inte är mappat för just den kringutrustningen utlöses en kernel panic.

## DMA-skydd för Intel-baserade Mac-datorer

Intel-baserade Mac-datorer med Intel VT-d (Intel Virtualization Technology for Directed I/O) initierar IOMMU, vilket möjliggör DMA-ommappning och avbryter ommappning mycket tidigt i processen för att begränsa olika klasser av säkerhetssvagheter. Apple IOMMU-maskinvaran påbörjar driften med en policy att som förval alltid neka, så i samma ögonblick som systemet slås på börjar det automatiskt att blockera DMA-förfrågningar från kringutrustning. Efter initiering av programvara börjar IOMMU:er att tillåta DMA-förfrågningar till minnesregioner som explicit har mappats för deras användning.

*Obs!* Avbruten ommappning för PCIe krävs inte på Mac-datorer med Apple Silicon eftersom varje IOMMU hanterar MSI:er för sin egen kringutrustning.

Från och med macOS 11 kör alla Mac-datorer med Apple T2-säkerhetskrets UEFI-drivrutiner som förenklar DMA i en begränsad ring 3-miljö när dessa drivrutiner parkopplas med externa enheter. Den här egenskapen gör det enklare att stå emot säkerhetssvagheter som kan inträffa när en skadlig enhet interagerar med en UEFI-drivrutin på ett oväntat sätt vid start. Det minskar särskilt den påverkan svagheterna har på en drivrutins hantering av DMA-buffrar.

# Säker utökning av kärnan i macOS

Om kärntillägg från tredje part är aktiverade kan de inte läsas in i kärnan på begäran från och med macOS 11. De slås istället samman i en *AuxKC (Auxiliary Kernel Collection)* som läses in under startprocessen. För Mac-datorer med Apple Silicon loggas mätvärdet för AuxKC i LocalPolicy (AuxKC fanns på datavolymen för äldre maskinvara). Att bygga om AuxKC kräver användarens godkännande och att macOS startar om för att läsa in ändringarna i kärnan, och det kräver att säker start konfigureras som Minskad säkerhet.

**Viktigt:** Kärntillägg rekommenderas inte längre för macOS. Kärntillägg riskerar operativsystemets integritet och tillförlitlighet och Apple rekommenderar användare att välja lösningar som inte kräver kärntillägg.

## Kärntillägg på Mac-datorer med Apple Silicon

Kärntillägg måste aktiveras särskilt för Mac-datorer med Apple Silicon genom att strömknappen hålls in vid start så att datorn går in i 1TR-läge (One True Recovery). Nedgradera sedan till Minskad säkerhet och markera rutan så att kärntillägg aktiveras. Åtgärden kräver också att ett administratörslösenord anges för att godkänna nedgraderingen. Kombinationen av 1TR och lösenordskrav gör det svårt för programvaruangripare som börjar inuti macOS att överföra kärntillägg till macOS som de sedan kan utnyttja för att få kärnbehörigheter.

När en användare godkänner att kärntillägg läses in används det ovanstående flödet Användargodkänd inläsning av kärntillägg till att godkänna installationen av kärntillägg. Godkännandet som används för ovanstående flöde används också till att samla in en SHA384-hash för UAKL (User Authorized Kext List) i LocalPolicy. Kärnhanteringsdemonen (`kmd`) ansvarar sedan för att validera endast de kärntillägg som finns i UAKL och inkludera dem i AuxKC.

- Om SIP (systemintegritetsskydd) är aktiverat verifieras signaturen för varje kärntillägg innan det läggs till i AuxKC.
- Om SIP är avaktiverat krävs inte kärntilläggssignaturen.

Den här metoden gör det möjligt att använda flöden för Tillåtande säkerhet för utvecklare eller användare som inte deltar i Apple Developer Program när de testar kärntillägg innan de signeras.

När AuxKC är skapad skickas dess mätvärde till Secure Enclave för signering och inkluderas sedan i en Image4-datastruktur som kan utvärderas av iBoot vid start. Ett kärntilläggskvitto genereras också som en del av AuxKC-konstruktionen. Det här kvittot innehåller listan med kärntillägg som faktiskt inkluderades i AuxKC eftersom uppsättningen kan vara en undermängd av UAKL om förbjudna kärntillägg hittades. En SHA384-hash av Image4-datastrukturen i AuxKC och kärntilläggskvittot läggs till i LocalPolicy. Image4-hashen för AuxKC används för extra verifiering av iBoot vid start för att säkerställa att det inte går att starta en äldre Secure Enclave-signerad Image4-fil för AuxKC med en nyare LocalPolicy. Kärntilläggskvittot används av undersystem som Apple Pay till att avgöra om det finns några kärntillägg som läses in för tillfället och som kan minska trovärdigheten hos macOS. Om det finns sådana kärntillägg kan Apple Pay-funktioner avaktiveras.

## Systemtillägg

I macOS 10.15 får utvecklare utöka funktionerna i macOS genom att installera och hantera systemtillägg som körs i användarutrymmet istället för på kärnnivå. Det faktum att systemtilläggen körs i användarutrymmet ökar stabiliteten och säkerheten i macOS. Även om kärntillägg till sin natur har full åtkomst till hela operativsystemet har tillägg som körs i användarutrymmet endast de behörigheter som krävs för att utföra deras avsedda funktion.

Utvecklare kan använda ramverk som DriverKit, EndpointSecurity och NetworkExtension till att skriva drivrutiner för USB- och användargränssnitt, verktyg för slutpunktssäkerhet (förhindra dataförlust eller andra slutpunktsagenter) och VPN- och nätverksverktyg utan att behöva skriva kärntillägg. Säkerhetsagenter från tredje part ska endast användas om de utnyttjar dessa API:er eller om de har en stabil plan för att övergå till dem och bort från kärntillägg.

### Användargodkänd inläsning av kärntillägg

För att förbättra säkerheten krävs godkännande av användaren för att läsa in kärntillägg som har installerats med eller efter installationen av macOS 10.13. Den här processen kallas *User-Approved Kernel Extension Loading*. Auktorisering på administratörsnivå krävs för att godkänna ett kärntillägg. Kärntillägg kräver inte auktorisering om de:

- Installerades på en Mac som körde macOS 10.12 eller tidigare
- Ersätter tidigare godkända tillägg.
- Är tillåtna att läsas in utan användarens godkännande via kommandoradsverktyget `spctl` som är tillgängligt när datorn startats från recoveryOS.
- Är tillåtna att läsas in via en MDM-konfiguration.

  Från och med macOS 10.13.2 kan användarna använda MDM till att ange en lista över kärntillägg som läses in utan att användaren behöver godkänna det. Det här alternativet kräver en Mac med macOS 10.13.2 som är registrerad i MDM via Apple School Manager, Apple Business Manager eller en MDM-registrering som användaren har gjort.

## Option ROM-säkerhet i macOS

*Obs!* Option ROM stöds för närvarande inte på Mac-datorer med Apple Silicon.

### Option ROM-säkerhet i Mac-datorer med Apple T2-säkerhetskrets

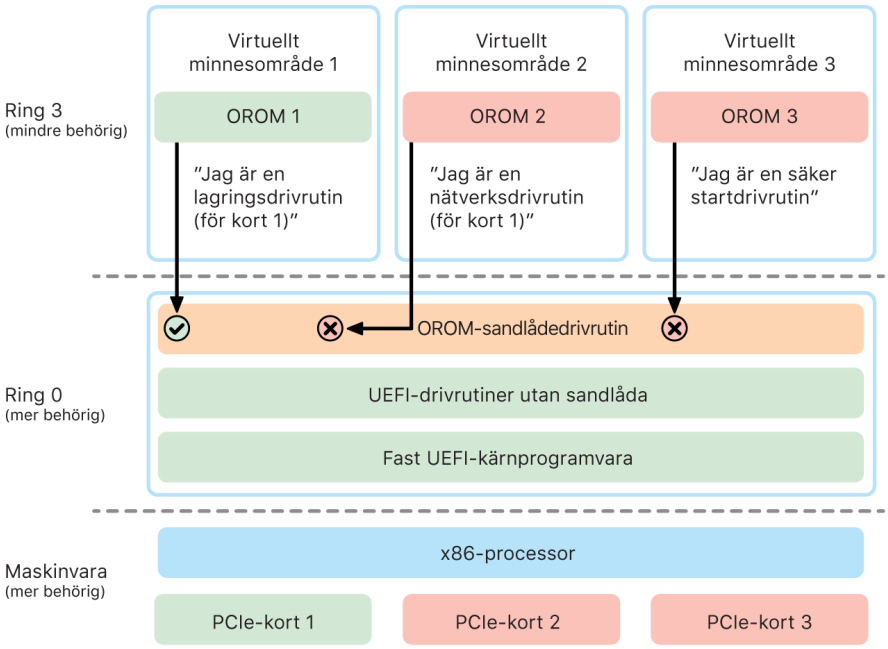
Både Thunderbolt- och PCIe-enheter kan ha en ”Option ROM (OROM)” fysiskt ansluten till enheten. (Detta är normalt inte en äkta ROM utan istället en överskrivningsbar krets som lagrar fast programvara.) I UEFI-baserade system är den fasta programvaran normalt en UEFI-drivrutin som läses och körs av den fasta UEFI-programvaran. Koden som körs har till uppgift att initiera och konfigurera den maskinvara som den hämtades ifrån så att maskinvaran blir tillgänglig för användning av resten av den fasta programvaran. Den här funktionen är nödvändig för att specialiserad maskinvara från tredje part ska kunna läsas in och gå att använda under de allra första faserna av startprocessen, exempelvis för att det ska vara möjligt att starta från externa RAID-uppsättningar.

Eftersom OROM normalt kan skrivas över kan dock en angripare skriva över OROM för legitim kringutrustning så att angriparens kod körs tidigt i startprocessen, vilket gör det möjligt att manipulera körmiljön och bryta integriteten för all programvara som läses in senare. Det är också möjligt för en angripare att köra skadlig kod genom att introducera en egen skadlig enhet i systemet.

I macOS 10.12.3 ändrades beteendet för Mac-datorer sålda efter 2011 så att OROM inte körs som förval när datorn startas, utan bara om användaren trycker ned en speciell tangentkombination. Denna tangentkombination skyddade mot oavsiktlig introduktion av skadlig OROM i startsekvensen för macOS. Det förvalda beteendet för Firmware-lösenordshanterare ändrades också så att OROM inte kunde köras när användaren ställde in ett lösenord för fast programvara, även om användaren tryckte ned tangentkombinationen. Detta skyddade mot att en fysiskt närvarande angripare avsiktligt introducerar skadlig OROM. För användare som fortfarande behöver köra OROM medan ett lösenord för fast programvara är inställt kan ett icke-förvalt alternativ konfigureras genom att använda kommandoradsverktyget `firmwarepasswd` i macOS.

## Säkerhet och OROM-sandlåda

I macOS 10.15 uppdaterades den fasta UEFI-programvaran med en mekanism för sandlåde-OROM och för behörighetssänkning av OROM. Den fasta UEFI-programvaran kör normalt all kod (inklusive OROM) med den högsta behörighetsnivån för processorn, ring 0, och har ett enda delat virtuellt minnesutrymme för all kod och alla data. Ring 0 är den behörighetsnivå där macOS-kärnan körs, medan appar körs på den lägre behörighetsnivån ring 3. OROM-sandlådan minskade behörigheten för OROM genom att använda separering av virtuellt minne på samma sätt som kärnan och sedan se till att OROM körs i ring 3.




Sandlådan begränsar dessutom rejält båda de gränssnitt som OROM kan anropa (ungefär som filtrering av systemanrop i kärnor) och den typ av enhet som en OROM kan registreras som (snarlikt godkännande av appar). Fördelen med den här designen är att skadliga OROM inte längre kan skriva direkt inuti minnet för ring 0. Istället är de begränsade till ett mycket smalt och väldefinierad sandlådegränssnitt. Detta begränsade gränssnitt minskar angreppsytan rejält och tvingar angripare att först ta sig runt sandlådehindret och öka behörigheten.

# Fast UEFI-programvara i Intel-baserade Mac-datorer

I Intel-baserade Mac-datorer med Apple T2-säkerhetskrets bygger säkerheten på fast UEFI (Intel)-maskinvara.

## Översikt

Sedan 2006 använder Mac-datorer med en Intel-baserad processor en fast Intel-programvara baserad på EFI (Extensible Firmware Interface) EDK (Development Kit) version 1 eller version 2. EDK2-baserad kod uppfyller specifikationerna för UEFI (Unified Extensible Firmware Interface). I det här avsnittet används termen *fast UEFI-programvara* för fast Intel-programvara. Den fasta UEFI-programvaran var den första koden som kördes på Intel-kretsen.

För Intel-baserade Mac-datorer utan Apple T2-säkerhetskrets är den fasta UEFI-programvarans betrodda rot den krets där den fasta programvaran lagras. Uppdateringarna av den fasta UEFI-programvaran är digitalt signerade av Apple och verifieras av den fasta programvaran innan uppdatering av lagringsutrymmet sker. För att förhindra nedgraderingsattacker måste uppdateringarna alltid vara en nyare version än den befintliga. En angripare med fysisk tillgång till Mac-datorn kan dock potentiellt använda maskinvara till att ansluta till den fasta programvarans lagringskrets och uppdatera kretsen med skadligt innehåll. Om sårbarheter identifieras i den tidiga startprocessen för den fasta UEFI-programvaran (innan lagringskretsen skrivskyddas) kan även detta leda till bestående infektion i den fasta UEFI-programvaran. Det här är en arkitektonisk maskinvarubegränsning som är vanlig i de flesta Intel-baserade PC-datorer och som finns i alla Intel-baserade Mac-datorer utan T2-krets.

För att förhindra fysiska angrepp som förstör den fasta UEFI-programvaran gjordes arkitekturen i Mac-datorer om så att förtroendet för den fasta UEFI-programvaran byggdes in i T2-kretsen. På dessa Mac-datorer är den fasta UEFI-programvarans betrodda rot specifikt den fasta T2-programvaran på det sätt som beskrivs i Startprocessen för Intel-baserade Mac-datorer.

## Delkomponenten Intel Management Engine (ME)

En delkomponent som lagras i den fasta UEFI-programvaran är den *fasta ME-programvaran (Intel Management Engine)*. ME – en separat processor och ett separat undersystem i Intel-kretsen – används primärt för copyrightskydd för video och data på Mac-datorer som endast har Intel-baserad grafik. För att minska den här delkomponentens angreppsyta kör Intel-baserade Mac-datorer en anpassad fast ME-programvara från vilken de flesta av komponenterna har tagits bort. Eftersom den fasta ME-programvaran för Mac som skapas är mindre än det minsta förvalda bygge som tillhandahålls av Intel finns många av de komponenter som har angripits offentligt av säkerhetsforskare vid tidigare tillfällen inte längre kvar.

## SMM (System Management Mode)

Intel-processorer har ett speciellt körläge som skiljer sig från normal drift. Det kallas *SMM (System Management Mode)* och introducerades ursprungligen för att hantera tidskritiska åtgärder som strömhantering. Mac-datorer använder dock sedan länge en diskret mikrostyrenhet kallad *SMC (System Management Controller)* till att utföra sådana åtgärder. SMC har integrerats i T2-kretsen och är därmed inte längre en separat mikrostyrenhet.

# Systemsäkerhet för watchOS

Apple Watch använder många av samma maskinvarubaserade säkerhetsfunktioner för plattformen som iOS gör. Apple Watch gör till exempel följande:

- Den utför säkra starter och säkra programuppdateringar.
- Den upprätthåller operativsystemets integritet.
- Den skyddar data, både på enheten och vid kommunikation med en parkopplad iPhone eller internet.

Tekniker som stöds inkluderar de som listas i Systemsäkerhet (t.ex. KIP, SKP och SCIP), liksom tekniker för dataskydd, nyckelring och nätverk.

## Uppdatera watchOS

watchOS kan konfigureras för uppdatering under natten. Information om hur lösenkoden till Apple Watch lagras och används under uppdateringen finns i avsnittet Nyckelsamlingar.

## Handledsavkänning

Om handledsavkänning är aktiverat låses enheten automatiskt kort efter att den tas bort från användarens handled. Om handledsavkänning är avaktiverat visas ett alternativ för låsning av Apple Watch i Kontrollcenter. När Apple Watch är låst kan Apple Pay endast användas genom att ange lösenkoden på Apple Watch. Handledsavkänning stängs av med Apple Watch-appen på iPhone. Den här inställningen kan även genomdrivas med en MDM-lösning.

## Aktiveringslås

När Hitta är aktiverat på iPhone kan den Apple Watch som iPhone är parkopplad med använda aktiveringslåset. Aktiveringslåset gör det svårare för någon att använda eller sälja en Apple Watch som har tappats bort eller stulits. Aktiveringslåset kräver användarens Apple-ID och lösenord för att bryta parkopplingen, radera eller återaktivera en Apple Watch.

## Säker parkoppling med iPhone

Apple Watch kan bara parkopplas med en iPhone i taget. När parkopplingen av Apple Watch tas bort skickar iPhone anvisningar för att radera allt innehåll och alla data från klockan.

Parkopplingen av Apple Watch med iPhone säkras med en direkt parkoppling vid utväxling av publika nycklar, följt av den Bluetooth Low Energy (BLE)-länkade delade hemligheten. Apple Watch visar ett animerat mönster som fångas av kameran på iPhone. Mönstret innehåller en kodad hemlighet som används för den direkta parkopplingen via BLE 4.1. En vanlig BLE-inmatning av kodnyckeln används som reservmetod vid parkopplingen om behov uppstår.

När BLE-sessionen har upprättats och krypterats med det högsta säkerhetsprotokollet som finns i Bluetooth-kärnspecifikationen utväxlar iPhone och Apple Watch nycklar på något av följande sätt:

- En process som har anpassats från IDS (Apple Identity Service) enligt beskrivningen i Säkerhet och iMessage i översikt
- Ett utbyte av nycklar med IKEv2/IPsec Det första utbytet av nycklar autentiseras med antingen Bluetooth-sessionsnyckeln (för parkopplingssituationer) eller IDS-nycklarna (för uppdatering av operativsystem). Varje enhet genererar ett nyckelpar med en slumpmässig offentlig och en privat 256-bitars Ed25519-nyckel, och under den första nyckelutbytesprocessen utbyts de offentliga nycklarna. När en Apple Watch först parkopplas i watchOS 10 eller senare grundas de privata nycklarna i dess Secure Enclave.

  På en iPhone med iOS 17 eller senare grundas inte de privata nycklarna i Secure Enclave eftersom en användare som återskapar sin iCloud-säkerhetskopia till samma iPhone bevarar den befintliga Apple Watch-parkopplingen utan att en flytt krävs.

*Obs!* Vilken mekanism som används för nyckelutbyte och kryptering varierar beroende på vilken operativsystemversion iPhone och Apple Watch har. iPhone-enheter med iOS 13 och senare som parkopplas med en Apple Watch med watchOS 6 och senare använder endast IKEv2/IPsec för nyckelutbyte och kryptering.

När nycklar har utväxlats:

- Bluetooth-sessionsnyckeln slängs och all kommunikation mellan iPhone och Apple Watch krypteras med en av metoderna ovan, med de krypterade Bluetooth-, Wi-Fi- och mobillänkarna som ett andra krypteringslager.
- (Endast IKEv2/IPsec) Nycklarna lagras i systemnyckelringen och används till att autentisera framtida IKEv2-/IPsec-sessioner mellan enheterna. Vidare kommunikation mellan de här enheterna krypteras och integriteten skyddas med AES-256-GCM på iPhone-enheter med iOS 15 eller senare som är parkopplade med en Apple Watch Series 4 eller senare med watchOS 8 eller senare. (ChaCha20-Poly1305 med 256-bitarsnycklar används på äldre enheter eller enheter med äldre versioner av operativsystem.)

BLE-enhetsadressen roteras var 15:e minut för att minska risken för att enheten spåras lokalt om någon sänder en bestående identifierare.

Appar som kräver strömmande data stöds genom att kryptering tillhandahålls med metoder som beskrivs under Säkerhet och FaceTime. Metoderna använder antingen Apple IDS-tjänsten (Apple Identity Service) som den parkopplade iPhone-enheten tillhandahåller eller en direkt internetanslutning.

Apple Watch implementerar maskinvarukrypterad lagring och klassbaserat skydd av filer och nyckelringsobjekt. Nyckelsamlingar med behörighetsstyrning för nyckelringsobjekt används också. Nycklar som används för kommunikation mellan Apple Watch och iPhone säkras dessutom med ett klassbaserat skydd. Mer information finns i Nyckelsamlingar för dataskydd.

## Autoupplåsning och Apple Watch

Vissa Apple-enheter kan låsa upp andra automatiskt i vissa situationer. Det gör det smidigare att använda flera Apple-enheter. Autoupplåsning kan användas på tre sätt:

- En Apple Watch kan låsas upp av en iPhone.
- En Mac kan låsas upp av en Apple Watch.
- En iPhone kan låsas upp av en Apple Watch när en användare vars näsa och mun är övertäckt upptäcks.

Alla tre sätt bygger på samma grund: ett gemensamt autentiserat STS-protokoll (Station-to-Station) där långvariga nycklar byts ut när funktionen aktiveras och unika, tillfälliga sessionsnycklar förhandlas för varje begäran. Oavsett vilken underliggande kommunikationskanal som används förhandlas STS-tunneln direkt mellan Secure Enclave i båda enheterna. Allt kryptografiskt material hålls inom den säkra domänen, förutom på Mac-datorer som inte har Secure Enclave. På dem avslutas STS-tunneln i kärnan.

### Upplåsning

En komplett upplåsningssekvens kan delas upp i två faser. Först genererar den enhet som ska låsas upp (”målenheten”) en kryptografisk upplåsningshemlighet och skickar den till enheten som utför upplåsningen (”startenheten”). Sedan utför startenheten upplåsningen med den tidigare genererade hemligheten.

För att kunna låsa upp automatiskt ansluter enheterna till varandra via en BLE-anslutning. En upplåsningshemlighet på 32 byte som målenheten har genererat slumpmässigt skickas sedan till startenheten genom STS-tunneln. Under nästa biometriska upplåsning eller lösenkodsupplåsning paketerar målenheten den lösenkodshärledda nyckeln (PDK) och upplåsningshemligheten och tar bort upplåsningshemligheten från minnet.

För att kunna utföra upplåsningen upprättar enheterna en ny BLE-anslutning och använder sedan P2P-Wi-Fi till att säkert bestämma avståndet mellan dem. Om enheterna är inom det angivna intervallet och de säkerhetspolicyer som krävs uppfylls skickar startenheten sin upplåsningshemlighet till målenheten genom STS-tunneln. Målenheten genererar sedan en ny upplåsningshemlighet på 32 byte och skickar tillbaka den till startenheten. Om den nuvarande upplåsningshemligheten som startenheten skickade kan avkryptera upplåsningsregistret låses målenheten upp och PDK paketeras om med en ny upplåsningshemlighet. Till sist tas den nya upplåsningshemligheten och PDK bort från målenhetens minne.

### Säkerhetspolicyer för autoupplåsning av Apple Watch

För att det ska gå smidigare kan Apple Watch låsas upp av en iPhone direkt efter den har startats första gången. Användaren behöver inte ange lösenkoden på själva Apple Watch först. För att det ska vara möjligt används den slumpmässiga upplåsningshemligheten (som genereras under den allra första upplåsningssekvensen efter att funktionen har aktiverats) till att skapa en långvarig deponeringspost som lagras i nyckelsamlingen på Apple Watch. Hemligheten i deponeringsposten lagras i iPhone-nyckelringen och används till att starta en ny session varje gång Apple Watch har startats om.

### Säkerhetspolicyer för autoupplåsning av iPhone

Ytterligare säkerhetspolicyer gäller för autoupplåsning av iPhone med Apple Watch. Apple Watch kan inte användas i stället för Face ID på iPhone till andra åtgärder som Apple Pay eller appgodkännanden. När Apple Watch låser upp en parkopplad iPhone visas en notis på klockan och en tillhörande haptisk notis utförs. Om användaren trycker på knappen Lås iPhone i notisen skickar klockan ett låskommando till iPhone via BLE. När iPhone får låskommandot låses den och både Face ID och upplåsning med Apple Watch avaktiveras. Nästa upplåsning av iPhone måste göras med iPhone-lösenkoden.

Följande villkor måste uppfyllas för att det ska gå att låsa upp en parkopplad iPhone via Apple Watch (när det är aktiverat):

- iPhone måste ha blivit upplåst med en annan metod minst en gång efter att den kopplade Apple Watch-enheten har placerats på handleden och låsts upp.
- Sensorerna måste kunna känna av att näsan och munnen är övertäckta.
- Det uppmätta avståndet måste vara högst 3 meter.
- Apple Watch får inte vara i läggdagsläget.
- Apple Watch eller iPhone måste ha låsts upp nyligen, eller så måste Apple Watch ha rört sig så att det är tydligt att bäraren är aktiv (inte sover eller något liknande).
- iPhone måste ha låsts upp minst en gång under de senaste 6,5 timmarna.
- iPhone måste vara i ett läge där det är tillåtet att låsa upp enheten med Face ID. (Mer information finns i Face ID, Touch ID, lösenkoder och lösenord.)

## Godkänna i macOS med Apple Watch

När Autoupplåsning med Apple Watch är aktiverat kan Apple Watch användas istället för eller tillsammans med Touch ID för att godkänna auktorisering och autentiseringsförfrågningar från:

- macOS- och Apple-appar som begär auktorisering
- Tredjepartsappar som begär auktorisering
- Sparade Safari-lösenord
- Säkra anteckningar

## Säker användning av Wi-Fi, mobildata, iCloud och Gmail

Om Apple Watch inte är inom Bluetooth-räckvidd går det att använda Wi-Fi eller mobilnät istället. Apple Watch ansluter automatiskt till Wi-Fi-nätverk som dess parkopplade iPhone redan har anslutit till och vars inloggningsuppgifter har synkroniserats till Apple Watch medan båda enheterna var inom räckvidden. Det här automatiska anslutningsbeteendet kan konfigureras på enskild nätverksnivå under Wi-Fi i appen Inställningar på Apple Watch. Wi-Fi-nätverk som ingen av enheterna har anslutit till tidigare kan anslutas manuellt under Wi-Fi i appen Inställningar på Apple Watch.

När Apple Watch och iPhone är utom räckvidd ansluter Apple Watch direkt till iCloud- och Gmail-servrar för att hämta mejl i Mail istället för att synkronisera Mail-data med dess parkopplade iPhone via internet. För Gmail-konton måste användaren autentisera till Google i avsnittet Mail i Apple Watch-appen på iPhone. Den OAuth-token som fås från Google skickas över till Apple Watch i ett krypterat format via IDS (Apple Identity Service) så att den kan användas till att hämta e-post. Denna OAuth-token används aldrig för anslutning till Gmail-servern från den iPhone som är parkopplad.

# Slumptalsgenerering

Kryptografisk pseudoslumptalsgenerering (CPRNG) är en viktig byggsten för säker programvara. Apple tillhandahåller därför en betrodd programvaru-CPRNG som körs i kärnan i iOS, iPadOS, macOS, tvOS och watchOS. Den ansvarar för att slå samman rå entropi från systemet och tillhandahålla säkra slumptal för användning både i kärnan och i användarutrymmet.

## Entropikällor

Kärnans CPRNG seedas från flera entropikällor både vid start och under enhetens livslängd. Dessa inkluderar (beroende på tillgänglighet):

- Maskinvaru-TRNG för Secure Enclave
- Timingbaserad jitterinsamling under start
- Entropi insamlad från maskinvaruavbrott
- En seed-fil för att upprätthålla entropi mellan starter
- Intel-slumpinstruktioner – t.ex. RDSEED och RDRAND (endast på Intel-baserade Mac-datorer)

## CPRNG för kärnan

CPRNG för kärnan är en Fortuna-härledd design som är avsedd för en 256-bitars säkerhetsnivå. Den tillhandahåller högkvalitativa slumptal för användning i användarutrymmet via följande API:er:

- Systemanropet `getentropy`(2)
- Slumpenheten (/dev/random)

CPRNG för kärnan accepterar entropi som är tillhandahållen av användaren genom skrivningar till slumpenheten.

# Apple Security Research Device

Apple Security Research Device är en specialbyggd iPhone som tillåter säkerhetsforskare att utföra forskning på iOS utan att behöva ta sig runt eller avaktivera funktionerna för plattformssäkerhet på iPhone. Med den här enheten kan en forskare läsa in innehåll som sedan körs med plattformsmotsvarande behörigheter, och på så vis utföra forskning på en plattform som i mycket hög grad liknar plattformen på produktionsenheter.

För att säkerställa att användarenheterna inte påverkas av körningspolicyn under säkerhetsforskning implementeras policyändringarna i en variant av iBoot och i startkärnsamlingen. Dessa startas inte på användarmaskinvara. Forsknings-iBoot kontrollerar vilken typ av enhet den körs på och utlöser en panic-loop om den körs på maskinvara som inte är specialbyggd för forskning.

Cryptex-delsystemet tillåter forskare att läsa in en personligt anpassad tillförlitlighetscache och en skivavbild som innehåller motsvarande innehåll. Ett antal djupgående skyddsåtgärder har implementerats. De ska säkerställa att detta delsystem inte tillåter körning på användarenheter:

- `launchd` läser inte in launchd-egenskapslistan `cryptexd` ifall den upptäcker en vanlig kundenhet.
- `cryptexd` stoppas ifall den upptäcker en vanlig kundenhet.
- `AppleImage4` godkänner inte anti-replay-värdet som används till att verifiera en forsknings-cryptex på en vanlig kundenhet.
- Signeringsservern vägrar att skapa en personligt anpassad cryptex-skivavbild för en enhet som inte finns på en lista över enheter med explicit tillstånd.

Säkerhetsforskarens integritet respekteras genom att endast mätvärdena (t.ex. hashvärden) för de körbara koderna eller kärncachen och identifierarna för enheter avsedda för säkerhetsforskning skickas till Apple under anpassningen. Apple tar inte emot något innehåll från den cryptex som läses in på enheten.

För att undvika att en forskningsenhet maskeras som en användarenhet i skadligt syfte, för att lura ett mål att använda den som en vanlig enhet, finns följande speciella egenskaper inbyggda i enheter avsedda för säkerhetsforskning:

- Enheter avsedda för säkerhetsforskning kan endast startas medan de laddas. Detta kan vara via en Lightning-kabel eller en Qi-kompatibel laddare. Om enheten inte laddas under start försätts den i återställningsläge. Om användaren börjar ladda och startar om enheten startar den som vanligt. När XNU har startat behöver inte enheten laddas för att fortsätta.
- Orden *Security Research Device* visas under Apples logotyp under iBoot-start.
- XNU-kärnan startar i utförligt läge.
- Enheten är märkt på sidan med följande inetsade meddelande: "Property of Apple. Confidential and Proprietary. Call +1 877 595 1125."

Följande är ytterligare steg som implementeras i programvara som visas efter start:

- Orden *Security Research Device* visas när enheten startas.
- Orden *Security Research Device* visas på låsskärmen och i appen Inställningar.

Enheter avsedda för säkerhetsforskning tillåter följande som en användarenhet inte tillåter. Forskare kan:

- Läsa in körbar kod på enheten med slumpmässig behörighet på samma behörighetsnivå som Apple-operativsystemkomponenter.
- Starta tjänster vid systemstart.
- Behålla beständigt innehåll mellan omstarter.
- Använda behörigheten `research.com.apple.license-to-operate` som tillåter en process som felsöker alla andra processer i systemet, inklusive systemprocesser.

  `research.`-namnrymden respekteras endast av varianten RESEARCH i AppleMobileFileIntegrity-kärntillägget. Alla processer med den här behörigheten avslutas på en kundenhet under signaturvalideringen.
- Personanpassa och återskapa en anpassad kärncache.

# Kryptering och dataskydd

## Kryptering och dataskydd i översikt

Funktionerna för säker startsekvens, systemsäkerhet och appsäkerhet verifierar alla tillsammans att endast betrodd kod och betrodda appar körs på en enhet. Apple-enheter har dessutom flera krypteringsfunktioner som skyddar användardata, även när säkerheten i andra delar av säkerhetsinfrastrukturen har komprometterats (t.ex. om en enhet har tappats bort eller kör obetrodd kod). Alla dessa funktioner innebär viktiga fördelar för både användare och IT-administratörer eftersom både personlig information och företagsdata skyddas och det finns metoder för snabb och fullständig fjärradering av stulna eller borttappade enheter.

På iPhone- och iPad-enheter används en filkrypteringsmetod som kallas *dataskydd*, medan data på Intel-baserade Mac-datorer skyddas med en volymkrypteringsteknik som kallas *FileVault*. En Mac med Apple Silicon använder en hybridmodell som stöder dataskydd med två försiktighetsåtgärder: den lägsta skyddsnivåklassen (D) stöds inte, och den förvalda nivån (klass C) använder en volymnyckel och fungerar precis som FileVault på en Intel-baserad Mac. I samtliga fall finns centrala nyckelhanteringshierarkier i den dedikerade kretsen för Secure Enclave (på enheter med en Secure Enclave) och en dedikerad AES-motor ger stöd för line-speed-kryptering som säkerställer att långlivade krypteringsnycklar aldrig behöver exponeras mot operativsystemets kärna eller processorn (där de kan komprometteras). (Intel-baserade Mac-datorer med T1-krets eller som saknar Secure Enclave använder inte någon dedikerad krets till att skydda FileVault-krypteringsnycklar.)

Utöver att använda dataskydd och FileVault till att förhindra obehörig åtkomst till data använder Apple *operativsystemskärnor* till att genomdriva skydd och säkerhet. Kärnan använder åtkomstkontroller till att placera appar i sandlådor (vilket begränsar vilka data en app har tillgång till) och en mekanism som kallas *datavalv* (som istället för att begränsa de anrop som en app kan göra begränsar åtkomsten till data från en enskild app när andra appar skickar en förfrågan).

# Lösenkoder och lösenord

För att skydda användardata från skadliga attacker använder Apple lösenkoder i iOS och iPadOS och lösenord i macOS. Ju längre en lösenkod eller ett lösenord är, desto starkare är det och desto enklare är det att avvärja automatiserade intrångsförsök. För att ytterligare avvärja attacker genomdriver Apple tidsfördröjningar (för iOS och iPadOS) och ett begränsat antal lösenordsförsök (för macOS).

När en användare ställer in en lösenkod eller ett lösenord för en enhet i iOS och iPadOS aktiveras dataskydd automatiskt. Dataskydd aktiveras också på andra enheter som har ett Apple-SoC, exempelvis en Mac med Apple Silicon, Apple TV och Apple Watch. I macOS använder Apple det inbyggda volymkrypteringsprogrammet *FileVault*.

## Så här ökar lösenkoder och lösenord säkerheten

iOS och iPadOS stöder lösenkoder med sex eller fyra siffror och alfanumeriska lösenkoder med valfri längd. Förutom att låsa enheten tillhandahåller lösenkoden eller lösenordet entropi för vissa krypteringsnycklar. Det innebär att en angripare som har enheten i sin ägo inte kan komma åt data i specifika skyddsklasser utan lösenkoden.

Lösenkoderna eller lösenorden är knutna till enheternas UID:n, så automatiserade intrångsförsök måste utföras på själva enheterna. Ett högt antal beräkningsiterationer gör att varje försök tar lång tid. Iterationsantalet har kalibrerats så att ett försök tar ungefär 80 millisekunder. Det skulle faktiskt ta mer än fem och ett halvt år att testa alla kombinationer av en sex tecken lång alfanumerisk lösenkod med små bokstäver och siffror.

Ju starkare lösenkod användaren har, desto starkare blir krypteringsnyckeln. Och genom att använda Face ID och Touch ID kan användaren skapa en mycket starkare lösenkod än vad som annars skulle vara praktiskt. Den starkare lösenkoden ökar den effektiva mängden entropi som skyddar krypteringsnycklarna för dataskydd utan att inverka negativt på användarupplevelsen av att låsa upp en enhet många gånger varje dag.

Om du anger ett långt lösenord som bara innehåller siffror visas ett numeriskt tangentbord på låsskärmen istället för det fullständiga tangentbordet. En längre numerisk lösenkod kan vara enklare att ange än en kortare alfanumerisk och ändå ge samma säkerhet.

Användaren kan ange längre alfanumeriska lösenkoder genom att öppna Inställningar > Touch ID och lösenkod eller Face ID och lösenkod, trycka på Lösenkodsalternativ och välja Alfanumerisk kod.

## Hur ökande tidsfördröjningar avvärjer automatiserade försök att knäcka lösenord

För att ytterligare avvärja automatiserade försök att knäcka lösenkoder i iOS, iPadOS och macOS ökar fördröjningen när en felaktig lösenkod, lösenord eller PIN-kod anges (beroende på enhet och i vilket läge enheten befinner sig) enligt tabellen nedan.

| Försök | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 eller fler |
|---|---|---|---|---|---|---|---|---|
| iOS- och iPadOS-låsskärm | Ingen | 1 minut | 5 minuter | 15 minuter | 1 timme | 3 timmar | 8 timmar | Enheten är avaktiverad och måste ansluta till en Mac eller PC |
| watchOS-låsskärm | Ingen | 1 minut | 5 minuter | 15 minuter | 1 timme | 3 timmar | 8 timmar | Enheten är avaktiverad och måste ansluta till en iPhone |
| macOS-inloggningsfönster och -låsskärm | Ingen | 1 minut | 5 minuter | 15 minuter | 1 timme | 3 timmar | 8 timmar | 8 timmar |
| macOS-återställningsläge | Ingen | 1 minut | 5 minuter | 15 minuter | 1 timme | 3 timmar | 8 timmar | Se ”Hur ökande tidsfördröjningar avvärjer automatiserade försök att knäcka lösenord i macOS” nedan |
| FileVault med återställningsnyckel (personlig, organisationens eller iCloud) | Ingen | 1 minut | 5 minuter | 15 minuter | 1 timme | 3 timmar | 8 timmar | Se ”Hur ökande tidsfördröjningar avvärjer automatiserade försök att knäcka lösenord i macOS” nedan |
| macOS-fjärrlåsning med PIN-kod | 1 minut | 5 minuter | 15 minuter | 30 minuter | 1 timme | 1 timme | 1 timme | 1 timme |

Om alternativet Radera data är aktiverat för iPhone eller iPad (i Inställningar > [Face ID] eller [Touch ID] och lösenkod) tas allt innehåll och alla inställningar bort från lagringsutrymmet efter tio på varandra följande felaktiga försök att ange lösenkoden. Flera på varandra följande försök att använda samma felaktiga lösenkod räknas inte i den gränsen. Den här inställningen kan användas som en administrativ policy via en MDM-lösning som stöder funktionen och via Microsoft Exchange ActiveSync, och det går även att ange ett lägre tröskelvärde.

På enheter med Secure Enclave framtvingas fördröjningarna av Secure Enclave. Fördröjningen gäller även om enheten startas om under en fördröjning. Timern startas om för den aktuella perioden.

## Hur ökande tidsfördröjningar avvärjer automatiserade försök att knäcka lösenord i macOS

För att förhindra automatiserade intrångsförsök när Mac-datorn startar tillåts inte fler än 10 lösenordsförsök i inloggningsfönstret, och stigande tidsfördröjningar läggs till efter ett visst antal felaktiga försök. Fördröjningarna genomdrivs av Secure Enclave. Fördröjningen gäller även om datorn startas om under en fördröjning. Timern startas om för den aktuella perioden.

För att förhindra att sabotageprogram orsakar permanent dataförlust genom att försöka angripa användarens lösenord drivs de här gränserna inte igenom efter att användaren har lyckats logga in på datorn, utan de gör det efter omstart. Om de 10 försöken används upp blir ytterligare 10 försök möjliga efter omstart till recoveryOS. Om de också används upp blir ytterligare 10 försök tillgängliga för varje FileVault-återställningsmekanism (iCloud-återställning, FileVault-återställningsnyckel och organisationsnyckel) för upp till högst 30 ytterligare försök. När även dessa ytterligare försök har använts upp bearbetar Secure Enclave inte längre någon begäran om att avkryptera eller verifiera lösenord, utan alla data på hårddisken blir oåterkalleligt otillgängliga.

För att bidra till att skydda data i en företagsmiljö bör IT-ansvariga tydligt definiera och genomdriva FileVault-konfigurationspolicyer via en MDM-lösning. Organisationer har flera alternativ för hantering av krypterade volymer som omfattar institutionella återställningsnycklar, personliga återställningsnycklar (som även kan lagras med MDM för deponering) eller en kombination av båda. Nyckelrotation kan också ställas in som en policy i MDM.

På Mac-datorer med Apple T2-säkerhetskretsen har lösenordet en snarlik funktion, med undantag för att den nyckel som genereras används till FileVault-kryptering istället för dataskydd. macOS innehåller också fler alternativ för lösenordsåterställning:

- iCloud-återställning
- FileVault-återställning
- FileVault-organisationsnyckel

# Dataskydd

## Dataskydd i översikt

Apple använder en teknik som kallas dataskydd till att skydda data som lagras i enhetens flashminne på enheter med en av Apples SoC:er, som iPhone, iPad, Apple Watch, Apple TV och Mac-datorer med Apple Silicon. Med dataskyddet kan en enhet svara på vanliga händelser, som inkommande telefonsamtal, samtidigt som man tillhandahåller en hög nivå när det gäller kryptering av användardata. Några systemappar (som Meddelanden, Mail, Kalender, Kontakter och Bilder) och datavärden för Hälsa använder dataskydd som förval. Tredjepartsappar får automatiskt detta skydd.

### Implementering

Dataskyddet implementeras genom att skapa och hantera en hierarki av nycklar. Det bygger på den teknik för maskinvarukryptering som finns inbyggd i Apple-enheter. Dataskyddet styrs på filnivå genom att tilldela varje fil till en klass. Tillgängligheten avgörs sedan baserat på om klassnycklarna har låsts upp eller ej. Med APFS (Apple File System) kan filsystemet dela nycklarna i mindre delar på extentgrund (där delar av en fil kan ha olika nycklar).

Varje gång en fil skapas på datavolymen skapar dataskyddet en ny 256-bitarsnyckel (*filnyckeln*) och överlämnar den till maskinvarans AES-motor som använder nyckeln till att kryptera filen när den sparas i flashminnet. På enheter med A14 till A17 och M1 till M3 använder krypteringen AES-256 i XTS-läge där 256-bitarsfilnyckeln genomgår en nyckelhärledningsfunktion (NIST Special Publication 800-108) för att härleda en 256-bitars förvrängningsnyckel och en 256-bitars kodnyckel. På enheter med A9 till A13 och S5 till S9 använder krypteringen AES-128 i XTS-läge där 256-bitarsfilnyckeln delas upp för att tillhandahålla en 128-bitars förvrängningsnyckel och en 128-bitars kodnyckel.

På Mac-datorer med Apple Silicon är förvalet för dataskydd klass C (se Dataskyddsklasser), men en volymnyckel används istället för en extentnyckel eller filnyckel – vilket i praktiken återskapar FileVault-säkerhetsmodellen för användardata. Användarna måste fortfarande välja FileVault för att få det kraftigare skydd som att knyta krypteringsnyckelhierarkin till ett lösenord innebär. Utvecklare kan också välja en högre skyddsklass som använder filnycklar eller extentnycklar.

## Dataskydd i Apple-enheter

På Apple-enheter med dataskydd skyddas varje fil med en unik filnyckel (eller filextentnyckel). Nyckeln – paketerad med nyckelpaketeringsalgoritmen NIST AED – paketeras ytterligare en gång med någon av klassnycklarna, beroende på hur filen ska vara tillgänglig. Den paketerade filnyckeln sparas i filens metadata.

Enheter med APFS-format kan stöda filkloning (enkla kopior med copy-on-write-teknik). Om en fil klonas får varje halva av klonen en ny nyckel för att godkänna inkommande skrivförfrågningar så att nya data skrivs i mediet med en ny nyckel. Över tid kan filen komma att bestå av olika extent (fragment) som är mappade till olika nycklar. Alla extent som utgör en fil vaktas dock av samma klassnyckel.

När en fil öppnas avkrypteras dess metadata med filsystemnyckeln, vilket öppnar nyckelpaketet och en notering om vilken klass som skyddar filen. Filnyckeln (eller filextentnyckeln) öppnas tillsammans med klassnyckeln och skickas sedan till maskinvarans AES-motor som avkrypterar filen när den läses från flashminnet. All hantering av paketerade filnycklar sker i Secure Enclave. Filnyckeln hanteras aldrig direkt av approcessorn. Vid start förhandlar Secure Enclave fram en tillfällig nyckel med AES-motorn. När Secure Enclave öppnar filens nycklar paketeras de om med den tillfälliga nyckeln och skickas tillbaka till approcessorn.

Metadata för alla filer i datavolymens filsystem krypteras med en slumpmässig volymnyckel som skapas när operativsystemet installeras första gången eller när enheten raderas av en användare. Den här nyckeln krypteras och paketeras för långtidslagring av en nyckelpaketeringsnyckel som endast Secure Enclave känner till. Nyckelpaketeringsnyckeln ändras varje gång en användare raderar sin enhet. På A9 (och senare) med SoC använder Secure Enclave entropi, förstärkt med anti-replay-system, till att uppnå raderingsbarhet och för att skydda sin nyckelpaketeringsnyckel och andra tillgångar. Mer information finns i Säker icke-flyktig lagring.

I likhet med filnycklar eller filextentnycklar är datavolymens metadatanyckel aldrig exponerad mot approcessorn eftersom Secure Enclave tillhandahåller en tillfällig version varje gång enheten startas. Vid lagring är den krypterade systemfilsnyckeln dessutom paketerad med en tillfällig nyckel som lagras i det raderingsbara lagringsutrymmet eller med en medienyckelpaketeringsnyckel som skyddas av Secure Enclaves anti-replay-mekanism. Den här nyckeln ger inget extra dataskydd. Den är istället utformad för att kunna raderas snabbt (antingen av användaren med hjälp av alternativet Radera allt innehåll och inst. eller av en användare eller administratör med hjälp av ett fjärraderingskommando från en MDM-lösning, Microsoft Exchange ActiveSync eller iCloud). Om du tar bort nyckeln på det sättet blir alla filer kryptografiskt oåtkomliga.

Innehållet i en fil kan krypteras med en eller flera filnycklar (eller filextentnycklar) som sparas tillsammans med klassnyckeln i filens metadata, vilka i sin tur krypteras med filsystemnyckeln. Klassnyckeln skyddas av maskinvarans UID och, för vissa klasser, av användarens lösenkod. Den här hierarkin ger både flexibilitet och prestanda. Att till exempel ändra klass för en fil kräver bara att filnyckeln ompaketeras, och ett byte av lösenkod ompaketerar bara klassnyckeln.

## Dataskyddsklasser

När en ny fil skapas på enheter med stöd för dataskydd tilldelas den en klass av den app som skapar den. Varje klass har olika policyer för att avgöra när informationen i filen är tillgänglig. De grundläggande klasserna och policyerna beskrivs nedan. Apple Silicon-baserade Mac-datorer stöder inte Klass D: No Protection och en säkerhetsgräns upprättas runt in- och utloggning (inte låsning eller upplåsning som på iPhone och iPad).

| Klass | Skyddstyp |
| --- | --- |
| Klass A: Complete Protection | NSFileProtectionComplete |
| Klass B: Protected Unless Open | NSFileProtectionCompleteUnlessOpen |
| Klass C: Protected Until First User Authentication<br>*Obs!* macOS använder en volymnyckel till att återskapa egenskaperna för FileVault-skydd. | NSFileProtectionCompleteUntilFirstUserAuthentication |
| Klass D No Protection<br>*Obs!* Stöds inte i macOS. | NSFileProtectionNone |

### Complete Protection

*NSFileProtectionComplete:* Klassnyckeln skyddas av en nyckel som härleds ur användarens lösenkod eller lösenord och enhetens UID. Strax efter att användaren har låst enheten (tio sekunder om inställningen Kräv lösenkod är Direkt) kastas den avkrypterade klassnyckeln. Det innebär att inga data i klassen går att komma åt förrän användaren anger lösenkoden igen eller låser upp (loggar in på) enheten med Face ID eller Touch ID.

Strax efter att den sista användaren loggat ut från macOS kastas den avkrypterade klassnyckeln. Det innebär att inga data i klassen går att komma åt förrän en användare anger lösenkoden igen eller loggar in på enheten med Touch ID eller Face ID.

### Protected Unless Open

*NSFileProtectionCompleteUnlessOpen:* En del filer kanske måste skrivas medan enheten är låst eller när användaren inte är inloggad. Ett bra exempel är en e-postbilaga som hämtas i bakgrunden. Detta beteende uppnås genom att använda asymmetrisk kryptering med elliptiska kurvor (ECDH över Curve25519). Den vanliga filnyckeln skyddas av en nyckel som härleds genom One-Pass Diffie-Hellman Key Agreement enligt beskrivningen i NIST SP 800-56A.

Den tillfälliga publika nyckeln för överenskommelsen lagras tillsammans med den paketerade filnyckeln. KDF är Concatenation Key Derivation Function (Approved Alternative 1) enligt beskrivningen i 5.8.1 i NIST SP 800-56A. AlgorithmID utelämnas. PartyUInfo och PartyVInfo är den tillfälliga respektive den statiska publika nyckeln. SHA256 används som hashfunktion. Så fort filen stängs raderas filnyckeln från minnet. När filen ska öppnas igen återskapas den delade hemligheten med hjälp av den privata nyckeln till klassen Protected Unless Open och filens tillfälliga publika nyckel. Dessa används till att packa upp filnyckeln som i sin tur används till att avkryptera filen.

I macOS är den privata delen av NSFileProtectionCompleteUnlessOpen åtkomlig så länge någon användare på systemet är inloggad eller har autentiserat sig.

### Protected Until First User Authentication

*NSFileProtectionCompleteUntilFirstUserAuthentication:* Den här klassen uppträder på samma sätt som Complete Protection, förutom att den avkrypterade klassnyckeln inte raderas från minnet när enheten låses eller användaren loggar ut. Skyddet i den här klassen har egenskaper som liknar kryptering av hela enheter på en dator och skyddar mot angrepp som innefattar en omstart. Det här är den förvalda klassen för alla data som tillhör tredjepartsappar och inte har tilldelats någon annan dataskyddsklass.

I macOS använder den här klassen en volymnyckel som är tillgänglig så länge som volymen är inlänkad och fungerar precis som FileVault.

### No Protection

*NSFileProtectionNone:* Den här klassnyckeln skyddas endast med UID:t och förvaras i det raderingsbara lagringsutrymmet. Eftersom alla nycklar som behövs för att avkryptera filer i den här klassen förvaras på enheten ger kryptering bara den fördelen att fjärradering går snabbt. Om en fil inte har tilldelats någon dataskyddsklass sparas den ändå i krypterad form (precis som alla andra data på en iOS- och iPadOS-enhet).

Detta stöds inte i macOS.

*Obs!* För volymer i macOS som inte motsvarar ett startat operativsystem är alla dataskyddsklasser åtkomliga så länge volymen är inlänkad. Den förvalda dataskyddsklassen är NSFileProtectionCompleteUntilFirstUserAuthentication. Filextentnycklar fungerar för både Rosetta 2 och inbyggda appar.

## Nyckelsamlingar för dataskydd

I iOS, iPadOS, tvOS och watchOS samlas nycklarna till både filens och nyckelringens dataskyddsklasser in och hanteras i nyckelsamlingar. Dessa operativsystem använder följande nyckelsamlingar: användare, enhet, säkerhetskopia, deponerad och iCloud-säkerhetskopia.

### Användarnyckelsamling

Användarnyckelsamlingen är den nyckelsamling där de paketerade klassnycklarna som används vid normal körning av enheten förvaras. När användaren exempelvis anger en lösenkod läses *NSFileProtectionComplete* in från användarnyckelsamlingen och packas upp. Det är en binär egenskapslistfil (.plist) som lagras i klassen No Protection.

För enheter med SoC äldre än A9 är innehållet i .plist-filen krypterat med en nyckel som finns i ett raderingsbart lagringsutrymme. För att skydda nyckelsamlingarna raderas och omskapas den här nyckeln varje gång en användare ändrar sin lösenkod.

För enheter med A9 eller senare SoC:er innehåller .plist-filen en nyckel som indikerar att nyckelsamlingen lagras i ett utrymme som skyddas av det Secure Enclave-styrda anti-replay-värdet.

Secure Enclave hanterar användarnyckelsamlingen och kan svara på sökningar angående enhetens låsningsstatus. Det rapporterar att enheten är upplåst endast om alla klassnycklarna i användarnyckelsamlingen är tillgängliga och har packats upp.

## Enhetsnyckelsamling

Enhetsnyckelsamlingen används till att förvara de paketerade klassnycklarna som används vid åtgärder som innehåller enhetsspecifika data. iPadOS-enheter som har konfigurerats för delad användning behöver ibland få tillgång till inloggningsuppgifter innan en användare har loggat in. Därför krävs det en nyckelsamling som inte är skyddad med användarens lösenkod.

iOS och iPadOS har inte stöd för kryptografisk uppdelning av filsystemsinnehåll per användare, vilket betyder att systemet använder klassnycklar från enhetsnyckelsamlingen till att paketera filnycklar. Nyckelringen använder däremot klassnycklar från användarnyckelsamlingen till att skydda objekt i användarens nyckelring. På iPhone-och iPad-enheter som har konfigurerats för att användas av endast en användare (den förvalda konfigurationen) är enhetsnyckelsamlingen densamma som användarnyckelsamlingen och båda skyddas av användarens lösenkod.

## Nyckelsamlingen för säkerhetskopiering

Nyckelsamlingen för säkerhetskopiering skapas när en krypterad säkerhetskopia görs i Finder (i macOS 10.15 eller senare) eller i iTunes (macOS 10.14 eller tidigare) och sparas på den dator som enheten säkerhetskopieras till. I samband med detta skapas en ny nyckelsamling med en ny uppsättning nycklar och säkerhetskopierade data krypteras om med de nya nycklarna. Som beskrivits tidigare förblir icke-flyttbara nyckelringsobjekt paketerade tillsammans med den UID-härledda nyckeln. Det innebär att de kan återskapas till den enhet som de ursprungligen säkerhetskopierades från, men inte är tillgängliga på andra enheter.

Nyckelsamlingen – skyddad med det inställda lösenordet – kör genom 10 miljoner iterationer av nyckelhärledningsfunktion PBKDF2. Trots det höga iterationsantalet finns det ingen koppling till en specifik enhet, och därför skulle ett automatiserat intrångsförsök som körs parallellt på många datorer teoretiskt sett kunna utföras på nyckelsamlingen för säkerhetskopiering. Den här typen av hot kan undvikas med ett tillräckligt starkt lösenord.

Om användaren väljer att inte kryptera säkerhetskopian krypteras inte några av filerna, oavsett deras dataskyddsklass, men nyckelringen skyddas fortfarande med en UID-härledd nyckel. Det är därför nyckelringsobjekt bara flyttas till en ny enhet om användaren har ställt in ett lösenord för säkerhetskopiering.

## Deponerad nyckelsamling

Den deponerade nyckelsamling används för synkronisering med Finder (i macOS 10.15 eller senare) eller iTunes (macOS 10.14 eller tidigare) via USB och MDM. Med den här nyckelsamlingen kan Finder eller iTunes säkerhetskopiera och synkronisera utan att användaren behöver ange någon lösenkod. Den tillåter också att en MDM-lösning kan fjärradera en användares lösenkod. Samlingen förvaras på datorn som används till att synkronisera med Finder eller iTunes, eller i MDM-lösningen som fjärradministrerar enheten.

En deponerad nyckelsamling ger en bättre användarupplevelse vid synkronisering av enheter eftersom det då kan krävas tillgång till alla dataklasser. När en enhet med lösenkodslås ansluts till Finder eller iTunes för första gången blir användaren ombedd att ange lösenkoden. Enheten skapar sedan en deponerad nyckelsamling som innehåller samma klassnycklar som de som används på enheten, skyddade av en ny nyckel. Den deponerade nyckelsamlingen och nyckeln som skyddar den delas upp mellan enheten och värden eller servern. Data lagras på enheten i klassen Protected Until First User Authentication. Det är därför användaren måste ange lösenkoden till enheten innan den säkerhetskopieras med Finder eller iTunes första gången efter en omstart.

Om det rör sig om en trådlös programuppdatering uppmanas användaren att ange sin lösenkod när uppdateringen startas. Denna används till att skapa en engångsupplåsningstoken som låser upp användarnyckelsamlingen efter uppdateringen. Denna token går inte att generera utan att användarens lösenkod anges, och en eventuellt tidigare genererad token blir ogiltig om användarens lösenkod ändras.

En engångsupplåsningstoken är avsedd för antingen övervakad eller oövervakad installation av en programuppdatering. Den krypteras med en nyckel som härleds från det aktuella värdet av en monoton räknare i Secure Enclave, nyckelsamlingens UUID och Secure Enclaves UID.

På SoC:er med A9 (eller senare) är engångsupplåsningstoken inte längre beroende av räknare eller raderingsbart lagringsutrymme. Istället skyddas den av ett Secure Enclave-styrt anti-replay-värde.

En engångsupplåsningstoken för övervakade programuppdateringar upphör att gälla efter 20 minuter. I iOS 13 och iPadOS 13.1 eller senare lagras token i ett utrymme som skyddas av Secure Enclave. Före iOS 13 exporterades denna token från Secure Enclave och sparades i det raderingsbara lagringsutrymmet eller skyddades av Secure Enclaves anti-replay-mekanism. En policytimer räknade upp räknaren om enheten inte hade startat om inom 20 minuter.

Oövervakade programuppdateringar sker när systemet upptäcker att det finns en uppdatering och när något av följande är sant:

- Automatiska uppdateringar är konfigurerade i iOS 12 eller senare.
- Användaren väljer Installera senare när det visas ett meddelande om uppdateringen.

När användaren anger sin lösenkod skapas en engångsupplåsningstoken som är giltig i upp till åtta timmar i Secure Enclave. Så länge uppdateringen inte har installerats förstörs denna engångsupplåsningstoken vid varje låsning och återskapas vid varje efterföljande upplåsning. Varje upplåsning startar om nedräkningen på åtta timmar. Efter åtta timmar gör en policytimer denna engångsupplåsningstoken ogiltig.

## Nyckelsamlingen för iCloud-säkerhetskopiering

Nyckelsamlingen för iCloud-säkerhetskopiering liknar nyckelsamlingen för säkerhetskopiering. Alla klassnycklar i denna nyckelsamling är asymmetriska (de använder Curve25519, precis som dataskyddsklassen Protected Unless Open). En asymmetrisk nyckelsamling används även till att skydda den säkerhetskopierade nyckelringen för återställning av iCloud-nyckelring.

## Skydda nyckar i alternativa startlägen

Dataskydd är utformat för att tillhandahålla åtkomst till användardata endast efter korrekt autentisering, och endast till den behöriga användaren. Dataskyddsklasser är utformade för att stödja en mängd olika användningssituationer, som att kunna skriva och läsa vissa data även efter det att en enhet har låsts (men efter det att den låsts upp första gången). Ytterligare steg tas för att skydda åtkomsten till användardata under alternativa startlägen som de som används för DFU-läge (Device Firmware Update), återställningsläge, Apple Diagnostics eller till och med under programuppdateringar. Dessa egenskaper baseras på en kombination av maskin- och programvarufunktioner, och har utökats allteftersom Apple Silicon-kretsar har utvecklats.

| Funktion | A10 | A11–A17<br>S3–S9<br>M1, M2, M3 |
|---|---|---|
| Återställning: Alla dataskyddsklasser skyddas | ✓ | ✓ |
| Alternativa starter för DFU-läge, återställning och programuppdateringar: Dataskydd för klass A, B och C | ✗ | ✓ |

AES-motorn i Secure Enclave är utrustad med låsbara programvarustartbitar. När nycklar skapas från UID:t ingår de här startbitarna i härledningsfunktionen för nycklar som används till att skapa ytterligare nyckelhierarkier. Hur startbiten används varierar beroende på SoC:

- Med början i enheter med Apple A10 och S3 med SoC är en startbit dedikerad till att särskilja nycklar som skyddas av användarens lösenkod. Startbiten används för nycklar som kräver användarens lösenkod (inklusive nycklar med dataskyddsklass A, klass B och klass C) och rensas för nycklar som inte kräver användarens lösenkod (inklusive filsystemets metadatanyckel och klass D-nycklar).
- På enheter med en A10 eller senare och med iOS 13 eller iPadOS 13.1 eller senare görs alla användardata kryptografiskt oåtkomliga när enheterna startas i diagnosläge. Detta är möjligt genom introduktionen av ytterligare en startbit vars inställning reglerar möjligheten att komma åt medienyckeln som behövs för att i nästa steg komma åt metadata (och därmed innehållet i alla filer) på den datavolym som är krypterad med dataskydd. Det här skyddet omfattar filer som skyddas i alla klasser (A, B, C och D), och inte bara just de som kräver användarens lösenkod.
- I enheter som har A12 med SoC låser Secure Enclaves Boot ROM lösenkodens startbit om approcessorn försätts i DFU-läge eller Återställningsläge. När lösenkodens startbit är låst går den inte att ändra. Det är utformat så för att förhindra åtkomst till data som skyddas med användarens lösenkod.

Återskapning av en enhet efter att den går in i DFU-läge återställer den till ett känt fungerande tillstånd som garanterat endast innehåller oförändrad Apple-signerad kod. DFU-läget kan aktiveras manuellt.

Se följande Apple Support-artiklar om hur du försätter en enhet i DFU-läge:

| Enhet | Apple Support-artikel |
| --- | --- |
| iPhone, iPad | Om du har glömt lösenkoden till din iPhone |
| Apple TV | Om du ser en varningssymbol på din Apple TV |
| Mac-dator med Apple Silicon | Så här återupplivar eller återskapar du fast mjukvara för Mac |

## Skydda användardata i händelse av angrepp

Angripare som försöker extrahera användardata provar ofta ett antal olika tekniker: extrahering av krypterade data till andra medier för automatiserade intrångsförsök eller manipulering av operativsystemversionen, eller andra sätt att ändra eller försvaga säkerhetspolicyn på enheten för att underlätta ett angrepp. Dataangrepp på en enhet kräver ofta kommunikation med enheten via fysiska gränssnitt som Thunderbolt, Lightning eller USB-C. Apple-enheter innehåller funktioner som hjälper till att förhindra sådana angrepp.

Apple-enheter har stöd för en teknik som kallas *SKP (Sealed Key Protection)* som är utformad för att säkerställa att kryptografiskt material blir otillgängligt utanför enheten, eller som används om operativsystemversionerna eller säkerhetsinställningarna manipuleras utan giltig användarauktorisering. Den här funktionen tillhandahålls *inte* av Secure Enclave, utan stöds istället av maskinvaruregister som finns i ett lägre lager för att tillhandahålla ytterligare ett lager av skydd åt de nycklar som krävs för att avkryptera användardata fristående från Secure Enclave.

*Obs!* SKP är endast tillgängligt på enheter med Apple-designad SoC.

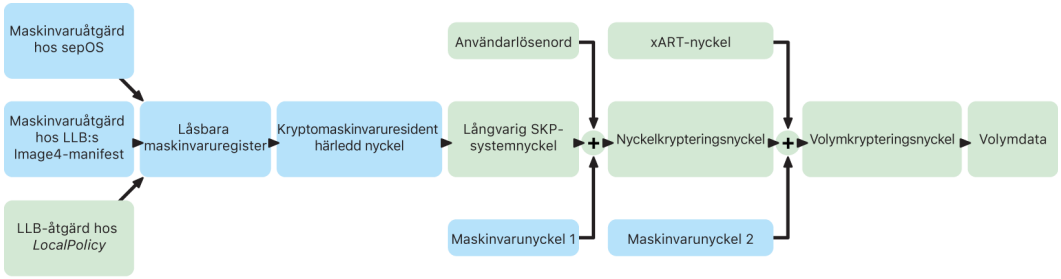
| Funktion | A11–A17<br>S3–S9<br>M1, M2, M3 |
| --- | --- |
| Sealed Key Protection | ✅ |

iPhone- och iPad-enheter kan också konfigureras så att dataanslutningar endast aktiveras under villkor som mest troligt indikerar att enheten fortfarande befinner sig under den auktoriserade ägarens fysiska kontroll.

## SKP (Sealed Key Protection)

På Apple-enheter som stöder dataskydd skyddas KEK (Key Encryption Key) med åtgärder i programvaran i systemet, och är också knuten till det UID som endast är tillgängligt från Secure Enclave. På Mac-datorer med Apple Silicon stärks skyddet av KEK ytterligare genom användning av information om säkerhetspolicyn i systemet eftersom macOS har stöd för kritiska ändringar av säkerhetspolicyer (t.ex. avaktivering av säker start eller SIP) som inte stöds på andra plattformar. På Mac-datorer med Apple Silicon omfattar det här skyddet FileVault-nycklar eftersom FileVault implementeras via dataskydd (klass C).

Nyckeln som skapas när användarens lösenord, långvariga SKP-nyckel och maskinvarunyckel 1 (UID från Secure Enclave) knyts ihop kallas en *lösenordshärledd nyckel*. Nyckeln används till att skydda användarnyckelsamlingen (på alla plattformar som stöds) och KEK (endast i macOS). Sedan används den till att aktivera biometrisk eller automatisk upplåsning med andra enheter som Apple Watch.




Secure Enclaves Boot Monitor samlar in mätvärdena för det Secure Enclave-operativsystem som lästs in. När approcessorns Boot ROM samlar in mätvärden från det Image4-manifest som är kopplat till LLB innehåller det manifestet mätvärden från alla andra systemparkopplade fasta programvaror som också är inlästa. LocalPolicy innehåller de centrala säkerhetskonfigurationerna för det macOS som lästs in. LocalPolicy innehåller också fältet `nsih` som är ett hashvärde för macOS Image4-manifestet. macOS Image4-manifestet innehåller mätvärden för alla macOS-parkopplade fasta maskinvaror och centrala macOS-startobjekt som startkärnsamlingen eller rothashen för den signerade systemvolymen (SSV).

Om en angripare kommer åt att oväntat ändra några mätvärden för fast programvara, programvara eller säkerhetskonfigurationskomponenter ändras de mätvärden som lagras i maskinvaruregistren. Ändringen av mätvärdena leder till att kryptomaskinvaruhärledda *SMRK (System Measurement Root Key)* härleds till ett annat värde, vilket i praktiken bryter sigillet på nyckelhierarkin. Detta leder till att *SMDK (System Measurement Device Key)* blir otillgänglig, vilket i sin tur leder till att KEK (och därmed alla data) blir otillgängliga.

Systemet måste dock hantera legitima programuppdateringar som ändrar mätvärdena för den fasta programvaran och fältet `nsih` i LocalPolicy så att det pekar mot nya macOS-mätvärden när det inte är under attack. I andra system som försöker använda mätvärden från fast programvara, men som inte har någon känd fungerande SOT (Source of Truth), måste användaren avaktivera säkerheten, uppdatera den fasta programvaran och sedan återaktivera den så att det går att samla in en ny mätvärdesbaslinje. Detta ökar risken för att en angripare ska kunna manipulera den fasta programvaran under en programuppdatering. Systemet får hjälp av det faktum att Image4-manifestet innehåller alla de mätvärden som behövs. Maskinvaran som avkrypterar SMDK med SMRK, när mätvärdena matchar under en normal start, kan också kryptera SMDK till ett föreslaget framtida SMRK. Genom att specificera de mätvärden som är förväntade efter en programuppdatering kan maskinvaran kryptera en SMDK som är tillgänglig i ett nuvarande operativsystem så att det fortsätter att vara tillgängligt i ett framtida operativsystem. På samma sätt måste SMDK krypteras till framtida SMRK, baserat på det mätvärde för LocalPolicy som LLB beräknar vid nästa start, när en kund legitimt ändrar sina säkerhetsinställningar i LocalPolicy.

# Apple File System

Apple File System (APFS) är ett företagsägt filsystem som har utformats för att användas med kryptering. APFS fungerar på alla Apples plattformar – på iPhone, iPad, Mac, Apple TV och Apple Watch. Det är optimerat för flash-/SSD-lagring med funktioner som stark kryptering, CoW-metadata (Copy on Write), utrymmesdelning, kloning av filer och kataloger, ögonblicksbilder, snabb beräkning av katalogstorlek, atomic safe-save primitives och förbättrade filsystemsgrunder. Filsystemet har dessutom en unik CoW-design som använder I/O-sammanslagning för att öka prestanda och samtidigt säkerställa tillförlitligheten hos data.

## Utrymmesdelning

APFS tilldelar lagringsutrymme efter behov. När en enskild APFS-behållare har flera volymer delas behållarens lediga utrymme och kan efter behov tilldelas till vilken som helst av de enskilda volymerna. Varje volym använder enbart en del av den totala behållaren, så det tillgängliga utrymmet är behållarens totala storlek minus utrymmet som används i alla volymer i behållaren.

## Flera volymer

I macOS 10.15 eller senare måste en APFS-behållare som används till att starta datorn innehålla minst fem volymer, varav de första tre är gömda för användaren:

- *Förstartsvolym:* Den här volymen är okrypterad och innehåller data som behövs för att starta de enskilda systemvolymerna i behållaren.
- *VM-volym:* Den här volymen är okrypterad och används av macOS till att lagra krypterade växelfiler.
- *Återställningsvolym:* Den här volymen är okrypterad och måste vara tillgänglig utan att låsa upp en systemvolym för att kunna starta i recoveryOS.
- *Systemvolym:* Innehåller följande:
  - Alla filer som behövs för att starta datorn.
  - Alla inbyggda appar som installeras med macOS (appar som förut fanns i mappen /Appar finns nu i /System/Appar).

  *Obs!* Som förval har inga processer skrivbehörighet i systemvolymen, inte ens Apple-systemprocesser.
- *Datavolym:* Innehåller data som brukar ändras, exempelvis:
  - Alla data i användarens mapp, t.ex. bilder, musik, videor och dokument.
  - Appar som användaren har installerat, inklusive AppleScript och Automator-appar.
  - Anpassade ramverk och bakgrundsprocesser som har installerats av användaren, organisationen eller appar från tredje part.
  - Andra platser som användaren äger och kan skriva till, som /Appar, /Bibliotek, /Användare, /Volymer, /usr/local, /private, /var och /tmp.

En datavolym skapas för varje ytterligare systemvolym. Förstarts-, VM- och återställningsvolymerna delas och dupliceras därför inte.

I macOS 11 eller senare är systemvolymen registrerad i en ögonblicksavbild. Operativsystemet startar från en ögonblicksavbild av systemvolymen, inte bara från en skrivskyddad inlänkning av den muterbara systemvolymen.

I iOS och iPadOS är lagringen uppdelad på minst två APFS-volymer:

- Systemvolym
- Datavolym

## Dataskydd genom nyckelring

Många appar måste hantera lösenord och andra korta men viktiga data, till exempel nycklar och inloggningstokens. Nyckelringen gör att du kan lagra dessa objekt på ett säkert sätt. De olika Apple-operativsystemen använder olika mekanismer för att genomdriva de garantier som är kopplade till de olika nyckelringsskyddsklasserna. I macOS (inklusive Mac-datorer med Apple Silicon) används inte dataskydd direkt till att genomdriva dessa garantier.

### Översikt

Nyckelringsobjekt krypteras med två olika AES-256-GCM-nycklar: en tabellnyckel (metadata) och en per-row-nyckel (hemlig nyckel). Nyckelringsmetadata (alla attribut utöver kSecValue) krypteras med metadatanyckeln för att öka hastigheten vid sökningar, och det hemliga värdet (kSecValueData) krypteras med den hemliga nyckeln. Metadatanyckeln skyddas av Secure Enclave, men cachelagras i approcessorn så att det snabbt går att skicka förfrågningar till nyckelringen. Den hemliga nyckeln kräver alltid att förfrågningar görs via Secure Enclave.

Nyckelringen implementeras som en SQLite-databas som lagras i filsystemet. Det finns bara en databas. Bakgrundsdemonen `securityd` avgör vilka nyckelringsobjekt som varje process eller app kan komma åt. Nyckelhanterares API:er genererar anrop till bakgrundsdemonen som sedan söker efter appens behörigheter för ”Keychain-access-groups”, ”application-identifier” och ”application-group”. Med hjälp av tillgångsgrupper kan nyckelringsobjekt delas mellan appar istället för att tillgången begränsas till en enda process.

Nyckelringsobjekt kan endast delas mellan appar från samma utvecklare. För att dela nyckelringsobjekt måste appar från tredje part använda tillgångsgrupper med ett prefix som tilldelas via Apple Developer Program genom appgrupper. Kravet på prefix och på att appgruppen ska vara unik upprätthålls genom kodsignering, tillhandahållandeprofiler och Apple Developer Program.

Nyckelringsdata skyddas med hjälp av en klasstruktur som liknar den som används för dataskyddet på filnivå. Klasserna fungerar på samma sätt som dataskyddsklasserna för filer, men har andra nycklar och funktioner.

| Tillgänglighet | Dataskydd på filnivå | Dataskydd genom nyckelring |
|---|---|---|
| När den är upplåst | NSFileProtectionComplete | kSecAttrAccessibleWhenUnlocked |
| När den är låst | NSFileProtectionComplete UnlessOpen | ✕ |
| Efter första upplåsningen | NSFileProtectionComplete UntilFirstUserAuthentication | kSecAttrAccessibleAfterFirstUnlock |
| Alltid | NSFileProtectionNone | kSecAttrAccessibleAlways |
| Lösenkodsaktiverad | ✕ | kSecAttrAccessibleWhen PasscodeSetThisDeviceOnly |

Appar som använder uppdateringstjänster i bakgrunden kan använda *kSecAttrAccessibleAfterFirstUnlock* för nyckelringsobjekt som behöver vara tillgängliga under bakgrundsuppdateringar.

Klassen *kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly* beter sig likadant som *kSecAttrAccessibleWhenUnlocked*, men den är bara tillgänglig när enheten har konfigurerats med en lösenkod. Den här klassen finns bara i systemets nyckelsamling och den:

- Synkroniseras inte till iCloud-nyckelring
- Säkerhetskopieras inte
- Ingår inte i deponerade nyckelsamlingar

Om lösenkoden tas bort eller nollställs görs objekten oanvändbara genom att klassnycklarna kastas.

Andra nyckelringsklasser har en motsvarighet till funktionen ”endast denna enhet”, som alltid skyddas med UID:t när den kopieras från enheten vid säkerhetskopiering, vilket gör att den blir oanvändbar om den återskapas på en annan enhet. För att hitta en bra balans mellan säkerhet och användbarhet väljer Apple nyckelringsklasser som varierar beroende på vilken typ av information som ska skyddas och när iOS och iPadOS behöver tillgång till den.

## Dataklasskydd genom nyckelring

De klasskydd som listas nedan genomdrivs för nyckelringsobjekt.

| Objekt | Tillgängligt |
|---|---|
| Wi-Fi-lösenord | Efter första upplåsningen |
| E-postkonton | Efter första upplåsningen |
| Microsoft Exchange ActiveSync-konton | Efter första upplåsningen |
| VPN-lösenord | Efter första upplåsningen |
| LDAP, CalDAV, CardDAV | Efter första upplåsningen |
| Tokens för sociala nätverkskonton | Efter första upplåsningen |
| Krypteringsnycklar för Handoff-annonsering | Efter första upplåsningen |
| iCloud-token | Efter första upplåsningen |
| iMessage-nycklar | Efter första upplåsningen |
| Lösenord för Hemmadelning | När den är upplåst |
| Safari-lösenord | När den är upplåst |
| Safari-bokmärken | När den är upplåst |
| Säkerhetskopiering med Finder/iTunes | När den är upplåst, ej flyttbar |
| VPN-certifikat | Efter första upplåsningen, ej flyttbar |
| Bluetooth-nycklar | Alltid, ej flyttbar |
| APNs (Apples tjänst för pushnotiser)-token | Alltid, ej flyttbar |
| iCloud-certifikat och privata nycklar | Alltid, ej flyttbar |
| SIM-kortets PIN-kod | Alltid, ej flyttbar |
| Token för Hitta | Alltid |
| Röstbrevlåda | Alltid |

I macOS är alla nyckelringsobjekt som installeras av konfigurationsprofiler *alltid* tillgängliga. I iOS och iPadOS har nyckelringsobjekt som installeras av en konfigurationsprofil olika åtkomster beroende på deras typ, hur de refereras till och när de installerades. Som förval är nyckelringsobjekt som har installerats med konfigurationsprofiler *tillgängliga efter första upplåsningen och ej flyttbara*. Ett nyckelringsobjekt som har installerats av en konfigurationsprofil är dock *alltid* tillgängligt om det:

- Installerades före uppgradering till iOS 15, iPadOS 15 eller senare
- Är ett certifikat (inte en identitet)
- Är en identitet som refereras till av `IdentityCertificateUUID` i en `com.apple.mdm`-nyttolast

## Behörighetsstyrning av nyckelringen

Nyckelringar kan använda behörighetslistor (ACL:er) för att ange policyer för tillgång samt autentiseringskrav. Objekten kan upprätta villkor som kräver användarens närvaro genom att begära autentisering med Face ID, Touch ID eller att enhetens lösenkod eller lösenord måste anges. Du kan begränsa tillgången till objekt genom att ange att registreringen av Face ID eller Touch ID inte har ändrats sedan objektet lades till. Denna begränsning förhindrar att obehöriga personer lägger till egna fingeravtryck för tillgång till ett nyckelringsobjekt. ACL:erna granskas inuti Secure Enclave och lämnas endast ut till kärnan om de angivna villkoren uppfylls.

## Nyckelringsarkitektur i macOS

macOS ger också tillgång till nyckelringen för att smidigt och säkert lagra användarnamn och lösenord, digitala identiteter, krypteringsnycklar och säkra anteckningar. Du kommer åt den genom att öppna appen Nyckelhanterare i /Appar/Verktyg/. Genom att använda en nyckelring behöver användaren inte ange – eller ens komma ihåg – inloggningsuppgifterna för varje resurs. En första förvald nyckelring skapas för varje Mac-användare, men användarna kan också skapa andra nyckelringar för olika ändamål.

Utöver att förlita sig på användarnyckelringar använder macOS ett antal nyckelringar på systemnivå som håller reda på autentiseringsinformation som inte är användarspecifik, som nätverksbehörigheter och identiteter via PKI-certifikat (public key infrastructure). En av dessa nyckelringar, Systemrötter, kan inte ändras och lagrar internet-PKI-rotcertifikat från certifikatutfärdare (CA) för att underlätta vanliga åtgärder som banktjänster och e-handel via internet. Användaren kan även distribuera internt tillhandahållna CA-certifikat till hanterade Mac-datorer för att underlätta valideringen av interna webbplatser och tjänster.

# FileVault

## Volymkryptering med FileVault i macOS

Mac-datorer har FileVault som är en inbyggd krypteringsfunktion som skyddar alla data vid vila. FileVault använder AES-XTS-datakrypteringsalgoritmen till att skydda hela volymer för både interna och borttagbara lagringsenheter.

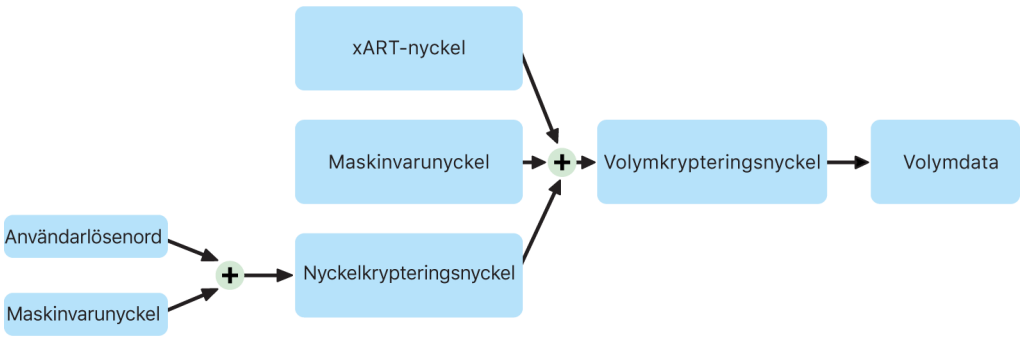
På Mac-datorer med Apple Silicon implementeras FileVault via dataskyddsklass C med en volymnyckel. På Mac-datorer med Apple Silicon och Mac-datorer med Apple T2-säkerhetskrets använder krypterade interna lagringsenheter som är anslutna till Secure Enclave de funktioner för maskinvarusäkerhet som de har liksom de som hör till AES-motorn. När en användare har aktiverat FileVault på en Mac krävs användarens inloggningsuppgifter under startprocessen.

*Obs!* För Mac-datorer (1) *som är äldre än de med T2-krets* eller (2) med intern lagring som *inte ursprungligen levererades med datorn* eller (3) med ansluten extern lagring: När FileVault har aktiverats blir alla befintliga filer och alla data som skrivs i fortsättningen krypterade. Data som lades till och därefter raderades innan FileVault aktiverades blir inte krypterade och kan eventuellt återskapas med forensiska dataräddningsverktyg.

### Intern lagring när FileVault är aktiverat

Utan giltiga inloggningsuppgifter eller en kryptografisk återställningsnyckel fortsätter de interna APFS-volymerna att vara krypterade. De är skyddade mot obehörig åtkomst även om den fysiska lagringsenheten tas bort och ansluts till en annan dator. I macOS 10.15 gäller detta både system- och datavolymen. Från och med macOS 11 skyddas systemvolymen av den signerade systemvolymen (SSV-volymen), men datavolymen fortsätter att skyddas av kryptering. Kryptering av interna volymer på Mac-datorer med Apple Silicon och på de med T2-krets implementeras genom att konstruera och hantera en hierarki av nycklar och bygger på de krypteringstekniker för maskinvara som är inbyggda i kretsen. Den här nyckelhierarkin är utformad för att uppnå fyra mål samtidigt:

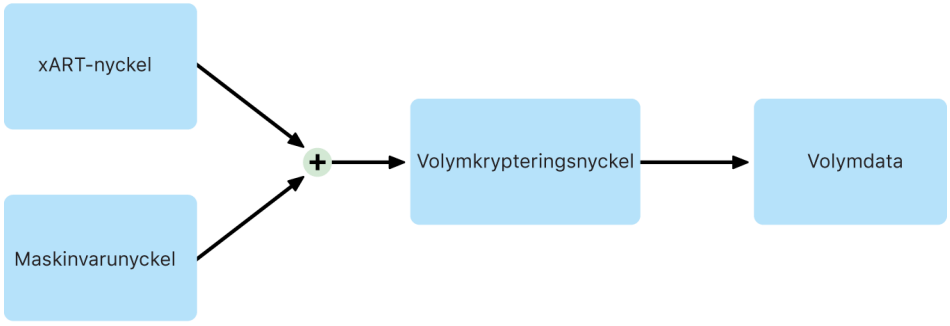
- kräva användarens lösenord för avkryptering
- skydda systemet från ett automatiserat intrångsförsök riktat direkt mot lagringsmedier som har tagits bort från datorn
- tillhandahålla en snabb och säker metod för att radera innehåll genom att nödvändigt kryptografiskt material raderas
- göra det möjligt för användare att byta lösenord (och därmed de kryptografiska nycklar som används till att skydda filerna) utan att kräva omkryptering av hela volymen



På Mac-datorer med Apple Silicon och Mac-datorer med T2-krets sker all FileVault-nyckelhantering i Secure Enclave. Krypteringsnycklarna exponeras aldrig direkt mot Intel-processorn. Som förval skapas alla APFS-volymer med en volymkrypteringsnyckel. Volym- och metadatainnehåll krypteras med den här volymkrypteringsnyckeln som paketeras med en KEK (key encryption key). Denna KEK skyddas av en kombination av användarens lösenord och maskinvaru-UID när FileVault är aktiverat.

## Intern lagring när FileVault är avaktiverat

Om FileVault inte aktiveras på Mac-datorer med Apple Silicon eller på Mac-datorer med T2-krets under den inledande processen med inställningsassistenten krypteras volymen fortfarande, men volymkrypteringsnyckeln skyddas bara av maskinvaru-UID: t i Secure Enclave.




Om FileVault aktiveras senare – en process som är omedelbar eftersom alla data redan har krypterats – förhindrar en anti-replay-mekanism att den gamla nyckeln (som endast baseras på maskinvaru-UID) kan användas till att avkryptera volymen. Volymen skyddas sedan av en kombination av användarens lösenord och maskinvaru-UID enligt vad som beskrivs tidigare.

## Radera FileVault-volymer

När en volym raderas blir dess volymkrypteringsnyckel raderad på ett säkert sätt av Secure Enclave. Detta hjälper till att förhindra senare åtkomst med hjälp av nyckeln, även för Secure Enclave. Dessutom paketeras alla volymkrypteringsnycklar med en medienyckel. Medienyckeln ger inte något ytterligare dataskydd. Det gör det istället möjligt att snabbt och säkert radera data eftersom ingen avkryptering kan ske utan den.

På Mac-datorer med Apple Silicon och Mac-datorer med T2-krets garanteras raderingen av medienyckeln av den teknik som Secure Enclave stöder, t.ex. genom fjärr-MDM-kommandon. Om du tar bort medienyckeln på det sättet blir volymen kryptografiskt oåtkomlig.

## Borttagbara lagringsenheter

Vid kryptering av borttagbara lagringsenheter används inte de säkerhetsfunktioner som finns i Secure Enclave, utan krypteringen av dem utförs på samma sätt som för Intel-baserade Mac-datorer utan T2-krets.

# Hantera FileVault i macOS

I macOS kan organisationer hantera FileVault med SecureToken eller Bootstrap Token.

## Använda SecureToken

APFS (Apple File System) i macOS 10.13 och senare ändrar hur FileVault-krypteringsnycklar genereras. I tidigare versioner av macOS på CoreStorage-volymer skapades de nycklar som används i FileVault-krypteringsprocessen när en användare eller organisation aktiverade FileVault på en Mac. I macOS på APFS-volymer genereras nycklarna antingen genom att användaren skapar dem, när den första användarens lösenord ställs in eller när en användare loggar in första gången på datorn. Den här implementeringen av krypteringsnycklarna, när de genereras och hur de lagras ingår tillsammans i en funktion som heter *Secure Token*. Mer specifikt är en säker token en paketerad version av en KEK (Key Encryption Key) som skyddas av en användares lösenord.

Vid driftsättning av FileVault på APFS kan användaren fortsätta att:

- använda befintliga verktyg och processer, som en personlig återställningsnyckel som kan deponeras i en MDM-lösning
- skjuta upp aktivering av FileVault tills en användare loggar in eller ut ur datorn
- skapa och använda organisationens återställningsnyckel

När det första lösenordet för den allra första användaren på datorn anges i macOS 11 tilldelas den användaren en säker token. I vissa arbetsflöden är det möjligtvis inte det önskade beteendet eftersom, som tidigare nämnts, beviljande av den första säkra token kräver att användaren loggar in. Lägg till `;DisabledTags;SecureToken` i användarens `AuthenticationAuthority`-attribut som skapats av programvara innan du ställer in användarens lösenord för att förhindra detta. Gör så här:

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

## Använda Bootstrap Token

Med macOS 10.15 introducerades en ny funktion – *Bootstrap Token* – som hjälper till att bevilja en säker token till både flyttbara konton och det valfria administratörskonto som kan skapas vid registreringen av en enhet (”hanterad administratör”). I macOS 11 kan en bootstrap token bevilja en säker token till en användare som loggar in på en Mac-dator, inklusive lokala användarkonton. För att använda Bootstrap Token-funktionen i macOS 10.15 eller senare krävs:

- Mac-registrering i MDM via Apple School Manager eller Apple Business Manager, vilket gör datorn övervakad
- att MDM-utvecklaren stöder funktionen

I macOS 10.15.4 eller senare skapas en bootstrap token som deponeras i MDM första gången en användare som är Secure Token-aktiverad loggar in om MDM-lösningen har stöd för funktionen. En Bootstrap Token kan vid behov genereras och deponeras i MDM genom användning av kommandoradsverktyget `profiles`.

I macOS 11 kan en bootstrap token också användas till mer än att bara bevilja säkra tokens till användarkonton. På Mac-datorer med Apple Silicon kan en eventuell bootstrap token användas till att auktorisera installationen av både kärntillägg och programuppdateringar när de hanteras med MDM.

### Organisationens jämfört med personliga återställningsnycklar

FileVault i både CoreStorage- och APFS-volymer stöder användningen av en organisations återställningsnyckel (IRK, kallades tidigare *FileVault Master identity*) för att låsa upp volymen. En IRK är visserligen praktisk för kommandoradsåtgärder som att låsa upp en volym eller stänga av FileVault helt och hållet, men dess användbarhet för organisationer är begränsad (särskilt i de senaste versionerna av macOS). På en Mac med Apple Silicon tillför IRK:er inget funktionellt värde av två huvudsakliga skäl: För det första kan IRK:er inte användas till att komma åt recoveryOS och för det andra kan volymen inte låsas upp genom att ansluta den till en annan Mac eftersom hårddiskläge inte längre stöds. På grund av dessa skäl och andra anledningar *rekommenderas inte längre en IRK för organisationshantering av FileVault på Mac-datorer*. Istället bör en personlig återställningsnyckel (PRK) användas.

# Hur Apple skyddar användarnas personliga data

## Skydda apptillgång till användardata

Utöver att kryptera data i viloläge hjälper Apple-enheter till att förhindra att appar kommer åt användarens personliga information utan behörighet genom användning av olika tekniker, inklusive datavalv. Användare kan öppna Inställningar i iOS och iPadOS, eller Systeminställningar i macOS, och se vilka appar som har tillåtits att komma åt viss information samt godkänna fortsatt tillgång eller återkalla den. Tillgångskrav gäller i följande fall:

- *iOS, iPadOS och macOS:* Kalendrar, kamera, kontakter, mikrofon, bilder, påminnelser och röstigenkänning
- *iOS och iPadOS:* Bluetooth, Hem, medier, medieappar och Apple Music, rörelse och kondition
- *iOS och watchOS:* Hälsa
- *macOS:* Övervakning av inmatning (t.ex. tangentbordstryckningar), dialogrutor, skärminspelningar (t.ex. statiska skärmavbildningar och video) och Systeminställningar

I iOS 13.4 eller senare och iPadOS 13.4 eller senare skyddas data från tredjepartsappar i ett datavalv. Datavalvet ger ett skydd mot obehörig åtkomst till data även från processer som inte körs i sandlåda. Ytterligare klasser i iOS 15 eller senare inkluderar Local Network, Nearby Interactions, Research Sensor & Usage Data och Focus.

Om användaren loggar in på iCloud i iOS och iPadOS får apparna tillgång till iCloud Drive som förval. Användaren kan styra tillgången per app under iCloud i Inställningar. iOS och iPadOS ger också möjlighet att begränsa dataöverföring mellan appar och konton som har installerats av en MDM-lösning och de som har installerats av användaren.

# Skydda tillgången till användarnas hälsodata

HealthKit tillhandahåller en central förvaringsplats för hälso- och fitnessdata på iPhone och Apple Watch. HealthKit fungerar också direkt med hälso- och träningstillbehör som Bluetooth LE-kompatibla anslutna pulsmätare och rörelsecoprocessorn som är inbyggd i många iOS-enheter. All HealthKit-interaktion med hälso- och träningsappar, vårdgivare och hälso- och träningstillbehör kräver godkännande från användaren. Dessa data lagras i dataskyddsklassen Protected Unless Open. Tillgången till dessa data upphör tio minuter efter att enheten låses och de blir tillgängliga igen nästa gång användaren anger sin lösenkod eller använder Face ID eller Touch ID till att låsa upp enheten.

## Samla in och lagra hälso- och fitnessdata

HealthKit samlar också in och lagrar hanteringsdata, som tillgångsbehörigheter för appar, namn på enheter som är anslutna till HealthKit samt schemaläggningsinformation som används till att starta appar när nya data är tillgängliga. Dessa data lagras i dataskyddsklassen Protected Until First User Authentication. Tillfälliga journalfiler sparar hälsodata som genereras när enheten är låst, till exempel när användaren tränar. Dessa lagras i dataskyddsklassen Protected Unless Open. När enheten låses upp importeras de tillfälliga journalfilerna till den primära hälsodatabasen och raderas sedan när sammanslagningen är klar.

Hälsodata kan lagras på iCloud. Heltäckande kryptering för hälsodata kräver iOS 12 eller senare och tvåfaktorsautentisering. Annars krypteras användarens data fortfarande under lagring och överföring, men de är inte heltäckande krypterade. När användaren har aktiverat tvåfaktorsautentisering och uppdaterar till iOS 12 eller senare blir personens hälsodata heltäckande krypterade.

Om användaren säkerhetskopierar sin enhet med Finder (i macOS 10.15 eller senare) eller iTunes (macOS 10.14 eller tidigare) lagras hälsodata endast om säkerhetskopian är krypterad.

## Medicinska journaler

Användare kan logga in på hälsosystem hos kompatibla vårdgivare inuti appen Hälsa och hämta en kopia av sina medicinska journaler. När en användare ansluter till ett hälsosystem autentiserar användaren med OAuth 2-klientautentiseringsuppgifter. När användaren är ansluten hämtas medicinska journaldata direkt från vårdgivaren via en anslutning som skyddas med TLS 1.3. När de har hämtats lagras medicinska journaler tryggt och säkert tillsammans med andra hälsodata.

## Hälsodata och autenticitet

Bland de data som lagras i databasen finns också metadata som kan användas till att spåra posternas ursprung. Dessa metadata innehåller en appidentifierare som identifierar den app som lagrar respektive post. Dessutom finns ett valfritt metadataobjekt som kan innehålla en digitalt signerad kopia av posten. Syftet med det är att erbjuda dataautenticitet för poster som har genererats av en betrodd enhet. Formatet som används för den digitala signaturen är CMS (Cryptographic Message Syntax) som specificeras i RFC 5652.

## Tillgång till hälsodata för tredjepartsappar

Tillgången till API:t för HealthKit styrs med hjälp av behörigheter och appar måste anpassa sig efter begränsningar för hur hälsodata får användas. Exempelvis får appar inte använda hälsodata i reklamsyften. Alla appar måste också innehålla en integritetspolicy som specificerar appens användning av hälsodata.

Appars tillgång till hälsodata regleras av användarens integritetsinställningar. Användaren tillfrågas när appar begär tillgång till hälsodata, på samma sätt som med Kontakter, Bilder och andra datakällor i iOS. När det gäller hälsodata ges dock apparna separat tillgång för läsning och skrivning av data, liksom separat tillgång för de olika typerna av hälsodata. Användarna kan visa och återkalla behörigheter för tillgång till hälsodata under Inställningar > Hälsa > Datatillgång och enheter.

Appar med behörighet att skriva data kan också läsa de data de skriver. Appar med behörighet att läsa data kan läsa data från alla källor. Däremot kan ingen app avgöra vilka behörigheter andra appar har. Dessutom kan appar inte avgöra helt säkert om de har fått behörighet att läsa hälsodata eller inte. När en app inte har läsbehörighet returnerar alla förfrågningar ”inga data” – samma svar som en tom databas skulle ge. Detta är tänkt att förhindra att appar drar slutsatser om användarens hälsa genom att ta reda på vilka typer av hälsodata användaren registrerar.

## Medicinskt ID för användare

I appen Hälsa har användarna möjlighet att fylla i ett formulär för ett medicinskt ID med information som kan vara viktig i medicinska nödsituationer. Informationen anges eller uppdateras manuellt och synkroniseras inte med informationen i hälsodatabaserna.

Den medicinska ID-informationen visas med ett tryck på knappen Nödsituation på låsskärmen. Informationen lagras på enheten i dataskyddsklassen No Protection så att den kan visas utan att enhetens lösenkod anges. Medicinskt ID är en valfri funktion där användarna själva kan bestämma hur de vill balansera säkerhet mot integritet. Dessa data säkerhetskopieras i iCloud-säkerhetskopiering i iOS 13 eller tidigare. I iOS 14 synkroniseras Medicinskt ID mellan enheter via CloudKit och har samma krypteringsegenskaper som övriga hälsodata.

## Hälsodelning

I iOS 15 har användare möjlighet att dela sina hälsodata med andra användare via appen Hälsa. Hälsodata delas mellan två användare med heltäckande iCloud-kryptering och Apple har inte tillgång till de data som skickas via hälsodelning. För att kunna använda den här funktionen måste både avsändaren och mottagaren använda iOS 15 eller senare och ha tvåfaktorsautentisering aktiverad.

Användare kan också välja att dela sina hälsodata med en vårdgivare via funktionen Dela med vårdgivare i appen Hälsa. Data som delas med den här funktionen görs endast tillgängliga för vårdgivare som har valts av användaren genom att använda heltäckande kryptering, och Apple varken sparar eller har tillgång till krypteringsnycklarna som krävs för att avkryptera, visa eller på annat sätt få tillgång till de hälsodata som delas via funktionen Dela med vårdgivare. Mer information om hur utformningen av den här tjänsten skyddar användares hälsodata finns i Security and Privacy section i Apple Registration Guide for Healthcare Organizations.

# Digital signering och kryptering

## Behörighetslistor

Data i nyckelringen delas upp och skyddas via behörighetslistor (ACL:er). Därför kan inloggningsuppgifter som lagras av tredjepartsappar inte nås av andra appar med andra identiteter om inte användaren uttryckligen godkänner det. Detta skydd gör det möjligt att skydda autentiseringsuppgifter på Apple-enheter i olika appar och tjänster inom organisationen.

## Mail

Med Mail kan användarna skicka mejl som är digitalt signerade och krypterade.
Mail upptäcker automatiskt lämpliga RFC 5322-skiftlägeskänsliga e-postadressämnen eller alternativa ämnesnamn på certifikat för digital signering eller kryptering på bifogade PIV-token (Personal Identification Verification) i kompatibla smarta kort.
Om ett konfigurerat e-postkonto stämmer överens med en e-postadress på ett certifikat för digital signering eller kryptering på en bifogad PIV-token visar Mail automatiskt signeringsknappen i verktygsfältet i ett nytt mejlfönster. Om Mail har tillgång till mottagarens krypteringscertifikat för e-post, eller kan hitta det i Microsoft Exchanges globala adresslista (GAL, Global Adress List), visas en upplåst låssymbol i verktygsfältet för ett nytt mejl. En låst låssymbol innebär att mejlet kommer att skickas krypterat med mottagarens publika nyckel.

## S/MIME per mejl

iOS, iPadOS och macOS har stöd för S/MIME per mejl. Det innebär att S/MIME-användare kan välja att alltid signera och kryptera mejl som förval, eller selektivt signera och kryptera enskilda mejl.

Identiteter som används med S/MIME kan överföras till Apple-enheter via en konfigurationsprofil, en MDM-lösning, SCEP (Simple Certificate Enrollment Protocol) eller Microsoft Active Directory-certifikatutfärdare.

## Smarta kort

macOS 10.12 och senare innehåller inbyggt stöd för PIV-kort. De här korten används i många kommersiella organisationer och myndigheter för tvåfaktorsautentisering, digital signering och kryptering.

Smarta kort innehåller en eller två digitala identiteter med ett offentligt och privat nyckelpar och tillhörande certifikat. När du låser upp ett smart kort med PIN-koden får du tillgång till de privata nycklar som används för autentisering, kryptering och signering. Certifikatet avgör vad en nyckel kan användas till, vilka attribut som är kopplade till den och om den är validerad (signerad) av ett certifikat från en certifikatutfärdare.

Smarta kort kan användas för tvåfaktorsautentisering. De två faktorer som krävs för att låsa upp ett kort är ”någonting användaren har” (kortet) och ”någonting användaren känner till” (PIN-koden). macOS 10.12 och senare har också systemspecifikt stöd för autentisering i inloggningsfönster för smarta kort och autentisering med klientcertifikat på webbplatser i Safari. Den stöder även Kerberos-autentisering med nyckelpar (PKINIT) för enkel inloggning till tjänster som Kerberos stöder. Du kan läsa mer om smarta kort och macOS i Introduktion till integrering av smarta kort i *Apple och driftsättning*.

## Krypterade skivavbilder

I macOS fungerar krypterade skivavbilder som säkra behållare som kan användas till att lagra eller överföra känsliga dokument och andra filer. Krypterade skivavbilder skapas med Skivverktyg som finns i /Appar/Verktyg/. Skivavbilder kan krypteras med antingen 128-bitars eller 256-bitars AES-kryptering. Eftersom en inlänkad skivavbild behandlas som en lokal volym som är ansluten till en Mac kan användarna kopiera, flytta och öppna filer och mappar som lagras på den. Precis som med FileVault krypteras och avkrypteras innehållet i en skivavbild i realtid. Med krypterade skivavbilder kan användarna utbyta dokument, filer och mappar säkert genom att spara en krypterad skivavbild på ett löstagbart medium, skicka den som en mejlbilaga eller lagra den på en fjärrserver. Mer information om krypterade skivavbilder finns i Skivverktyg Användarhandbok.

# Appsäkerhet

## Appsäkerhet i översikt

Nuförtiden tillhör appar de mest sårbara delarna av en säkerhetsarkitektur. Appar kan innebära enorma produktivitetsfördelar för användarna, men de kan också ha negativa effekter på systemets säkerhet, stabilitet och användardata om de inte hanteras korrekt.

Därför tillhandahåller Apple flera lager av skydd för att ytterligare säkerställa att appar inte innehåller kända sabotageprogram och inte har manipulerats. Ytterligare skydd ser till att åtkomsten till användardata via appar inte förmedlas hur som helst. De här säkerhetskontrollerna skapar en stabil och säker plattform för appar med tusentals utvecklare som levererar hundratusentals appar för iOS, iPadOS och macOS utan att det påverkar systemets integritet. Och användarna kan öppna apparna på sina Apple-enheter utan att oroa sig för virus, sabotageprogram eller attacker från obehöriga.

På iPhone och iPad hämtas alla appar från App Store – och alla appar körs i sandlåda – för största möjliga kontroll.

På Mac-datorer hämtas många appar från App Store, men Mac-användare kan även hämta och använda appar från internet. För att tillhandahålla säkert stöd för internethämtning har macOS flera lager av kontroller. För det första måste alla Mac-appar i macOS 10.15 och senare som förval attesteras av Apple för att kunna startas. Detta krav säkerställer att apparna inte innehåller kända sabotageprogram utan att kräva att apparna tillhandahålls via App Store. För det andra innehåller macOS ett kraftigt antivirusskydd för att blockera – och om det behövs ta bort – sabotageprogram.

Som ett extra skyddsnät mellan plattformarna hjälper sandlådor till att skydda användardata från icke-behörig appåtkomst. Och i macOS skyddas alla de data som finns inom kritiska områden – vilket ytterligare säkerställer att användaren styr åtkomsten till filer på skrivbordet och i mapparna Dokument och Hämtade filer och inom andra områden från alla appar, vare sig apparna som söker åtkomst själva körs i sandlåda eller inte.

| **Inbyggd funktion** | **Motsvarighet från tredje part** |
|---|---|
| Listor med ej godkända insticksfiler och Safari-tillägg | Definitioner för virus/sabotageprogram |
| Filkarantän | Definitioner för virus/sabotageprogram |
| XProtect-/YARA-signaturer | Definitioner för virus/sabotageprogram, slutpunktsskydd |
| Gatekeeper | Slutpunktsskydd genomdriver kodsignering av appar för att säkerställa att endast betrodd programvara körs. |

| Inbyggd funktion | Motsvarighet från tredje part |
| --- | --- |
| eficheck<br>(behövs för Mac-datorer utan Apple T2-säkerhetskrets) | Slutpunktsskydd, spökprogramsavkänning |
| Appbrandvägg | Slutpunktsskydd, brandvägg |
| Paketfilter (pf) | Brandväggslösningar |
| Systemintegritetsskydd | Inbyggt i macOS |
| Obligatoriska åtkomstkontroller | Inbyggt i macOS |
| Kext-exkluderingslista | Inbyggt i macOS |
| Obligatorisk appkodsignering | Inbyggt i macOS |
| Appattestering | Inbyggt i macOS |

# Appsäkerhet i iOS och iPadOS

## Introduktion till appsäkerhet för iOS och iPadOS

Till skillnad från andra mobilplattformar tillåter inte iOS och iPadOS att användarna installerar potentiellt skadliga, osignerade appar från webbplatser eller kör appar som inte är betrodda. Istället (utanför EU) måste alla appar hämtas från App Store där alla appar kommer från identifierade utvecklare och måste passera automatiserad och mänsklig granskning. Vid körning kontrolleras kodsignaturen för alla körbara minnessidor när sidorna läses in – detta för att bekräfta att appen inte har modifierats sedan installationen eller den senaste uppdateringen.

När en app har kontrollerats och bekräftats komma från en godkänd källa använder iOS och iPadOS tvingande säkerhetsåtgärder som förhindrar att den påverkar andra appar eller resten av systemet.

## Om App Store-säkerhet

App Store är en tillförlitlig plats där användare tryggt kan upptäcka och hämta appar. Apparna på App Store kommer från identifierade utvecklare som har samtyckt till att följa Apples riktlinjer och distribueras på ett säkert sätt till användare med kryptografiska garantier mot ändringar. Varenda app och varje appuppdatering granskas för att utvärdera om den uppfyller kraven på integritet och säkerhet. Den här processen, som ständigt förbättras, är utformad för att skydda användare genom att hålla sabotageprogram, cyberbrottslingar och bedrägerier borta från App Store. Dessutom måste appar som är utformade för barn följa strikta riktlinjer gällande datainsamling och säkerhet som är utformade så att barn ska vara trygga, och de måste vara tätt integrerade med föräldrakontrollsfunktionerna i iOS och iPadOS.

Säkerhetsskydd i App Store inkluderar:

- *Automatiska skanningar efter kända sabotageprogram:* För att förhindra att de någonsin når App Store och därmed någonsin når eller skadar användare.
- *Mänsklig granskning av ett expertteam:* För att kontrollera att appbeskrivningen, inklusive marknadsföringstext och skärmavbilder, är korrekt. Det här skapar en hög tröskel mot de vanligaste metoderna som används till att sprida spionprogram: maskera spionprogrammet som en populär app eller locka användare genom att erbjuda funktioner som inte finns.
- *Manuella kontroller:* För att säkerställa att appen inte begär tillgång till känsliga data i onödan och extra noggrann utvärdering av appar som riktar sig till barn för att se till att de följer strikta regler gällande datainsamling och säkerhet.
- *Tillförlitliga, centraliserade användarrecensioner:* För att göra problem kända och kraftigt minska angripares möjligheter att vilseleda många användare. Även om en skadlig app fullständigt lyckades gömma sitt beteende under granskningsprocessen kan appanvändare som stöter på och rapporterar problem varna andra – och Apple – och på det sättet bidra till att upptäcka sabotageprogram. App Store arbetar hårt med att bekämpa falska recensioner så att signalvärdet förbättras.
- *Processer för korrigering och borttagning:* Ifall ett problem inträffar. Ifall en app tar sig till App Store, och det senare upptäcks att den bryter mot riktlinjer, samarbetar Apple med utvecklaren för att snabbt lösa problemet. I allvarliga fall, vilket kan omfatta bedrägerier och skadlig aktivitet, tas appen omedelbart bort från App Store och användare som har hämtat apparna kan meddelas om appens skadliga beteende.

Säkerheten för appar i iOS och iPadOS bygger på en kombination av alla lager – en robust appgranskning bidrar till att förhindra installation av skadliga appar och robusta plattformsskydd begränsar den skada som skadliga appar kan orsaka. Säkerheten som är inbyggd i iOS och iPadOS ger användarna kraftfulla skydd som är oöverträffade på konsumentenheter, men de skydden är inte utformade så att de skyddar mot val som en användare kan bli lurad att göra. Appgranskningen genomdriver App Store-policyerna som har utformats för att skydda användare från appar som kan försöka skada dem eller lura dem att ge tillgång till känsliga data. I de allvarliga fall där skadliga appar försöker förbigå skydd på enheten gör appgranskningen det svårare för dem att först och främst hamna på användares enheter.

App Stores säkerhetsåtgärder är aldrig perfekta på egen hand, men som en del av en djupgående strategi för plattformssäkerhet bidrar de till att omfattande angrepp mot iOS- och iPadOS-användare blir svårare och olönsamma för angripare med ekonomiska motiv. Genom att granska varje app innan den blir tillgänglig på App Store säkerställer Apple att den är fri från sabotageprogram och framställs korrekt för användare. Genom att snabbt ta bort appar från distribution om de upptäcks vara skadliga och begränsa spridningen av kommande varianter skyddar Apple säkerheten i ekosystemet och gör att kunder kan känna sig trygga.

# Kodsigneringsprocess för appar i iOS och iPadOS

I iOS och iPadOS erbjuder Apple appsäkerhet genom obligatorisk kodsignering, strikt inloggning för utvecklare med mera.

## Obligatorisk kodsignering

När iOS- eller iPadOS-kärnan har startat styr den vilka användarprocesser och appar som kan köras. För att garantera att alla appar kommer från en känd och godkänd källa, och inte har manipulerats, kräver iOS och iPadOS att all körbar kod ska vara signerad med ett certifikat som utfärdas av Apple. De appar som följer med enheten, som Mail och Safari, är signerade av Apple. Tredjepartsappar måste också valideras och signeras med ett certifikat som utfärdas av Apple. Med obligatorisk kodsignering utökas tillförlitlighetskedjan från operativsystemet till apparna. Det förhindrar att tredjepartsappar läser in osignerade kodresurser eller använder självmodifierande kod.

## Hur utvecklare signerar sina appar

Utvecklare kan signera sina appar genom certifikatvalidering (via Apple Developer Program). De kan också bädda in ramverk inuti apparna och låta den koden valideras med ett certifikat som är utfärdat av Apple (via en team-ID-sträng).

- *Certifikatvalidering:* För att utveckla och installera appar på iPhone- eller iPad-enheter måste utvecklare registrera sig hos Apple och gå med i Apple Developer Program. Varje utvecklares verkliga identitet – oavsett om det är en individ eller ett företag – kontrolleras av Apple innan ett certifikat utfärdas. Med det här certifikatet kan utvecklarna signera appar och skicka in dem till App Store för distribution. Det innebär att alla appar i App Store har skickats in av en identifierbar person eller organisation, vilket avskräcker från att skicka in skadliga appar. Apparna har också granskats av Apple som kontrollerar att de generellt fungerar enligt beskrivningen och inte innehåller några uppenbara buggar eller andra större problem. Utöver de tekniska aspekterna gör den här urvalsprocessen att användarna får förtroende för att de appar de köper håller hög kvalitet.
- *Validering av kodsignatur:* Med iOS och iPadOS kan utvecklare bädda in ramverk inuti apparna. De här ramverken kan användas av appen själv eller av tillägg som är inbäddade i appen. För att skydda systemappar och andra appar mot inläsning av kod från tredje part inom deras adressutrymme kontrollerar systemet kodsignaturerna för alla dynamiska bibliotek som någon process länkar till under start. Den här kontrollen genomförs med hjälp av teamidentifieraren (Team ID) som utvinns ur ett certifikat som har utfärdats av Apple. En teamidentifierare är en alfanumerisk sträng på tio tecken – till exempel 1A2B3C4D5F. Ett program kan länka till vilket plattformsbibliotek som helst som följer med systemet, eller vilket annat bibliotek som helst som har samma teamidentifierare som primär körbar kod i sin kodsignatur. Eftersom den körbara kod som ingår i systemet inte har någon teamidentifierare kan den endast länka till de bibliotek som är inbyggda i systemet.

### Verifiering av företagsägda interna appar

Behöriga företag har möjlighet att utveckla företagsägda interna appar för användning i organisationen och distribuera dem till sina anställda. Företag och organisationer kan ansöka till Apple Developer Enterprise Program (ADEP). Mer information och behörighetskrav finns på webbplatsen för Apple Developer Enterprise Program. När en organisation blir medlem i ADEP kan den registrera sig och erhålla en tillhandahållandeprofil som gör det möjligt att köra företagsägda interna appar på enheter som organisationen auktoriserar.

Användarna måste ha tillhandahållandeprofilen installerad för att kunna köra de här apparna. Detta garanterar att endast behöriga användare inom organisationen kan läsa in apparna på sin iPhone eller iPad. Appar som installeras via MDM är implicit betrodda eftersom relationen mellan organisationen och enheten redan är upprättad. I annat fall måste användarna godkänna appens tillhandahållandeprofil i Inställningar. Organisationer kan även förhindra att användare godkänner appar från okända utvecklare. Första gången en företagsägd intern app startas måste enheten få en positiv bekräftelse från Apple om att appen tillåts att köras.

## Säkerhet vid körning i iOS och iPadOS

iOS och iPadOS säkerställer säkerhet under körning med hjälp av en ”sandlåda”, fastställda behörigheter och ASLR (Address Space Layout Randomization).

### Körning i sandlåda

Alla tredjepartsappar körs i en ”sandlåda”, vilket innebär att de inte har tillgång till filer som har sparats av andra appar och inte kan göra några ändringar på enheten. Sandlådor är utformade för att förhindra att appar samlar in eller ändrar information som har sparats av andra appar. Varje apps filer sparas i en unik hemkatalog som tilldelas slumpmässigt när appen installeras. Om en tredjepartsapp behöver tillgång till information utifrån kan den endast få det via tjänster som uttryckligen tillhandahålls av iOS och iPadOS.

Systemfiler och systemresurser är också skyddade från användarnas appar. De flesta iOS- och iPadOS-systemfiler och resurser körs, liksom alla tredjepartsappar, som användaren ”mobile” som saknar behörigheter. Hela operativsystemets partition är inlänkad som skrivskyddad. Verktyg som inte behövs, till exempel tjänster för fjärrinloggning, inkluderas inte i systemprogramvaran och API:er tillåter inte att appar utökar sina egna behörigheter så att de kan modifiera andra appar eller själva iOS och iPadOS.

### Användning av behörigheter

Tredjepartsappars tillgång till användarinformation och till funktioner som iCloud och utökning genom tillägg styrs genom fastställda behörigheter. Behörigheterna är par av nyckelvärden som skrivs in i en app och tillåter autentisering utöver faktorer vid körning, som UNIX-användar-ID. Eftersom behörigheter signeras digitalt kan de inte ändras. Behörigheter används ofta av systemappar och bakgrundsprocesser till att utföra specifika uppgifter som kräver behörighet och som annars skulle kräva att processen kördes som rotanvändare. Detta minskar risken för att eventuella systemappar eller bakgrundsprocesser som har manipulerats ska utöka sina behörigheter.

Dessutom kan appar bara utföra bakgrundsbearbetningar genom system-API:er. Det gör att appar kan fortsätta köras utan att prestanda eller batteritid påverkas nämnvärt.

### ASLR (Address Space Layout Randomization)

ASLR (Address Space Layout Randomization) skyddar mot att minnesfelsbuggar utnyttjas. Inbyggda appar använder ASLR till att tilldela alla minnesregioner en slumpmässig plats vid start. Utöver att ASLR används vid start ordnas minnesadresserna för körbar kod, systembibliotek och sammanhörande programmeringsstrukturer slumpmässigt, vilket minskar sannolikheten för många angrepp. En return-to-libc-attack försöker till exempel lura enheter att köra skadlig kod genom att manipulera stackens och systembibliotekens minnesadresser. Om placeringen av minnesadresserna är slumpmässig är den här typen av angrepp svårare att genomföra, särskilt mot flera enheter samtidigt. Xcode, och utvecklingsmiljöerna för iOS eller iPadOS, aktiverar ASLR automatiskt vid kompilering av tredjepartsprogram.

### Execute Never-funktion

Ytterligare skydd tillhandahålls av iOS och iPadOS genom ARM:s Execute Never-funktion (XN) som markerar minnessidor som icke-körbara. Minnessidor som både har markerats som skrivbara och körbara kan endast användas av appar under strängt kontrollerade former: Kärnan söker efter Apples särskilda behörighet till dynamisk kodsignering. Även om den hittas kan endast ett `mmap`-anrop göras för att begära en körbar och skrivbar sida, vilken tilldelas en slumpgenererad adress. Safari använder den här funktionen i sin JIT (just-in-time)-kompilator för JavaScript.

## Stöd för tillägg i iOS, iPadOS och macOS

Appar kan utöka funktionerna hos andra appar i iOS, iPadOS och macOS genom tillägg. Tillägg är signerade körbara binärfiler för speciella ändamål som paketeras inuti en app. Under installation upptäcker systemet tillägg automatiskt och gör dem tillgängliga för andra appar genom ett matchningssystem.

### Tilläggspunkter

Ett systemområde som har stöd för tillägg kallas en *tilläggspunkt*. Varje tilläggspunkt tillhandahåller API:er och ser till att policyer följs för det området. Systemet avgör vilka tillägg som är tillgängliga utifrån specifika matchningsregler för respektive tilläggspunkt. Systemet startar tilläggsprocesser automatiskt vid behov och ser till att de är aktiva under den tid som behövs. Behörigheter kan användas till att begränsa tillgängligheten för tillägg till specifika systemappar. Exempelvis visas en widget för vyn Idag bara i Notiscenter, medan ett tillägg för delning bara visas på panelen Delning. Exempel på tilläggspunkter är Idag-widgetar, delning, åtgärder, bildredigering, filleverantör och anpassat tangentbord.

### Hur tillägg kommunicerar

Tilläggen körs i sitt eget adressutrymme. Vid kommunikation mellan tillägget och appen det har aktiverats från används interprocesskommunikation som förmedlas via systemramverket. De har inte tillgång till varandras filer eller minnesutrymmen. Tillägg har utformats för att vara isolerade från varandra, från apparna som innehåller dem och från apparna som använder dem. De körs i en sandlåda, precis som andra tredjepartsappar, och har en behållare som är åtskild från behållaren för den app de finns i. De har dock samma tillgång till integritetsinställningar som appen som innehåller dem. Så om en användare ger Kontakter tillgång till en app omfattar tillgången också de tillägg som är inbäddade i appen, men inte de tillägg som aktiveras av appen.

### Hur anpassade tangentbord används

Anpassade tangentbord är en speciell typ av tillägg som användaren aktiverar för hela systemet. När ett tangentbordstillägg har aktiverats används det för alla textfält, med undantag för lösenkodsinmatningen och eventuella säkra textvyer. För att begränsa överföringen av användardata körs anpassade tangentbord normalt i en mycket restriktiv sandlåda som blockerar tillgång till nätverket, till tjänster som utför nätverksåtgärder för processers räkning och till API:er som skulle kunna ge tillägget obehörig tillgång till data som skrivs på tangentbordet. Utvecklare av anpassade tangentbord kan begära att deras tillägg får så kallad öppen tillgång (Open Access), vilket innebär att systemet får köra tillägget i den förvalda sandlådan efter användarens medgivande.

### MDM och tillägg

För enheter som är registrerade i en MDM-lösning styrs dokument- och tangentbordstillägg av reglerna för Managed Open In. Till exempel kan MDM-lösningen förhindra att användarna exporterar ett dokument från en hanterad app till en ohanterad dokumentapp eller använder ett ohanterat tangentbord med en hanterad app. Dessutom kan apputvecklare förhindra att tangentbordstillägg från tredje part används i deras app.

## Appskydd och appgrupper i iOS och iPadOS

I iOS och iPadOS kan organisationen skydda appar med IOS SDK och genom att gå med i en appgrupp i Apple Developer-portalen.

### Använda dataskydd i appar

iOS Software Development Kit (SDK) för iOS och iPadOS innehåller en komplett uppsättning API:er som gör det enkelt för tredjepartsutvecklare och internutvecklare på företag att använda dataskydd och göra sina appar så säkra som möjligt. Dataskydd finns för fil- och databas-API:er, som NSFileManager, CoreData, NSData och SQLite.

Mail-appens databas (inklusive bilagor), hanterade böcker, Safari-bokmärken, appstartbilder och platsdata lagras också i krypterad form med nycklar som skyddas av användarens lösenkod på enheten. Kalender (exklusive bilagor), Kontakter, Påminnelser, Anteckningar, Meddelanden och Bilder använder `Data Protection`-behörigheten *Protected Until First User Authentication*.

Appar som installeras av användaren, utan att de har en angiven dataskyddsklass, använder Protected Until First User Authentication som förval.

### Ansluta till en appgrupp

Appar och tillägg som ägs av ett och samma utvecklarkonto kan dela innehåll om de konfigureras så att de ingår i samma appgrupp. Det är upp till utvecklaren att skapa rätt grupper på Apples utvecklarportal och inkludera den önskade uppsättningen appar och tillägg. En app som har konfigurerats för att ingå i en viss appgrupp har tillgång till följande:

- En delad lagringsbehållare på volymen. Behållaren sparas på enheten så länge som minst en app från gruppen är installerad.
- Delade inställningar.
- Delade nyckelringsobjekt.

Apples utvecklarportal säkerställer att grupp-ID:n (GID:n) för appar är unika i hela appekosystemet.

# Appsäkerhet och macOS

## Introduktion till appsäkerhet för macOS

Appsäkerhet i macOS består av ett antal överlappande lager – varav det första är alternativet att köra bara signerade och betrodda appar från App Store. Därtill finns flera skyddslager i macOS som säkerställer att appar som hämtas från internet inte innehåller kända sabotageprogram. macOS innehåller även tekniker för att upptäcka och ta bort sabotageprogram samt ytterligare skydd som är avsedda att förhindra obetrodda appar från att komma åt användardata. Apple-tjänster som attestering och XProtect-uppdateringar är utformade för att förhindra installation av sabotageprogram. Vid behov lokaliserar de här tjänsterna sabotageprogram som till en början har undgått upptäckt och tar sedan snabbt och effektivt bort dem. Till syvende och sist står det macOS-användare fritt att använda den säkerhetsmodell som passar dem bäst – inklusive att köra helt osignerad och obetrodd kod.

## Kodsigneringsprocess för appar i macOS

Alla appar från App Store har signerats av Apple. Signeringen är utformad för att garantera att apparna inte har manipulerats eller ändrats. Apple signerar alla appar som levereras med Apple-enheter.

I macOS 10.15 måste alla appar som distribueras utanför App Store vara signerade av utvecklaren med ett Apple-utfärdat utvecklar-ID-certifikat (i kombination med en privat nyckel) och även vara attesterade av Apple för körning med de förvalda Gatekeeper-inställningarna. Även appar som utvecklas internt bör vara signerad med ett Apple-utfärdat utvecklar-ID så att användare kan validera dess integritet.

I macOS körs kodsignering och attestering oberoende av varandra, och kan utföras av olika aktörer, för olika syften. Kodsignering utförs av utvecklaren som använder sitt utvecklar-ID-certifikat (utfärdas av Apple). Verifiering av den här signaturen bevisar för användaren att en utvecklares programvara inte har manipulerats sedan utvecklaren byggde och signerade den. Attestering kan utföras av vem som helst i distributionskedjan för programvaran och är ett bevis för att Apple har fått en kopia av koden för att kontrollera om den innehåller sabotageprogram samt att inget känt sabotageprogram har identifierats. Resultatet av attesteringen är en biljett som lagras på Apples servrar och går att fästa vid appen (av vem som helst) utan att utvecklarens signatur blir ogiltig.

MAC (Mandatory Access Controls) kräver kodsignering för att aktivera behörigheter som skyddas av systemet. Exempelvis måste appar som kräver anslutning genom brandväggen vara kodsignerade med lämpliga MAC-behörigheter.

# Gatekeeper och säkerhet vid användning i macOS

macOS erbjuder Gatekeeper-teknik och körningsskydd som säkerställer att endast betrodd programvara körs på en användares dator.

## Gatekeeper

macOS innehåller en säkerhetsteknik som kallas *Gatekeeper*. Den är utformad för att säkerställa att endast betrodd programvara körs på en användares dator. När en användare hämtar och öppnar en app, en insticksfil eller ett installationspaket från något annat ställe än App Store verifierar Gatekeeper att programvaran kommer från en identifierad utvecklare, har attesterats av Apple som fri från skadligt innehåll samt inte har ändrats. Gatekeeper begär också godkännande från användaren innan hämtad programvara öppnas första gången för att säkerställa att användaren inte har lurats att starta körbar kod i tron att det var en vanlig datafil. Gatekeeper håller också koll på historiken för filer som skrivs av hämtad programvara.

Som förval säkerställer Gatekeeper att all hämtad programvara har signerats av App Store eller av en registrerad utvecklare och har attesterats av Apple. Både förhandsgranskningsprocessen i App Store och attesteringsfunktionen är utformade för att säkerställa att appar inte innehåller några kända sabotageprogram. Därför *blir all programvara i macOS som förval genomsökt efter skadligt innehåll första gången den öppnas, oavsett hur den har hamnat på datorn*.

Användare och organisationer kan välja att bara tillåta installation av programvara från App Store. Användare kan också välja att förbigå Gatekeeper-policyer och öppna alla programvaror, förutsatt att det inte begränsas av en MDM-lösning. Organisationer kan använda MDM till att konfigurera Gatekeeper-inställningar, inklusive att tillåta programvara som är signerad med alternativa identiteter. Om det behövs kan Gatekeeper också avaktiveras helt och hållet.

Gatekeeper skyddar också mot att skadliga insticksfiler distribueras tillsammans med ofarliga appar. I dessa fall kan användningen av appen leda till att en skadlig insticksfil läses in utan att användaren vet om det. När det behövs öppnar Gatekeeper appar från slumpmässiga, skrivskyddade platser. Det ska förhindra att insticksfiler som distribueras tillsammans med appen läses in.

## Säkerhet vid användning

Systemfiler, resurser och operativsystemets kärna hålls skyddade från användarens apputrymme. Alla appar från App Store körs i en så kallad sandlåda som begränsar tillgången till data som har lagrats av andra appar. Om en app från App Store behöver komma åt en annan apps data måste den använda sig av godkända API:er och tjänster i macOS.

# Skydd mot sabotageprogram i macOS

Apple använder en process som drar nytta av den senaste informationen om nya hot till att snabbt identifiera och blockera sabotageprogram.

## Tre försvarslager

Försvaret mot sabotageprogram är strukturerat i tre lager:

1. *Förhindra start eller körning av sabotageprogram:* App Store eller Gatekeeper i kombination med attestering

2. *Blockera sabotageprogram så att de inte kan köras på kundsystem:* Gatekeeper, attestering och XProtect

3. *Åtgärda sabotageprogram som har körts:* XProtect

Det första försvarslagret är utformat för att förhindra distributionen av sabotageprogram och förhindra att det startar ens en enda gång – detta är målet med App Store och Gatekeeper i kombination med attestering.

Nästa försvarslager är att säkerställa att ett sabotageprogram som hamnar på en Mac snabbt identifieras och blockeras, både för att stoppa spridningen och för att åtgärda de Mac-system som det redan har fått fotfäste i. XProtect bidrar till den här typen av försvar tillsammans med Gatekeeper och attestering.

Slutligen agerar XProtect för att åtgärda sabotageprogram som ändå har körts.

Dessa skydd, som beskrivs ytterligare nedan, ger tillsammans bästa möjliga skydd från virus och sabotageprogram. Det finns ytterligare skydd, framförallt på Mac-datorer med Apple Silicon, för att begränsa den potentiella skadan orsakad av sabotageprogram där körningen lyckats. Se Skydda apptillgång till användardata för olika sätt som macOS kan skydda användardata mot sabotageprogram, och Operativsystemets integritet för olika sätt som macOS kan begränsa de åtgärder som sabotageprogram kan utföra i systemet.

## Attestering

*Attestering* är en tjänst för sabotageprogramskanning som tillhandahålls av Apple. Utvecklare som vill distribuera appar för macOS utanför App Store skickar in sina appar för skanning som en del av distributionsprocessen. Apple skannar programvaran efter kända skadeprogram, och om inga hittas utfärdas en attesteringsbiljett. Normalt fäster utvecklarna denna biljett vid appen så att Gatekeeper kan verifiera och starta appen, även i nedkopplat läge.

Apple kan också utfärda en återkallelsebiljett för appar som är kända för att vara skadliga – även om de tidigare har attesterats. macOS kontrollerar regelbundet om det finns nya återkallelsebiljetter så att Gatekeeper har den senaste informationen och kan blockera starten av sådana filer. Den här processen kan mycket snabbt blockera skadliga appar eftersom uppdateringar sker i bakgrunden betydligt oftare än till och med de bakgrundsuppdateringar som hämtar nya XProtect-signaturer. Dessutom kan det här skyddet användas både på appar som tidigare har attesterats och på appar som inte har det.

## XProtect

macOS innehåller en inbyggd antivirusteknik kallad *XProtect* för den signaturbaserade identifieringen och borttagningen av sabotageprogram. Systemet använder YARA-signaturer. Det är ett verktyg som används till att utföra signaturbaserad igenkänning av sabotageprogram och uppdateras regelbundet av Apple. Apple övervakar sabotageprograms spridningar och strängar och uppdaterar signaturer automatiskt, fristående från systemuppdateringar, för att skydda Mac-datorer från sabotageprogram. XProtect upptäcker automatiskt och stoppar körningen av kända sabotageprogram. I macOS 10.15 och senare söker XProtect efter känt skadligt innehåll varje gång:

- En app startas första gången
- En app har ändrats (i filsystemet)
- XProtect-signaturer uppdateras

När XProtect upptäcker sabotageprogram blockeras programvaran, användaren får ett meddelanden och ges möjlighet att flytta programvaran till papperskorgen.

*Obs!* Attestering är effektivt mot kände filer (eller filhasher) och kan användas på appar som har startats tidigare. De signaturbaserade reglerna i XProtect är mer allmänna än en specifik filhash, så funktionen kan hitta varianter som Apple inte har sett. XProtect söker endast igenom appar som har ändrats eller appar vid den första körningen.

Skulle ett sabotageprogram ändå ta sig in på en Mac innehåller XProtect dessutom teknik som rensar infektioner. Den innehåller exempelvis en motor som rensar infektioner baserat på automatiska leveranser av uppdateringar från Apple (som en del av automatiska uppdateringar av systemdatafiler och säkerhetsuppdateringar). Det här systemet tar bort sabotageprogram när uppdaterad information tas emot och fortsätter att periodvis leta efter infektioner, men XProtect startar inte om datorn automatiskt. Utöver detta innehåller XProtect en avancerad motor som upptäcker okända sabotageprogram baserat på beteendeanalys. Information om sabotageprogram som upptäckts av denna motor, inklusive vilken programvara som var ansvarig för att hämta den, används till att förbättra XProtect-signaturer och macOS-säkerhet.

## Automatiska XProtect-säkerhetsuppdateringar

Apple utfärdar automatiskt uppdateringar för XProtect baserade på den senaste informationen om nya hot. Som förval kontrollerar macOS dagligen om sådana uppdateringar finns. Attesteringsuppdateringar som distribueras via CloudKit-synkronisering är mycket mer frekventa.

## Hur Apple svarar när ny sabotageprogramvara upptäcks

När nya sabotageprogram upptäcks kan ett antal steg utföras:

- Alla tillhörande utvecklar-ID-certifikat återkallas.
- Biljetter för återkallande av attesteringar utfärdas för alla filer (appar och tillhörande filer).
- XProtect-signaturer utvecklas och släpps.

  Dessa signaturer används också retroaktivt på tidigare attesterad programvara. Eventuella nya upptäckter kan leda till att en eller flera av åtgärderna ovan utförs.

Sammanfattningsvis startas en serie steg över de närmast sekunderna, timmarna och dagarna efter upptäckten av sabotageprogram för att sprida bästa möjliga skydd till Mac-användare.

## Styra appåtkomst till filer i macOS

Apple anser att användarna ska ha full överblick, godkännanderätt och kontroll över vad appar gör med deras data. I macOS 10.15 drivs den här modellen igenom av systemet för att säkerställa att alla appar måste få godkänt av användaren innan de kan komma åt filer i mapparna Dokument och Hämtade filer, på skrivbordet, i iCloud Drive och på nätverksvolymer. I macOS 10.13 och senare måste appar som kräver åtkomst till hela lagringsenheten uttryckligen läggas till i Systeminställningar. Dessutom kräver funktioner för åtkomst och automatisering användarens godkännande för att säkerställa att de inte kringgår andra skydd. Beroende på åtkomstpolicy kan användare bli ombedda eller tvungna att ändra inställningen i:

- macOS 13 eller senare: Systeminställningar > Integritet och säkerhet > Integritet.
- macOS 12 eller tidigare: Systeminställningar > Säkerhet och integritet > Integritet.

| Objekt | Användare ombedd av app | Användaren måste redigera integritetsinställningarna för systemet |
|---|---|---|
| Åtkomst | ✗ | ✓ |
| Fullständig åtkomst till intern lagring | ✗ | ✓ |
| Filer och mappar<br>*Obs!* Omfattar mappen Skrivbord, mappen Dokument, mappen Hämtade filer, nätverksvolymer och borttagbara volymer | ✓ | ✗ |
| Automatisering (Apple-händelser) | ✓ | ✗ |

En användare som slår på FileVault på en Mac blir ombedd att ange giltiga inloggningsuppgifter innan startprocessen fortsätter och för att få tillgång till specialiserade startlägen. Utan giltiga inloggningsuppgifter eller en återställningsnyckel fortsätter hela volymen att vara krypterad. Den är skyddad mot obehörig åtkomst även om den fysiska lagringsenheten tas bort och ansluts till en annan dator.

För att skydda data i en företagsmiljö bör IT-ansvariga tydligt definiera och genomdriva FileVault-konfigurationspolicyer via en MDM-lösning. Organisationer har flera alternativ för hantering av krypterade volymer som omfattar institutionella återställningsnycklar, personliga återställningsnycklar (som även kan lagras med MDM för deponering) eller en kombination av båda. Nyckelrotation kan också ställas in som en policy i MDM.

# Säkerhetsfunktioner i appen Anteckningar

I Anteckningar finns en funktion för säkra anteckningar – på iPhone, iPad, Mac och iCloud-webbplatsen – som användare kan använda till att skydda innehållet i särskilda anteckningar. Användare kan också säkert dela anteckningar med andra.

## Säkra anteckningar

Säkra anteckningar krypteras heltäckande med en lösenfras som användaren anger och som sedan måste anges för att visa anteckningarna på iOS-, iPadOS- och macOS-enheter samt på iCloud-webbplatsen. Varje iCloud-konto (inklusive enhetskonton av typen På min) kan ha en separat lösenfras.

När en användare skyddar en anteckning härleds en nyckel på 16 byte från användarens lösenfras med hjälp av PBKDF2 och SHA256. Anteckningen och alla dess bilagor krypteras med AES-GCM (Galois/Counter Mode). Nya poster skapas i Core Data och CloudKit för att lagra den krypterade anteckningen, bilagor, taggen och initieringsvektorn. När de nya posterna har skapats raderas alla ursprungliga, okrypterade data. Bilagor som har stöd för kryptering är bland annat bilder, skisser, tabeller, kartor och webbplatser. Anteckningar som innehåller andra slags bilagor kan inte krypteras, och bilagor som inte stöds kan inte läggas till i säkra anteckningar.

För att visa en säker anteckning måste användaren ange sin lösenfras eller autentisera med Face ID eller Touch ID. När användaren har autentiserats korrekt, antingen för att visa eller skapa en säker anteckning, öppnar Anteckningar en säker session. När den säkra sessionen är öppen kan användaren visa eller säkra andra anteckningar utan någon ytterligare autentisering. Den säkra sessionen gäller dock bara för de anteckningar som skyddas med den angivna lösenfrasen. Användaren måste fortfarande autentisera sig för anteckningar som skyddas av en annan lösenfras. Den säkra sessionen stängs när:

- Användaren trycker på knappen Lås nu i Anteckningar
- Anteckningar placeras i bakgrunden under mer än tre minuter (åtta minuter i macOS)
- iOS- eller iPadOS-enheten låses

För att ändra lösenfrasen för en säker anteckning måste användaren ange den aktuella lösenfrasen eftersom Face ID och Touch ID inte är tillgängliga vid byte av lösenfras. Efter att ha valt en ny lösenfras paketerar Anteckningar om nycklarna för alla befintliga anteckningar som är krypterade med den tidigare lösenfrasen i samma konto.

Om en användare skriver fel lösenfras tre gånger i rad visar Anteckningar ett tips som användaren har angett (om användaren har ställt in ett sådant). Om användaren fortfarande inte kommer ihåg lösenfrasen går det att skapa en ny i inställningarna för Anteckningar. Med den här funktionen kan användare skapa nya säkra anteckningar med en ny lösenfras, men de kan inte se de anteckningar som har skyddats tidigare. Det går fortfarande att visa de tidigare skyddade anteckningarna om du kommer ihåg den gamla lösenfrasen. Användarens lösenfras för iCloud-kontot krävs för att skapa en ny lösenfras.

## Delade anteckningar

Anteckningar som inte är heltäckande krypterade med en lösenfras kan delas med andra. Delade anteckningar använder fortfarande den krypterade CloudKit-datatypen för text eller bilagor som användaren placerar i en anteckning. Resurser krypteras alltid med en nyckel som är krypterad i CKRecord. Metadata, som skapelse- och ändringsdatum, krypteras inte. CloudKit hanterar processen genom vilken deltagarna kan kryptera och avkryptera varandras data.

# Säkerhetsfunktioner i appen Genvägar

Du kan välja att synkronisera genvägar i Genvägar med andra Apple-enheter med hjälp av iCloud. Genvägar kan också delas med andra användare via iCloud. Genvägar lagras lokalt i ett krypterat format.

Anpassade genvägar är mångsidiga – de liknar skript eller program. Vid hämtning av genvägar från internet varnas användaren om att genvägen inte har granskats av Apple och användaren ges möjlighet att granska genvägen. Som skydd mot skadliga genvägar hämtas uppdaterade definitioner för sabotageprogram så att bedrägliga genvägar kan identifieras vid körning.

Anpassade genvägar kan också köra användarspecificerade JavaScript-skript på webbplatser i Safari när de anropas från delningsbladet. För att skydda mot bedrägliga JavaScript-skript, som exempelvis lurar användaren att köra ett skript på en social mediewebbplats för att stjäla data, valideras JavaScript-skriptet mot de tidigare nämnda definitionerna för sabotageprogram. Första gången en användare kör JavaScript på en domän blir användaren uppmanad att tillåta att genvägar som innehåller JavaScript körs på den aktuella webbsidan för den domänen.

# Tjänstesäkerhet

## Tjänstesäkerhet i översikt

Apple har skapat en stabil uppsättning tjänster som hjälper användarna att få ut ännu mer funktion och produktivitet ur sina enheter. De här tjänsterna ger kraftfulla funktioner för molnlagring, synkronisering, lösenordslagring, autentisering, betalningar, meddelanden, kommunikation med mera, samtidigt som användarnas integritet och data skyddas.

Det här kapitlet omfattar säkerhetstekniker som används för iCloud, Logga in med Apple, Apple Pay, iMessage, Apple Messages for Business, FaceTime, Hitta och Kontinuitet.

*Obs!* Alla Apples tjänster och innehåll är inte tillgängliga i alla länder eller regioner.

## Apple-ID och hanterat Apple-ID

### Säkerhet och Apple-ID i översikt

Ett Apple-ID är det konto som används till att logga in till Apple-tjänster. Det är viktigt att användarna skyddar sina Apple-ID:n så att ingen obehörig får tillgång till deras konton. Som hjälp för detta kräver Apple-ID:n starka lösenord som:

- måste vara minst åtta tecken långa
- måste innehålla både bokstäver och siffror
- inte får innehålla tre eller fler på varandra följande likadana tecken
- inte är ett vanligt använt lösenord

Användarna uppmuntras att gå ännu längre än riktlinjerna kräver genom att lägga till extra tecken och skiljetecken som gör lösenorden ännu starkare.

Apple meddelar också användarna via e-post eller pushnotiser, eller både och, när viktiga ändringar sker i deras konton – exempelvis om lösenordet eller faktureringsinformationen ändras, eller om deras Apple-ID har använts till att logga in på en ny enhet. Om användaren inte känner igen ändringarna uppmanas denna att genast byta lösenord till sitt Apple-ID.

Dessutom har Apple en mängd olika policyer och rutiner som har utformats för att skydda användarkonton. Detta innebär bland annat att antalet inloggningsförsök och försök att skapa ett nytt lösenord begränsas, att bedrägeriförsök övervakas mer aktivt så att angrepp kan identifieras när de inträffar och att policyer regelbundet granskas så att Apple kan anpassa sig efter eventuell ny information som kan påverka användarnas säkerhet.

*Obs!* Lösenordspolicyn för ett hanterat Apple-ID ställs in av en administratör i Apple School Manager eller Apple Business Manager.

### Tvåfaktorsautentisering

För att hjälpa användare att skydda sina konton ytterligare använder Apple som förval *tvåfaktorsautentisering* som är ett extra skyddslager för Apple-ID:n. Syftet med tvåfaktorsautentisering är att se till att bara kontots ägare kan komma åt kontot även om någon annan kan lösenordet. Med tvåfaktorsautentisering går det bara att komma åt en användares konto på betrodda enheter, till exempel användarens iPhone, iPad eller Mac, eller på andra enheter efter att ha genomfört en verifiering via en av de betrodda enheterna eller ett betrott telefonnummer. Första gången du ska logga in på en ny enhet måste du ange två saker: lösenordet till Apple-ID:t och en sexsiffrig verifieringskod som visas på användarens betrodda enheter eller som skickas till ett betrott telefonnummer. Genom att ange koden bekräftar användaren att den nya enheten är tillförlitlig och att det är säkert att logga in. Eftersom det inte längre räcker med ett lösenord för att komma åt en användares konto gör tvåfaktorsautentiseringen att användarens Apple-ID och all personlig information som användaren har sparat hos Apple är bättre skyddade. Funktionen är inbyggd i iOS, iPadOS, macOS, tvOS, watchOS och autentiseringssystemen som används på Apples webbplatser.

När en användare loggar in på en Apple-webbplats med en webbläsare skickas en andra faktorförfrågan till alla betrodda enheter som är associerade till användarens iCloud-konto med en begäran om att godkänna webbsessionen. Om användaren loggar in på en Apple-webbplats via en webbläsare på en betrodd enhet visas verifieringskoden lokalt på den enhet som används. När användaren skriver in koden på den enheten godkänns webbsessionen.

### Nya lösenord och kontoåterställning

Om en användare glömmer bort lösenordet för sitt Apple-ID-konto kan användaren skapa ett nytt på en betrodd enhet. Om en betrodd enhet inte är tillgänglig, och lösenordet är känt, kan ett betrott telefonnummer användas till autentisering genom en SMS-verifiering. Dessutom kan en tidigare använd lösenkod tillsammans med SMS-verifiering användas till att omedelbart skapa ett nytt Apple-ID. Om de här alternativen inte kan användas måste rutinen för kontoåterställning följas. Mer information finns i Apple Support-artikeln Så här använder du kontoåterställning när du inte kan nollställa ditt Apple-ID-lösenord.

## Säkerhet och hanterade Apple-ID:n

Hanterade Apple-ID:n fungerar på ungefär samma sätt som ett Apple-ID, men de ägs och hanteras av ett företag eller en utbildningsorganisation. De här organisationerna kan skapa nya lösenord och stänga av kommunikation via t.ex. FaceTime och Meddelanden samt ställa in rollbaserade behörigheter för anställda, personal, lärare och elever.

För hanterade Apple-ID:n är vissa tjänster avaktiverade (till exempel App Store, HomeKit och Hitta).

## Åtkomsthantering för hanterade Apple-ID:n

Organisationer kan använda åtkomsthanteringen i Apple Business Manager, Apple School Manager och Apple Business Essentials till att definiera var hanterade Apple-ID:n kan användas och vilka tjänster som är tillgängliga för dem.

Med åtkomsthantering kan du definiera om användare får logga in med ett hanterat Apple-ID på valfri enhet, endast på hanterade enheter eller endast på hanterade och övervakade enheter. Administratörer kan även konfigurera om användare får logga in på iCloud på webben. Det här gör det möjligt för organisationer att använda hanteringsstatusen för enheten som en faktor vid beslutet om åtkomst ska tillåtas till organisationsdata.

Dessutom kan administratörer definiera vilka iCloud-tjänster som är tillgängliga för deras användare. Det omfattar att definiera åtkomsten till Apple Developer-program och betaprogrammet AppleSeed for IT samt att avgöra om användare får åtkomst till Apples integritetsportal på privacy.apple.com.

Hanterade Apple-ID:n stöder även samarbeten på dokument med Keynote, Numbers, Pages, Påminnelser och Anteckningar liksom kommunikation med FaceTime och iMessage. För de tjänsterna kan organisationer definiera om användare kan samarbeta med vem som helst eller bara med konton som har skapats inom samma organisation med Apple School Manager, Apple Business Manager eller Apple Business Essential.

Om reglerna för åtkomsthantering förändras återspeglas de på enheterna där användaren är inloggad med sitt hanterade Apple-ID. Om kraven för hanteringsstatus på en enhet förändras blir ett hanterat Apple-ID automatiskt utloggat från en enhet om enhetens status inte uppfyller de nya kraven.

## Granska hanterade Apple-ID:n

Hanterade Apple-ID:n som har skapats i Apple School Manager har också stöd för *granskning* så att organisationer kan följa juridiska regler och integritetsregler. En användare med rollen som administratör, platsansvarig, personansvarig eller lärare kan granska specifika hanterade Apple-ID-konton.

Granskare kan endast övervaka konton som ligger under dem i organisationens hierarki. Det innebär att lärare kan övervaka elever, att ansvariga kan granska lärare och elever och att administratörer kan granska ansvariga, lärare och elever.

När inloggningsuppgifter för granskning begärs i Apple School Manager skapas ett särskilt konto som endast kan användas för att komma åt det hanterade Apple-ID som angavs när granskningen begärdes. Granskaren kan läsa och ändra användarens innehåll som har sparats på iCloud eller i appar med CloudKit aktiverat. Varje begäran om granskningsåtkomst registreras i Apple School Manager. I loggarna finns information om vem som var granskare, vilket hanterat Apple-ID granskaren begärde åtkomst till, tidpunkten för begäran och om granskningen har genomförts eller inte.

# iCloud

## iCloud-säkerhet i översikt

iCloud lagrar användarens kontakter, kalendrar, bilder, dokument med mera och håller informationen uppdaterad på alla användarens enheter, helt automatiskt. iCloud kan också användas av tredjepartsappar till att lagra och synkronisera dokument och viktiga appdata som definieras av utvecklaren. Användare ställer in iCloud genom att logga in med ett Apple-ID och välja vilka tjänster de vill använda. Vissa iCloud-funktioner, till exempel iCloud Drive och iCloud-säkerhetskopiering, kan avaktiveras av IT-administratörer via MDM-konfigurationsprofiler.

iCloud använder starka säkerhetsmetoder och har strikta policyer för att skydda användardata. Huvuddelen av iCloud-data krypteras först på användarens enhet med enhetsgenererade iCloud-nycklar innan de överförs till iCloud. För data som inte är heltäckande krypterade överför användarens enhet dessa iCloud-nycklar på ett säkert sätt till iCloud Hardware Security Modules i Apples datacenter. Detta tillåter att Apple hjälper användaren med dataåterställning och avkrypterar dessa data å användarens vägnar när den behöver det (till exempel när användaren loggar in på en ny enhet, återskapar från en säkerhetskopia eller ansluter till sina iCloud-data på webben). Data som flyttas mellan användarens enhet och iCloud-servrar krypteras separat vid överföring med TLS och iCloud-servrar lagrar användardata med ett ytterligare krypteringslager.

När Apple har tillgång till krypteringsnycklar skyddas dessa i Apples datacenter. Vid bearbetning av data som lagras i tredje parts datacenter används dessa krypteringsnycklar endast av Apple-programvara som körs på säkra servrar och endast medan den nödvändiga bearbetningen utförs. För ytterligare integritet och säkerhet använder många Apple-tjänster heltäckande kryptering, vilket innebär att endast användare kan komma åt sina iCloud-data och endast från betrodda enheter där de är inloggade med sitt Apple-ID.

Apple erbjuder användare två alternativ att kryptera och skydda data de lagrar på iCloud:

- **Standarddataskydd (den förvalda inställningen):** Användarens iCloud-data krypteras, krypteringsnycklarna skyddas i Apples datacenter och Apple kan hjälpa till med data- och kontoåterställning. Endast vissa iCloud-data – 14 datakategorier, inklusive hälsodata och lösenord i iCloud-nyckelring – är heltäckande krypterade.
- **Avancerat dataskydd för iCloud:** En valfri inställning som erbjuder Apples högsta säkerhetsnivå för molndata. Om en användare väljer att slå på Avancerat dataskydd har endast användarens betrodda enheter tillgång till krypteringsnycklarna för huvuddelen av dennas iCloud-data, vilka därmed skyddas med heltäckande kryptering. När användare slår på Avancerat dataskydd ökar antalet kategorier som använder heltäckande kryptering till 23 och omfattar iCloud-säkerhetskopiering, Bilder, Anteckningar med mera.

De specifika kategorierna iCloud-data som skyddas med heltäckande kryptering listas i Apple Support-artikeln Översikt över iCloud-datasäkerhet.

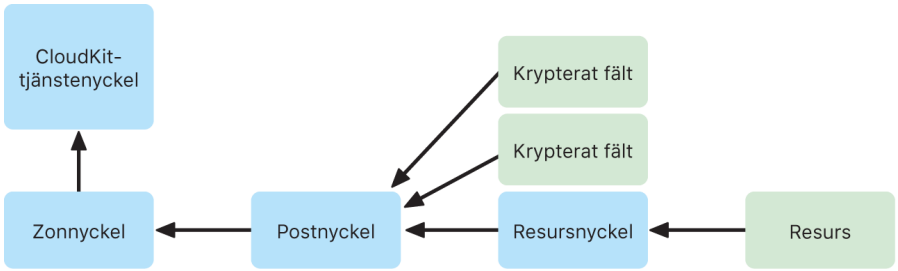
# iCloud-kryptering

Datakryptering på iCloud är nära knuten till datalagringsmodellen som utgår från CloudKit-ramverken och API:erna som gör det möjligt för appar och systemprogramvara att lagra data på iCloud å användarens vägnar samt håller allt uppdaterat mellan enheter och på webben.

## CloudKit-kryptering

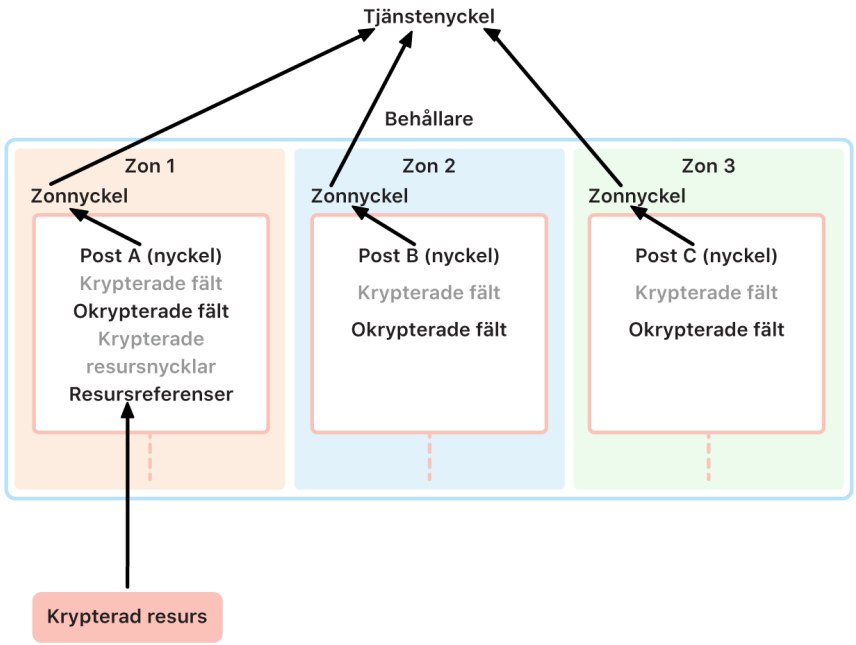
CloudKit är ett ramverk som tillåter att apputvecklare lagrar nyckelvärdedata, strukturerade data och resurser (större data lagrade separat från databasen, exempelvis bilder eller videor) på iCloud. CloudKit hanterar både publika och privata databaser grupperade i behållare. Publika databaser delas globalt, används vanligtvis för allmänna resurser och krypteras inte. Privata databaser lagrar enskilda användares iCloud-data.

iCloud använder en nyckelhierarki som matchar datastrukturen. Varje behållares privata databas skyddas av en nyckelhierarki som bygger på en asymmetrisk nyckel, en så kallad *CloudKit-tjänstenyckel*. De här nycklarna är unika för varje iCloud-användare och genereras på deras betrodda enheter. När data skrivs i CloudKit genereras alla postnycklar på användarens betrodda enhet och paketeras med lämplig nyckelhierarki innan data överförs.




Många Apple-tjänster som listas i Apple Support-artikeln Översikt över iCloud-datasäkerhet använder heltäckande kryptering med en CloudKit-tjänstenyckel som skyddas på samma sätt som synkronisering av iCloud-nyckelring. För de här CloudKit-behållarna är tjänstenycklarna endast tillgängliga på användarens betrodda enheter och varken Apple eller tredje parter har tillgång till dem. Nycklarna synkroniseras mellan en användares enheter även om användaren väljer att inte använda iCloud-nyckelring till att synkronisera sina lösenord, nycklar och andra användardata. Om en enhet förloras kan användare återställa sina data i iCloud-nyckelring genom att använda säker återställning av iCloud-nyckelring, kontakter för kontoåterställning eller en kontoåterställningsnyckel.

## Hantering av krypteringsnycklar




Säkerheten hos krypterade data i CloudKit bygger på säkerheten hos motsvarande krypteringsnycklar. CloudKit-tjänstenycklar delas upp i två kategorier: heltäckande krypterade och available-after-authentication (tillgängliga efter autentisering).

- **Tjänstenycklar som är heltäckande krypterade:** För heltäckande krypterade iCloud-tjänster blir de relevanta privata CloudKit-tjänstenycklarna aldrig tillgängliga för Apples servrar. Tjänstenyckelpar, inklusive de privata nycklarna, skapas lokalt på en användares betrodda enhet och överförs till användarens övriga enheter med iCloud-nyckelringssäkerhet. Flödena för återställning och synkronisering av iCloud-nyckelring förmedlas via Apples servrar, men servrarna är kryptografiskt förhindrade från att komma åt användarens nyckelringsdata. I värsta fall förloras alla heltäckande krypterade data i CloudKit om användaren förlorar tillgången till iCloud-nyckelring och alla dess mekanismer för återställning. Apple kan inte hjälpa till med återställning av dessa data.
- **Tjänstenycklar som är available-after-authentication:** För andra tjänster, som Bilder och iCloud Drive, lagras tjänstenycklarna i iCloud Hardware Security Modules i Apples datacenter och är tillgängliga för vissa Apple-tjänster. När en användare loggar in på iCloud på en ny enhet och autentiserar sitt Apple-ID kan Apples servrar komma åt dessa nycklar utan vidare interaktion eller inmatning från användaren. När användaren till exempel loggar in på iCloud.com kan den omedelbart se sina bilder på webben. De här tjänstenycklarna är nycklar som är *available-after-authentication*.

# Avancerat dataskydd för iCloud

Avancerat dataskydd för iCloud är en valfri inställning som erbjuder Apples högsta säkerhetsnivå för molndata. När en användare slår på Avancerat dataskydd har endast användarens betrodda enheter tillgång till krypteringsnycklarna för huvuddelen av dennas iCloud-data, vilka därmed skyddas med *heltäckande kryptering*. För användare som slår på Avancerat dataskydd ökar det totala antalet datakategorier som skyddas med heltäckande kryptering från 14 till 23 och omfattar iCloud-säkerhetskopiering, Bilder, Anteckningar med mera.

*Obs!* Den här funktionen är kanske inte tillgänglig i alla länder eller regioner.

Avancerat dataskydd är ett enkelt koncept: Alla CloudKit-tjänstenycklar som genererades på en enhet och senare överfördes till iCloud Hardware Security Modules (HSM:er) för *available-after-authentication* i Apples datacenter raderas från dessa HSM:er och förvaras istället helt och hållet inuti kontots skyddsdomän för iCloud-nyckelring. De hanteras som befintliga *heltäckande krypterade* tjänstenycklar, vilket innebär att Apple inte längre kan läsa eller komma åt de nycklarna.

Avancerat dataskydd skyddar även automatiskt CloudKit-fält som tredjepartsutvecklare väljer att markera som krypterade samt alla CloudKit-resurser.

## Aktivera Avancerat dataskydd

När användaren slår på Avancerat dataskydd utför dennas betrodda enhet två åtgärder: För det första kommunicerar den användarens avsikt att slå på Avancerat dataskydd till dennas övriga enheter som deltar i heltäckande kryptering. Den gör det genom att skriva ett nytt värde, som signeras av lokala enhetsnycklar, i enhetens metadata för iCloud-nyckelring. Apples servrar kan inte ta bort eller ändra denna attestering medan den synkroniseras med användarens övriga enheter.

För det andra inleder enheter borttagningen av tjänstenycklar för *available-after-authentication* från Apples datacenter. Eftersom de nycklarna skyddas av iClouds HSM:er är denna radering omedelbar, permanent och oåterkallelig. När nycklarna har raderats kan Apple inte längre komma åt *några som helst* data som skyddas av användarens tjänstenycklar. Vid den här tidpunkten påbörjar enheten en asynkron nyckelrotation som skapar en ny tjänstenyckel för varje tjänst vars nyckel tidigare var tillgänglig för Apples servrar. Om nyckelrotationen misslyckas, på grund av en nätverksstörning eller något annat fel, försöker enheten rotera nycklar tills den lyckas.

När tjänstenyckelrotationen är klar kan nya data som skrivs i tjänsten inte avkrypteras med den gamla tjänstenyckeln. Den skyddas med den nya nyckeln som hanteras helt och hållet av användarens betrodda enheter och var aldrig tillgänglig för Apple.

## Avancerat dataskydd och webbåtkomst till iCloud.com

Första gången en användare slår på Avancerat dataskydd blir webbåtkomsten till data på iCloud.com automatiskt avstängd. Det beror på att iCloud-webbservrarna inte längre har tillgång till nycklarna som krävs för att avkryptera och visa användarens data. Användaren kan välja att slå på webbåtkomst igen och använda sin betrodda enhet till att komma åt sina krypterade iCloud-data på webben.

När användaren har slagit på webbåtkomst måste användaren auktorisera webbinloggningen på en av sina betrodda enheter varje gång han eller hon besöker iCloud.com. Auktoriseringen förbereder enheten för webbåtkomst. Under den kommande timmen godkänner den här enheten förfrågningar från specifika Apple-servrar om att överföra enskilda tjänstenycklar, men endast sådana som motsvarar en lista med tillåtna tjänster som vanligtvis är tillgängliga på iCloud.com. Trots att användaren har auktoriserat en webbinloggning kan en serverförfrågan alltså inte förmå användarens enhet att överföra tjänstenycklar till data som inte är avsedda att visas på iCloud.com (till exempel hälsodata eller lösenord i iCloud-nyckelring). Apples servrar begär endast de tjänstenycklar som behövs för att avkryptera de specifika data som användaren begär tillgång till på webben. Varje gång en tjänstenyckel överförs blir den krypterad med en tillfällig nyckel som är kopplad till webbsessionen som användaren auktoriserade. En notis som visar den iCloud-tjänst vars data tillfälligt görs tillgängliga visas på användarens enhet.

## Bevara användarens val

Inställningarna för Avancerat dataskydd och webbåtkomst till iCloud.com kan endast ändras av användaren. De här värdena sparas i användarens enhetsmetadata för iCloud-nyckelring och kan endast ändras från en av användarens betrodda enheter. Apples servrar kan varken ändra de här inställningarna åt användaren eller återgå till tidigare inställningar.

## Säkerhetsfunktioner vid delning och samarbeten

I de flesta fallen när användare delar innehåll för att samarbeta med varandra – till exempel på delade anteckningar, delade påminnelser, delade mappar i iCloud Drive eller delade bildbibliotek – och alla användare har slagit på Avancerat dataskydd används Apples servrar endast till att upprätta delningen men har inte tillgång till krypteringsnycklarna för delade data. Innehållet förblir heltäckande krypterat och är endast tillgängligt på medverkande betrodda enheter. För varje delningsåtgärd kan en titel och representativ miniatyr lagras av Apple med standarddataskydd för att kunna förhandsvisa dem för mottagare.

Om användaren väljer alternativet ”alla som har en länk” när ett samarbete aktiveras blir innehållet tillgängligt på Apples servrar med standarddataskydd eftersom servrarna måste kunna ge åtkomst till alla som öppnar URL:en.

Samarbeten i iWork och delade album i Bilder saknar stöd för Avancerat dataskydd. När användare samarbetar på ett iWork-dokument, eller öppnar ett iWork-dokument från en delad mapp i iCloud Drive, överförs krypteringsnycklarna för dokumentet på ett säkert sätt till iWork-servrar i Apples datacenter. Det beror på att iWork-samarbeten i realtid kräver hantering på serversidan för att koordinera dokumentändringar mellan deltagare. Bilder som läggs till i delade album lagras med standarddataskydd eftersom funktionen tillåter att album delas offentligt på webben.

## Avaktivera Avancerat dataskydd

Användaren kan när som helst stänga av Avancerat dataskydd. Om den vill göra det:

1. Användarens enhet registrerar först det nya valet i medverkarmetadata för iCloud-nyckelring och den här inställningen synkroniseras säkert till alla deras enheter.

2. Användarens enhet överför tjänstenycklarna för alla *available-after-authentication*-tjänster till iCloud HSM:er i Apples datacenter på ett säkert sätt. Det här omfattar aldrig nycklar för tjänster som är heltäckande krypterade i standarddataskydd, till exempel iCloud-nyckelring och Hälsa.

Enheten överför både de ursprungliga tjänstenycklarna som genererades innan Avancerat dataskydd slogs på och de nya tjänstenycklarna som genererades efter att användaren slog på funktionen. Det här gör alla data i tjänsterna tillgängliga efter autentisering. Kontot återgår till standarddataskydd så att Apple återigen kan hjälpa användaren att återställa huvuddelen av sina data om användaren förlorar tillgången till sitt konto.

## iCloud-data som inte omfattas av Avancerat dataskydd

Eftersom de behöver samverka med globala e-post-, kontakt- och kalendersystem är inte Mail, Kontakter och Kalender på iCloud heltäckande krypterade.

iCloud lagrar en del data utan skydd från användarspecifika CloudKit-tjänstenycklar trots att Avancerat dataskydd är på. CloudKit-postfält måste uttryckligen deklareras som krypterade i behållarens schema om de ska skyddas och läsning och skrivning i krypterade fält kräver att dedikerade API:er används. Datumet och tiden då en fil eller ett objekt ändrades används till att sortera en användares information. Kontrollsummorna för fil- och bilddata hjälper Apple att ta bort dubbletter och optimera användarens iCloud- och enhetslagring utan att Apple får tillgång till de egentliga filerna och bilderna. Information om hur kryptering används för särskilda datakategorier finns i Apple Support-artikeln Översikt över iCloud-datasäkerhet.

Beslut som användningen av kontrollsummor för borttagning av dubbletter – en välkänd teknik som kallas *konvergent konvertering* – ingick i den ursprungliga utformningen av iCloud-tjänsterna vid lanseringen. Dessa metadata är alltid krypterade, men krypteringsnycklarna lagras av Apple med standarddataskydd. Apple arbetar med siktet inställt på att fortsätta stärka säkerhetsskydden för alla användare genom att se till att fler data, inklusive den här typen av metadata, är heltäckande krypterade när Avancerat dataskydd är på.

### Krav för Avancerat dataskydd

Kraven för att slå på Avancerat dataskydd för iCloud omfattar följande:

- Användarens konto måste ha stöd för heltäckande kryptering. Heltäckande kryptering kräver tvåfaktorsautentisering för Apple-ID:t och en lösenkod eller ett lösenord på den betrodda enheten. Om du vill veta mer läser du Apple Support-artikeln Tvåfaktorsautentisering för Apple-ID.
- Enheter där användaren är inloggad med sitt Apple-ID måste uppdateras till iOS 16.2, iPadOS 16.2, macOS 13.1, tvOS 16.2, watchOS 9.2 eller senare och den senaste versionen av iCloud för Windows. Det här kravet förhindrar att en äldre version av iOS, iPadOS, macOS, tvOS eller watchOS felhanterar de nyligen skapade tjänstenycklarna genom att överföra dem till *available-after-authentication*-HSM:er i ett missriktat försök att reparera kontostatusen.
- Användaren måste ställa in minst en alternativ återställningsmetod – en eller flera kontakter för återställning eller en återställningsnyckel – som kan användas till att återställa iCloud-data ifall användaren förlorar tillgången till sitt konto.

Om återställningsmetoden misslyckas, till exempel om återställningskontaktens information inte är aktuell eller om användaren glömmer den, kan Apple inte hjälpa till att återställa användarens heltäckande krypterade iCloud-data.

Avancerat dataskydd för iCloud kan endast slås på för Apple-ID:n. Hanterade Apple-ID:n och barnkonton (varierar beroende på land eller region) stöds inte.

## Säkerhet och iCloud-säkerhetskopiering

iCloud säkerhetskopierar information – inklusive enhetsinställningar, appdata, bilder och videor i kamerarullen och konversationer i appen Meddelanden – dagligen via Wi-Fi. iCloud-säkerhetskopiering sker endast när enheten är låst, ansluten till en strömkälla och har Wi-Fi-anslutning till internet. Med tanke på lagringskrypteringen som används i iOS och iPadOS är iCloud-säkerhetskopiering utformad för att skydda data samtidigt som det tillåter stegvis, oövervakad säkerhetskopiering och återställning. Som förval blir tjänstenyckeln för iCloud-säkerhetskopiering tryggt säkerhetskopierad till iCloud Hardware Security Modules i Apples datacenter som en del av datakategorin available-after-authentication. För användare som slår på Avancerat dataskydd för iCloud skyddas tjänstenycken för iCloud-säkerhetskopiering med heltäckande kryptering och är endast tillgänglig för användare på deras betrodda enheter.

När filer skapas i dataskyddsklasser som inte är tillgängliga när enheten är låst blir deras filnycklar krypterade med hjälp av klassnycklarna från nyckelsamlingen för iCloud-säkerhetskopiering, och filerna säkerhetskopieras till iCloud i det ursprungliga, krypterade läget. Alla filer krypteras under transport och krypteras vid lagring med kontobaserade nycklar, som det beskrivs under CloudKit-kryptering.

Nyckelsamlingen för iCloud-säkerhetskopiering innehåller asymmetriska nycklar (Curve25519) för dataskyddsklasser som inte är åtkomliga när enheten är låst. Uppsättningen för säkerhetskopiering lagras i användarens iCloud-konto och består av en kopia av användarens filer samt nyckelsamlingen för iCloud-säkerhetskopiering. Nyckelsamlingen för iCloud-säkerhetskopiering skyddas av en slumpgenererad nyckel som också lagras tillsammans med säkerhetskopieringsuppsättningen. Användarens iCloud-lösenord används inte för kryptering, så att byta iCloud-lösenord gör inte befintliga säkerhetskopior ogiltiga.

Vid återskapande hämtas de säkerhetskopierade filerna, nyckelsamlingen för iCloud-säkerhetskopiering och nyckeln till nyckelsamlingen från användarens iCloud-konto. Nyckelsamlingen för iCloud-säkerhetskopiering avkrypteras med dess egen nyckel. Sedan används filnycklarna i nyckelsamlingen till att avkryptera filerna i uppsättningen för säkerhetskopiering. Dessa skrivs som nya filer till filsystemet och omkrypteras därmed i enlighet med sin dataskyddsklass.

Följande innehåll säkerhetskopieras med iCloud-säkerhetskopiering:

- Poster för inköpt musik, filmer, TV-program, appar och böcker. En användares iCloud-säkerhetskopia innehåller information om inköpt innehåll som finns på användarens enhet, men inte själva det inköpta innehållet. När användaren återskapar från en iCloud-säkerhetskopia hämtas det inköpta innehållet automatiskt från iTunes Store, App Store, Apple TV-appen och Apple Books. Vissa typer av innehåll hämtas inte automatiskt i alla länder eller regioner, och tidigare inköp kanske inte är tillgängliga om återköp har gjorts eller om objekten inte längre är tillgängliga i respektive affär. Den fullständiga köphistoriken är kopplad till en användares Apple-ID.
- Bilder och videor på en användares enheter. Lägg märke till att om en användare aktiverar iCloud-bilder i iOS 8.1, iPadOS 13.1 eller OS X 10.10.3 eller senare lagras användarens bilder och videor redan i iCloud, så de ingår inte i iCloud-säkerhetskopian.
- Kontakter, kalenderaktiviteter, påminnelser och anteckningar
- Enhetsinställningar
- Appdata
- Hemskärmen och ordningen på appar
- HomeKit-konfiguration
- Data för medicinskt ID
- Lösenord för Röstmemon (vid behov krävs det SIM-kort som användes vid säkerhetskopieringen)
- Meddelanden, Apple Messages for Business-meddelanden samt SMS och MMS (vid behov krävs det SIM-kort som användes vid säkerhetskopieringen)

iCloud-säkerhetskopiering används även till att säkerhetskopiera den lokala enhetsnyckelringen som krypteras med en nyckel som härleds från enhetens kryptografiska UID-rotnyckel i Secure Enclave. Nyckeln är unik för enheten och inte känd av Apple. Det gör att databasen bara kan återskapas till samma enhet som den kom ifrån, och det betyder att ingen annan (inte heller Apple) kan läsa den. Mer information finns i Secure Enclave.

### Meddelanden på iCloud

Meddelanden på iCloud håller en användares hela meddelandehistorik uppdaterad och tillgänglig på alla enheter.

Med standarddataskydd krypteras Meddelanden till iCloud heltäckande när iCloud-säkerhetskopiering är avstängd. När iCloud-säkerhetskopiering är påslagen innehåller säkerhetskopian en kopia av krypteringsnyckeln för Meddelanden på iCloud så att Apple kan hjälpa användaren att återställa sina meddelanden även om den har förlorat tillgången till iCloud-nyckelring och sina betrodda enheter. Om användaren stänger av iCloud-säkerhetskopiering genereras en ny nyckel på enheten för att skydda framtida konversationer i Meddelanden på iCloud. Den nya nyckeln lagras endast i iCloud-nyckelring, är endast tillgänglig för användaren på betrodda enheter och nya data som skrivs i behållaren kan inte avkrypteras med den gamla behållarnyckeln.

Med Avancerat dataskydd är Meddelanden på iCloud alltid heltäckande krypterad. När iCloud-säkerhetskopiering är påslagen blir allt inuti den heltäckande krypterat, inklusive krypteringsnyckeln för Meddelanden på iCloud. Tjänstenyckeln för iCloud-säkerhetskopiering, liksom behållarnyckeln för Meddelanden på iCloud, byts ut när användaren slår på Avancerat dataskydd. Mer information finns i Apple Support-artikeln Översikt över iCloud-datasäkerhet.

## Säkerhet för Privat reläservice på iCloud

Privat reläservice på iCloud bidrar huvudsakligen till att skydda användare när de surfar på webben i Safari, men funktionen innehåller även alla förfrågningar om DNS-namnlösning. Den ser till att ingen enskild part, inte ens Apple, kan koppla samman användarens IP-adress med dess surfaktivitet. Den gör det genom att använda olika proxyer: en ingående proxy som hanteras av Apple och en utgående proxy som hanteras av en innehållsleverantör. För att kunna använda Privat reläservice på iCloud måste användaren ha iOS 15, iPadOS 15 eller macOS 12.0.1 eller senare och vara inloggad på sitt iCloud+-konto med sitt Apple-ID. Därefter går det att aktivera Privat reläservice på iCloud i Inställningar > iCloud eller Systeminställningar > iCloud.

Mer information finns i iCloud Private Relay Overview.

## Säkerhet för kontakt för kontoåterställning

Användare kan lägga till upp till fem personer de litar på som kontakter för kontoåterställning som kan hjälpa till att återställa användarens iCloud-konto och dess data, inklusive alla heltäckande krypterade data, oavsett om de har slagit på Avancerat dataskydd. Varken Apple eller återställningskontakten har den information som krävs för att enskilt återställa användarens heltäckande krypterade iCloud-data.

Kontakter för kontoåterställning är utformade med användarens integritet i åtanke. Apple känner inte till en användares valda kontakter för återställning. Apples servrar får ingen information om en kontakt för återställning förrän sent under ett återställningsförsök när användaren redan har bett kontakten om hjälp och kontakten börjar hjälpa till med återställningen. Informationen sparas inte när återställningen är klar.

## Säkerhetsprocess för återställningskontakter

När en användare ställer in en kontakt för kontoåterställning genereras en nyckel som associeras med den kontakten. Den här nyckeln skyddar åtkomsten till användarens iCloud-data, inklusive heltäckande krypterade CloudKit-data. Sedan genereras en slumpmässig 256-bitars AES-nyckel som används till att kryptera återställningskontaktens nyckel för att skapa ett återställningskontaktpaket. Det krypterade paketet skickas till återställningskontakten för säker förvaring och den slumpmässiga AES-nyckeln lagras hos Apple. Varken AES-nyckeln eller paketet tillhandahåller någon information om den underliggande nyckeln på egen hand. Vid tidpunkten för återställning, efter att användarens enhet har fått både återställningskontaktpaketet från sin återställningskontakt och AES-nyckeln från Apple, kan de båda delarna kombineras för att återställa den ursprungliga nyckeln och åtkomsten till användarens iCloud-data.

När en återställningskontakt skapas kommunicerar användarens enhet med Apples servrar så att den kan överföra den del av nyckelinformationen som Apple ska lagra (den AES-nyckel som nämns ovan). Sedan upprättar den en heltäckande krypterad CloudKit-behållare för återställningskontakten och delar den nyckeldel som återställningskontakten behöver (återställningskontaktpaketet som krypteras med AES-nyckeln). En auktoriseringshemlighet som skapas av Apple delas även med återställningskontakten. Den används till att återställa kontot och hjälper till att skapa ett nytt lösenord för kontot. Kommunikationen som sker för att bjuda in och godkänna återställningskontakter äger rum via en gemensamt autentiserad IDS-kanal. Återställningskontakten lagrar automatiskt den mottagna informationen i sin iCloud-nyckelring. Apple kan inte komma åt innehållet i CloudKit-behållaren eller den iCloud-nyckelring som lagrar informationen. När delningen utförs kan Apples servrar endast se ett anonymt ID för återställningskontakten.

När en användare sedan behöver återställa sitt konto och sina iCloud-data kan den begära hjälp från sin återställningskontakt. Vid den tidpunkten genereras en återställningskod av återställningskontaktens enhet och återställningskontakten tillhandahåller sedan koden till användaren manuellt (exempelvis öga mot öga eller via telefon). Därefter anger användaren återställningskoden på sin enhet för att upprätta en säker anslutning mellan enheterna via protokollet SPAKE2+ vars innehåll inte är tillgängligt för Apple. Den här interaktionen dirigeras av Apple-servrar, men Apple kan inte inleda återställningsprocessen.

När den säkra anslutningen har upprättats, och alla krävda säkerhetskontroller är genomförda, returnerar återställningskontaktens enhet sin del av nyckelinformationen och den tidigare upprättade auktoriseringshemligheten till användaren som begär återställning. Användaren presenterar den här auktoriseringshemligheten för en Apple-server som beviljar åtkomst till nyckelinformationen som Apple lagrar. När auktoriseringshemligheten tillhandahålls auktoriseras även att användaren skapar ett nytt kontolösenord för att återskapa kontoåtkomsten.

Slutligen kombinerar användarens enhet nyckelinformationen från Apple och kontakten för kontoåterställning och använder den till att avkryptera och återställa sina iCloud-data.

Det finns skyddsmekanismer som förhindrar att en återställningskontakt inleder en återställning utan användarens medgivande, vilka omfattar en aktivitetskontroll av användarens konto. Om kontot används kräver en återställning även kännedom om en nylig enhetslösenkod eller iCloud-säkerhetskoden.

## Säkerhet för kontakt för digitalt arv

Om en användare vill att dess data ska vara tillgängliga för utsedda förmånstagare efter användarens bortgång kan den skapa kontakter för digitalt arv för sitt konto. En kontakt för digitalt arv upprättas på liknande sätt som en återställningskontakt med undantag för att nyckelinformationen som används av en förmånstagare inte omfattar informationen som krävs för att avkryptera den avlidnes iCloud-nyckelring. Nyckelstrukturen som används är likadan som den för kontoåterställningskontakter förutom att Apple i det här fallet lagrar det krypterade paketet och förmånstagaren behåller AES-nyckeln. Det gör att delen som förmånstagaren får kan vara kortare – och därmed enklare att skriva ut vid behov – samtidigt som den ändå tillhandahåller samma egenskap som innebär att ingen del tillhandahåller någon information om den underliggande nyckeln på egen hand.

Nyckelinformationen som en förmånstagare får refereras till som en åtkomstnyckel i användarriktad dokumentation. Åtkomstnyckeln sparas automatiskt på enheter som stöds, men den kan också skrivas ut och arkiveras för användning vid behov. Mer information finns i Apple Support-artikeln Så här lägger du till en kontakt för digitalt arv för ditt Apple-ID.

Efter användarens bortgång loggar kontakten för digitalt arv in på Apples webbplats för att begära åtkomst. Detta kräver ett dödscertifikat och auktoriseras delvis med auktoriseringshemligheten som nämndes i det föregående avsnittet. När säkerhetskontrollerna är slutförda utfärdar Apple ett användarnamn och lösenord för det nya kontot och släpper den nyckelinformation som behövs till kontakten för digitalt arv.

För att underlätta inmatning av åtkomstkoden vid behov visas den som en alfanumerisk kod tillsammans med en associerad QR-kod. När den har matats in återställs åtkomsten till den bortgångnes iCloud-data. Det här kan utföras på en enhet, men åtkomsten kan också upprättas online. Mer information finns i Apple Support-artikeln Begär åtkomst till ett Apple-konto som kontakt för digitalt arv.

# Hantering av lösenkoder och lösenord

## Lösenordssäkerhet i översikt

iOS, iPadOS och macOS gör det enkelt för användare att autentisera sig i tredjepartsappar och på webbplatser som använder lösenord. Det bästa sättet att hantera lösenord är att inte behöva använda några. Med Logga in med Apple kan användarna logga in i appar från tredje part och på webbplatser utan att behöva skapa och hantera flera konton och lösenord eftersom inloggningen skyddas med tvåfaktorsautentiseringen för Apple-ID. För webbplatser som inte har stöd för Logga in med Apple kan funktionen Automatiska starka lösenord användas till att automatiskt skapa, synkronisera och ange unika starka lösenord för webbplatser och appar på enheter. I iOS och iPadOS sparas lösenord i en särskild Autofyll lösenord-nyckelring som styrs och hanteras av användaren under Inställningar > Lösenord.

I macOS kan användaren hantera lösenord i inställningspanelen Lösenord i Safari. Det här synkroniseringssystemet kan också användas till att synkronisera lösenord som skapas manuellt av användaren.

## Säkerhet och Logga in med Apple

Logga in med Apple är ett alternativ med hög integritet jämfört med andra system för enkel inloggning. Det innebär praktisk, effektiv och enkel inloggning med ett enda tryck samtidigt som användaren lättare kan överblicka och kontrollera sin personliga information.

Med Logga in med Apple kan användare ställa in ett konto och logga in på appar och webbplatser med de Apple-ID:n de redan har, och det ger dem större kontroll över den personliga informationen. Appar kan endast efterfråga användarens namn och e-postadress när ett konto ställs in, och användarna har alltid ett val: De kan dela sin personliga e-postadress med en app eller gömma sin personliga e-postadress och använda Apples nya tjänst för vidarebefordring av mejl istället. Den här tjänsten för vidarebefordring av mejl delar en unik, anonymiserad e-postadress som vidarebefordras till användarens personliga e-postadress så att användaren fortfarande kan ta emot mejl från utvecklaren, men ändå upprätthålla viss integritet och kontroll över sin personliga information.

Logga in med Apple är byggt för säkerhet. Varje användare av Logga in med Apple måste ha aktiverat tvåfaktorsautentisering för sitt Apple-ID. Tvåfaktorsautentisering skyddar inte bara användarnas Apple-ID:n, utan också de konton de skapar med sina appar. Apple har dessutom utvecklat och integrerat en integritetsvänlig signal som varnar för bedrägeri i Logga in med Apple. Signalen gör att utvecklare kan känna sig trygga i vetskapen att de nya användarna är riktiga personer och inte bottar eller skriptkonton.

## Automatiska starka lösenord

När iCloud-nyckelring är aktiverad skapar iOS, iPadOS och macOS starka, slumpmässiga och unika lösenord när användare registrerar sig eller ändrar lösenord på en webbplats i Safari. I iOS och iPadOS går det också att generera starka lösenord automatiskt i appar. Användare måste aktivt välja bort starka lösenord. Genererade lösenord sparas i nyckelringen och hålls uppdaterade mellan enheter med iCloud-nyckelring när den är aktiverad.

Lösenord som genereras av iOS och iPadOS är som förval 20 tecken långa. De innehåller en siffra, ett versalt tecken, två bindestreck och 16 gemena tecken. De genererade lösenorden är starka och innehåller 71-bitars entropi.

Lösenord genereras baserat på heuristik som bedömer om en password-field-upplevelse är avsedd för att skapa lösenord eller inte. Om heuristiken misslyckas med att upptäcka att ett kontextspecifikt lösenord används när ett nytt lösenord skapas kan apputvecklare ange `UITextContentType.newPassword` i textfältet och webbutvecklare ange `autocomplete="new-password"` i sina <input>-element.

För att säkerställa att genererade lösenord är kompatibla med de relevanta tjänsterna kan appar och webbplatser tillhandahålla regler. Utvecklare kan tillhandahålla de här reglerna med `UITextInputPasswordRules` eller attributet `passwordrules` i sina inmatningselement. Enheter skapar sedan det starkaste lösenord de kan som uppfyller dessa regler.

## Säkerhet och Autofyll lösenord

Autofyll lösenord fyller automatiskt i inloggningsuppgifter som lagras i nyckelringen. Lösenordshanteraren för iCloud-nyckelring och Autofyll lösenord gör följande:

- Fyller i inloggningsuppgifter i appar och på webbplatser
- Genererar starka lösenord
- Sparar lösenord i både appar och webbplatser i Safari
- Delar lösenord säkert till en användares kontakter
- Tillhandahåller lösenord till en Apple TV i närheten som efterfrågar inloggningsuppgifter

Generering och sparande av lösenord i appar, liksom att tillhandahålla lösenord för Apple TV, är endast tillgängligt i iOS och iPadOS.

### Autofyll lösenord i appar

Med iOS och iPadOS kan användare mata in sparade användarnamn och lösenord i relaterade fält i appar på ungefär samma sätt som med Autofyll lösenord i Safari. I iOS och iPadOS trycker användaren på ett alternativ i programvarutangentbordets QuickType-fält. I macOS visas rullgardinsmenyn Lösenord under relaterade fält i appar som är byggda med Mac Catalyst.

När en app är starkt kopplad till en webbplats som använder samma app-website association-mekanism, och som drivs av samma apple-app-site-association-fil, föreslår QuickType-fältet i iOS och iPadOS och rullgardinsmenyn i macOS direkt inloggningsuppgifter för appen om sådana har sparats i nyckelringen för Autofyll lösenord. Det innebär att användare kan välja att lämna ut inloggningsuppgifter som har sparats i Safari till appar som har samma säkerhetsegenskaper utan att de apparna måste använda en API.

Autofyll lösenord exponerar inte några inloggningsuppgifter för appen förrän en användare samtycker till att ge uppgifterna till appen. Behörighetslistorna skapas från eller presenteras utanför appens process.

När en app och webbplats har en betrodd relation, och en användare skickar inloggningsuppgifter inuti en app, kan iOS och iPadOS uppmana användaren att spara inloggningsuppgifterna i nyckelringen Autofyll lösenord för senare användning.

## Apptillgång till sparade lösenord

iOS-, iPadOS- och macOS-appar kan begära hjälp från nyckelringen Autofyll lösenord för att logga in en användare med `ASAuthorizationPasswordProvider` och `SecAddSharedWebCredential`. Lösenordsleverantören och dess begäran kan användas tillsammans med Logga in med Apple så att samma API anropas för att hjälpa användare att logga in i en app, oavsett om användarens konto är lösenordsbaserat eller har skapats med Logga in med Apple.

Apparna kan bara komma åt sparade lösenord om både apputvecklaren och webbplatsadministratören har gett sitt godkännande och användaren samtycker. Apputvecklare meddelar sin avsikt att använda sparade lösenord i Safari genom att inkludera behörigheterna till det i appen. I behörigheterna anges de fullt berättigade domännamnen på associerade webbplatser. Webbplatserna måste även placera en fil på servern som innehåller en lista på de unika appidentifierarna hos appar som har godkänts av Apple.

När en app med behörigheten com.apple.developer.associated-domains installeras skickar iOS och iPadOS en TLS-begäran till alla webbplatser på listan och begär en av följande filer:

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

Om filen innehåller appidentifieraren för appen som installeras kommer iOS och iPadOS att markera webbplatsen och appen som betrodda. Det är bara om relationen är betrodd som anrop till dessa två API:er leder till en fråga till användaren som sedan måste samtycka innan några lösenord lämnas ut till appen, uppdateras eller raderas.

## Säkerhetsrekommendationer för lösenord

Lösenordslistan för Autofyll lösenord i iOS, iPadOS och macOS indikerar vilka av en användares sparade lösenord som *återanvänds* på flera webbplatser, vilka lösenord som anses vara *svaga* och lösenord som har äventyrats genom en *dataläcka*.

### Översikt

Om samma lösenord används för fler än en tjänst kan dessa tjänster bli sårbara för ett inloggningsangrepp. Om en tjänst hackas och lösenord läcker ut kan angripare testa samma inloggningsuppgifter på andra tjänster för att manipulera ytterligare konton.

- Ett lösenord markeras som *återanvänt* om funktionen identifierar att samma lösenord använts för fler än en domän.

- Lösenord markeras som *svaga* om de enkelt kan gissas av en angripare. iOS, iPadOS och macOS identifierar vanliga mönster som används till att skapa lösenord som är lätta att komma ihåg, t.ex. genom att använda ord från en ordbok, vanliga teckenutbyten (t.ex. ”p4ssw0rd” istället för ”password”), mönster som går att hitta på tangentbordet (t.ex. ”q12we34r” från ett QWERTY-tangentbord) eller upprepade sekvenser (t.ex. ”123123”). Dessa mönster används ofta för att skapa lösenord som uppfyller olika tjänsters minimikrav för lösenord, men används också ofta av angripare som försöker knäcka lösenordet genom att prova stora mängder lösenord.

  Eftersom många tjänster specifikt begär en fyr- eller sexsiffrig PIN-kod utvärderas dessa korta lösenkoder med andra regler. PIN-koder anses svaga om de hör till någon av de vanligaste PIN-koderna, om de är en stigande eller fallande sekvens som ”1234” eller ”8765” eller om de följer ett upprepat mönster, t.ex. ”123123” eller ”123321”.

- Lösenord markeras som *läckta* om funktionen för lösenordsövervakning signalerar att de har ingått i en dataläcka Mer information finns i Lösenordsövervakning.

Svaga, återanvända och läckta lösenord indikeras antingen i listan över lösenord (macOS) eller finns med i det dedikerade gränssnittet Säkerhetsrekommendationer (iOS och iPadOS). Om användaren loggar in på en webbplats i Safari med ett tidigare sparat lösenord som är väldigt svagt, eller som har äventyrats genom en dataläcka, visas en varning som uppmanar användaren att uppgradera till ett automatiskt starkt lösenord.

## Uppgradera säkerheten för kontoautentisering i iOS och iPadOS

Appar som implementerar AAME (i ramverket Authentication Services) kan tillhandahålla enkel uppgradering med bara ett knapptryck för lösenordsbaserade konton – de kan nämligen växla över till att använda Logga in med Apple eller ett automatisk starkt lösenord. Den här tilläggspunkten är tillgänglig för iOS och iPadOS.

Om en app har implementerat tilläggspunkten och är installerad på enheten kan användaren se tilläggsuppgraderingsalternativ när säkerhetsrekommendationer visas för behörigheter kopplade till appen i lösenordshanteraren för iCloud-nyckelring i Inställningar. Uppgraderingsalternativen erbjuds också när en användare loggar in i appen med osäkra inloggningsuppgifter. Appar kan be systemet att inte visa uppgraderingsalternativ efter inloggning. Genom att använda nya AuthenticationServices API kan appar också anropa sina tillägg och utföra uppgraderingar själva, i första hand från en skärm med kontoinställningar eller kontohantering i appen.

Appar kan välja att stöda uppgradering till starka lösenord, uppgradering till Logga in med Apple eller både och. Vid en uppgradering till starka lösenord genererar systemet ett automatiskt starkt lösenord åt användaren. Om det behövs kan appen tillhandahålla anpassade lösenordsregler som sedan följs när det nya lösenordet genereras. När en användare byter från användning av ett lösenord till användning av Logga in med Apple för ett konto tillhandahåller systemet en ny behörighet för Logga in med Apple till tillägget för associering till kontot. Den e-postadress som användaren har kopplat till Apple-ID:t ingår inte i behörigheten. Efter en lyckad uppgradering till Logga in med Apple raderar systemet den tidigare använda lösenordsbehörigheten från användarens nyckelringen om den är sparad där.

AAME har möjlighet att utföra ytterligare användarautentisering innan en uppgradering genomförs. För uppgraderingar som initierats via lösenordshanteraren, eller efter inloggning till en app, tillhandahåller tillägget användarnamn och lösenord för kontot inför uppgraderingen. För uppgraderingar inuti appar tillhandahålls endast användarnamnet. Om tillägget kräver ytterligare användarautentisering kan det begära att få visa ett anpassat användargränssnitt innan uppgraderingen kan fortsätta. Möjligheten att visa ett det här användargränssnittet finns för att tillägget ska kunna be användaren att ange en andra autentiseringsfaktor för att auktorisera uppgraderingen.

## Lösenordsövervakning

Lösenordsövervakning är en funktion som kontrollerar lösenord som lagras i användarens nyckelring Autofyll lösenord mot en kontinuerligt uppdaterad och övervakad lista över lösenord som har exponerats i läckor från olika webborganisationer. Om funktionen är aktiverad kontrollerar det övervakande protokollet kontinuerligt användarens lösenord i nyckelringen Autofyll lösenord mot listan.

### Hur övervakning fungerar

Användarens enhet utför kontinuerligt round-robin-kontroller för användarens lösenord och skickar förfrågningar för ett intervall som är oberoende av användarens lösenord eller dess användningsmönster för lösenordshanteraren. Det säkerställer att verifieringsstatusen hålls uppdaterad med den aktuella listan över läckta lösenord. För att förhindra läckage av information, relaterad till hur många unika lösenord en användare har, sker förfrågningar gruppvis och utförs parallellt. Ett fast antal lösenord verifieras parallellt vid varje kontroll, och om användaren har färre lösenord än detta antal genereras slumpmässiga lösenord och läggs till i förfrågningarna för att fylla ut antalet.

### Hur lösenord matchas

Lösenord matchas i en tvådelad process. De vanligaste läckta lösenorden finns i en lokal lista på användarens enhet. Om användarens lösenord finns i den här listan meddelas användaren omedelbart utan någon extern interaktion. Detta ser till att ingen information läcker ut om de lösenord en användare har som utgör störst risk på grund en lösenordsläcka.

Om lösenordet inte finns i listan över de vanligaste läckta lösenorden matchas det mot mer sällan läckta lösenord.

### Jämföra användares lösenord mot en övervakad lista

För att verifiera om ett lösenord som inte finns i den lokala listan matchar mer sällan läckta lösenord krävs viss interaktion med Apple-servrar. För att säkerställa att legitima användares lösenord inte skickas till Apple används en typ av kryptografisk *privat uppsättningsskärningspunkt* som jämför användarnas lösenord mot en stor uppsättning läckta lösenord. Detta är tänkt att säkerställa att mycket lite information delas med Apple för lösenord med lägre risk. För en användares lösenord är den här informationen begränsad till ett 15-bitarsprefix av en kryptografisk hash. Att de vanligaste läckta lösenorden tagits bort från den här interaktiva processen, genom användning av den lokala listan över de vanligaste läckta lösenorden, minskar delta för den relativa frekvensen för lösenord i webbtjänstens behållare. Detta gör det opraktiskt att försöka komma åt användarlösenord via den här processen.

Det underliggande protokollet delar upp listan över övervakade lösenord, som innehåller ungefär 1,5 miljarder lösenord när denna text skrivs, i $2^{15}$ olika behållare. Vilken behållare ett lösenord hör till baseras på de första 15 bitarna av lösenordets SHA256-hashvärde. Dessutom är varje läckt lösenord, pw, kopplat till en elliptisk kurvpunkt på NIST P256-kurvan: $P_{pw} = \alpha \cdot H_{SWU}(pw)$, där $\alpha$ är en hemlig slumpmässig nyckel som endast Apple känner till, och $H_{SWU}$ är en slumpmässig oracle-funktion som mappar lösenord till kurvpunkter baserat på Shallue-van de Woestijne-Ulas-metoden. Den här transformationen är skapad för att beräkningsmässigt gömma värdena för lösenord och förhindra avslöjandet av nyligen läckta lösenord genom lösenordsövervakning.

För att beräkna den privata uppsättningsskärningspunkten avgör användarens enhet vilken behållare användarens lösenord hör till genom att använda λ, 15-bitarsprefixet för SHA256(upw), där upw är ett av användarens lösenord. Enheten genererar en egen slumpmässig konstant, β, och skickar punkten $P_c = \beta \cdot H_{SWU}(upw)$ till servern tillsammans med en begäran om behållaren som motsvarar λ. Här gömmer β information om användarens lösenord och begränsar till λ den information som exponeras från lösenordet till Apple. Slutligen tar servern den punkt som skickats av användarens enhet, beräknar $\alpha P_c = \alpha\beta \cdot H_{SWU}(upw)$ och returnerar informationen tillsammans med lämplig behållare för punkterna $B\lambda = \{ P_{pw} \mid \text{SHA256(pw) börjar med prefixet } \lambda\}$ till enheten.

Den returnerade informationen tillåter enheten att beräkna $B'_\lambda = \{\beta \cdot P_{pw} \mid P_{pw} \in B_\lambda\}$ och säkerställer att användarens lösenord har läckts om $\alpha P_c \in B'_\lambda$.

## Skicka lösenord till andra användare eller Apple-enheter

Apple skickar lösenord på ett säkert sätt till andra användare eller Apple-enheter med AirDrop och på Apple TV.

### Spara inloggningsuppgifter på en annan enhet med AirDrop

När iCloud är aktiverat kan användare skicka sparade inloggningsuppgifter till en annan enhet via AirDrop. Inloggningsuppgifterna omfattar användarnamn och lösenord samt för vilka webbplatser de är sparade. När inloggningsuppgifter skickas via AirDrop används alltid läget Bara kontakter, oberoende av användarens inställningar. När användaren har gett sitt medgivande på mottagarenheten sparas inloggningsuppgifterna i användarens nyckelring för Autofyll lösenord.

### Fylla i inloggningsuppgifter på Apple TV

Autofyll lösenord är tillgängligt för att fylla i inloggningsuppgifter i appar på Apple TV. När användaren fokuserar på ett textfält för användarnamn eller lösenord i tvOS börjar Apple TV att annonsera en förfrågan om Autofyll lösenord via Bluetooth Low Energy (BLE).

iPhone- och iPad-enheter i närheten visas ett meddelande som uppmanar användaren att dela en inloggningsuppgift med Apple TV. Så här upprättas krypteringsmetoden:

- Om enheten och Apple TV använder samma iCloud-konto sker kryptering av kommunikationen mellan de två enheterna automatiskt.
- Om enheten är inloggad på ett annat iCloud-konto än det som används av Apple TV blir användaren ombedd att upprätta en krypterad anslutning genom att använda en PIN-kod. För att få den här uppmaningen måste iPhone vara olåst och nära den Siri Remote som är parkopplad med Apple TV.

När den krypterade anslutningen har skapats med BLE-länkkryptering skickas inloggningsuppgiften till Apple TV och anges automatiskt i relevanta textfält i appen.

## Tillägg för inloggningsuppgifter

I iOS, iPadOS och macOS kan användare utse en deltagande app från tredje part som leverantör av inloggningsuppgifter för Autofyll lösenord på inställningspanelen Lösenord (iOS och iPadOS) eller på inställningspanelen Tillägg i Systeminställningar (macOS). Den här mekanismen bygger på apptillägg. Tillägget för inloggningsuppgifter *måste innehålla* en vy för att välja inloggningsuppgifter. Tillägget *kan som tillval* tillhandahålla metadata om sparade inloggningsuppgifter så att de kan erbjudas direkt i QuickType-fältet (iOS och iPadOS) eller som ett förslag på autokomplettering (macOS). Dessa metadata innehåller webbplatsen för inloggningsuppgiften och det associerade användarnamnet, men inte lösenordet. iOS, iPadOS och macOS kommunicerar med tillägget för att få lösenordet när användaren väljer att fylla i en inloggningsuppgift i en app eller på en webbplats i Safari. Metadata för inloggningsuppgifter sparas i behållaren för inloggningsuppgiftstillhandahållarens app och tas automatiskt bort när en app avinstalleras.

# iCloud-nyckelring

## Säkerhet för iCloud-nyckelring i översikt

Med iCloud-nyckelring kan användarna synkronisera sina lösenord och nycklar säkert mellan iPhone- och iPad-enheter och Mac-datorer utan att Apple kan se dem. Utöver stark integritet och säkerhet var andra mål för utformningen och arkitekturen hos iCloud-nyckelring användarvänlighet och möjlighet att återställa nyckelringsinnehåll även om ingen av en användares enheter är tillgänglig. iCloud-nyckelring består av två tjänster: synkronisering och återställning av nyckelringen.

iCloud-nyckelring och återställning av nyckelringen har utformats så att användarens lösenord och nycklar fortfarande skyddas under följande omständigheter:

- Om det uppstår säkerhetsproblem med användarens iCloud-konto.
- Om iCloud attackeras av obehöriga utifrån eller av en anställd.
- Om tredje part ansluter till användarkonton.

### Integrering av lösenordshantering med iCloud-nyckelring

iOS, iPadOS och macOS kan automatiskt generera kryptografiskt starka slumpmässiga strängar att använda som kontolösenord i Safari. iOS och iPadOS kan också generera starka lösenord för appar. Genererade lösenord lagras i nyckelringen och synkroniseras till andra enheter. Nyckelringsobjekt överförs från enhet till enhet och färdas då via Apples servrar, men de krypteras heltäckande så att varken Apple eller någon annan enhet kan läsa deras innehåll.

## Säker synkronisering av nyckelringen

När en användare slår på iCloud-nyckelring för första gången för ett konto med tvåfaktorsautentisering upprättar och skapar enheten en synkroniseringsidentitet åt sig själv. Synkroniseringsidentiteten består av asymmetriska elliptiska nycklar (med P-384) som lagras i enhetens nyckelring. Varje enhet upprätthåller sin egen lista med synkroniseringsidentiteter på användarens andra enheter och signerar denna lista med en av dess identitetsnycklar. Dessa listor lagras i CloudKit så att användarens enheter kan uppnå samstämmighet om hur nyckelringsdata ska synkroniseras säkert mellan sig själva.

För kompatibilitet med äldre iCloud-enheter skapas en liknande tillförlitlighetscirkel för synkronisering och en annan synkroniseringsidentitet bildas. Den publika nyckeln till synkroniseringsidentiteten läggs till i cirkeln och cirkeln signeras två gånger: först av den privata nyckeln i synkroniseringsidentiteten och sedan igen med en asymmetrisk elliptisk nyckel (med P-256) som härleds ur lösenordet till användarens iCloud-konto. I cirkeln sparas också parametrarna (slumpmässigt salt och iterationer) som används till att skapa nyckeln som baseras på användarens iCloud-lösenord.

### iCloud-lagring av synkroniseringscirkeln

För konton med tvåfaktorsautentisering lagras varje enhets lista med betrodda enheter i CloudKit. Listorna kan inte läsas utan att känna till användarens iCloud-lösenord och kan inte ändras utan tillgång till de privata nycklarna för ägarenheten.

På liknande sätt lagras den signerade synkroniseringscirkeln i användarens lagringsutrymme för nyckelvärden på iCloud. Den kan inte läsas utan att känna till användarens iCloud-lösenord och kan inte ändras på ett giltigt sätt utan tillgång till den privata nyckeln för medlemmens synkroniseringsidentitet.

### Så här läggs användares andra enheter till i synkroniseringscirkeln

När nya enheter loggar in på iCloud ansluter de till iCloud-nyckelrings synkroniseringscirkel på ett av två sätt: antingen genom att parkoppla med och sponsras av en befintlig enhet med iCloud-nyckelring eller genom att använda återställning för iCloud-nyckelring.

Under parkopplingsflödena skapar den sökande enheten nya synkroniseringsidentiteter för både synkroniseringscirkeln och synkroniseringslistorna (för konton med tvåfaktorsautentisering) och presenterar dem för sponsorn. Sponsorn lägger till den nya medlemmens publika nyckel i synkroniseringscirkeln och signerar den igen med både dess synkroniseringsidentitet och med nyckeln som härleds ur användarens iCloud-lösenord. Den nya synkroniseringscirkeln placeras på iCloud där den sedan signeras på ett liknande sätt av den nya cirkelmedlemmen. I konton med tvåfaktorsautentisering tillhandahåller sponsorenheten även ett *kvitto* som är signerat med dess identitetsnycklar till den anslutande enheten, vilka visar att den ansökande enheten är tillförlitlig. Sedan uppdaterar den dess enskilda lista med betrodda synkroniseringsidentiteter så att den ansökande inkluderas.

Det finns nu två medlemmar i signeringscirkeln och varje medlem har den publika nyckeln till den andra medlemmen. De börjar nu att utväxla enskilda nyckelringsobjekt via iCloud, eller lagringsutrymmet för nyckelvärden på iCloud, beroende på vad som är mest lämpligt i den aktuella situationen. Om båda cirkelmedlemmarna har uppdateringar för samma objekt väljs den ena eller den andra, vilket leder till eventuell konsekvens. Varje objekt som synkroniseras krypteras så att det bara kan avkrypteras av en enhet som finns i användarens tillförlitlighetscirkel. Det kan inte avkrypteras av någon annan enhet eller av Apple.

När nya enheter ansluter till synkroniseringscirkeln upprepas den här ”anslutningsprocessen”. När en tredje enhet ansluter kan den exempelvis parkopplas med vilken som helst av de befintliga enheterna. När nya enheter läggs till synkroniseras alla enheter med den nya enheten. Detta garanterar att alla har samma nyckelringsobjekt.

### Endast vissa objekt synkroniseras

En del nyckelringsobjekt är enhetsspecifika, exempelvis iMessage-nycklar, och måste därför stanna på enheten. I syfte att förhindra oväntad datatransport måste alla objekt som synkroniseras uttryckligen märkas med attributet `kSecAttrSynchronizable`.

Apple har ställt in det här attributet för Safari-användardata (inklusive användarnamn, lösenord och kreditkortsnummer), liksom för Wi-Fi-lösenord, HomeKit-krypteringsnycklar och andra nyckelringsobjekt som stöder heltäckande iCloud-kryptering.

Som förval synkroniseras inte nyckelringsobjekt som har lagts till av tredjepartsappar. Utvecklarna måste ställa in attributet `kSecAttrSynchronizable` när de lägger till objekt i nyckelringen.

## Säker återställning av iCloud-nyckelring

iCloud-nyckelring deponerar användarnas nyckelringsdata hos Apple utan att Apple får tillgång till lösenord eller andra data i dem. Även om användaren bara har en enda enhet ger återställning av nyckelringen ett skydd mot dataförlust. Det är särskilt viktigt när Safari används till att skapa slumpmässiga, starka lösenord eller nycklar för webbkonton eftersom dessa lösenord bara finns sparade i nyckelringen.

Sekundär autentisering och en säker deponeringstjänst som Apple har utvecklat särskilt för funktionen är centrala delar i tjänsten för återställning av nyckelringen. Användarens nyckelring krypteras med en stark lösenkod och deponeringstjänsten kräver att en strikt uppsättning villkor uppfylls för att en kopia av nyckelringen ska göras tillgänglig.

### Använda sekundär autentisering

Det går att upprätta en stark lösenkod på flera sätt:

- Om tvåfaktorsautentisering är aktiverad för användarens konto används enhetens lösenkod till att återskapa en deponerad nyckelring.
- Om tvåfaktorsautentisering inte är inställd blir användaren ombedd att skapa en iCloud-säkerhetskod genom att ange en sexsiffrig lösenkod. Utan tvåfaktorsautentisering kan användaren välja att ange en egen längre kod, eller låta enheten skapa en kryptografiskt slumpmässig kod som användaren kan registrera och spara på egen hand.

### Deponeringsprocess för nyckelring

När lösenkoden har upprättats deponeras nyckelringen hos Apple. iOS-, iPadOS- eller macOS-enheten exporterar först en kopia av användarens nyckelring. Sedan krypteras och paketeras den med nycklarna i en asymmetrisk nyckelsamling och placeras i användarens lagringsutrymme för nyckelvärden på iCloud. Nyckelsamlingen paketeras med användarens iCloud-säkerhetskod och den publika nyckeln för det HSM-modulkluster (Hardware Security Module) där deponeringsposten lagras. Detta blir användarens *deponeringspost på iCloud*. För konton med tvåfaktorsautentisering lagras nyckelringen också i CloudKit och paketeras till mellannycklar som endast går att hämta med innehållet i deponeringsposten på iCloud, vilket tillhandahåller samma skyddsnivå.

Innehållet i deponeringsposten tillåter även att återställningsenheten återansluter till iCloud-nyckelring, vilket bevisar för eventuella befintliga enheter att återställningsenheten har utfört deponeringsprocessen och därmed är auktoriserad av kontots ägare.

*Obs!* Utöver att upprätta en säkerhetskod måste användarna registrera ett telefonnummer för sitt iCloud-konto. Detta ger en andra autentiseringsnivå vid återställning av en nyckelring. Användaren får ett SMS som måste besvaras för att återställningen ska kunna fortsätta.

## Säkerhet vid deponering för iCloud-nyckelring

iCloud erbjuder en säker infrastruktur för deponering av nyckelringar för att säkerställa att endast behöriga användare och enheter kan utföra en återställning. Bakom iCloud finns kluster av HSM-moduler som skyddar deponeringsposterna. Enligt vad som beskrivits tidigare har varje kluster en nyckel som används till att kryptera deponeringsposterna som det ansvarar för.

För att återställa en nyckelring måste användaren ange användarnamn och lösenord till sitt iCloud-konto och svara på ett SMS som skickas till det telefonnummer som finns registrerat. När det är gjort måste användaren skriva in sin iCloud-säkerhetskod. HSM-klustret verifierar att användaren känner till sin iCloud-säkerhetskod med hjälp av SRP-protokollet (Secure Remote Password). Själva koden skickas inte till Apple. Varje medlem av klustret verifierar oberoende av de andra att användaren inte har överskridit antalet tillåtna försök att hämta posten (enligt beskrivningen nedan). Om majoriteten är överens packar klustret upp deponeringsposten och skickar den till användarens enhet.

Sedan använder enheten deponerade data till att packa upp de slumpmässiga nycklar som användarens nyckelring krypterades med. Med den nyckeln avkrypteras nyckelringen – som hämtats från CloudKit och lagringsutrymmet för nyckelvärden på iCloud – och återskapas på enheten. Deponeringstjänsten tillåter endast tio försök att autentisera och hämta en deponeringspost. Efter några misslyckade försök låses posten och användaren måste kontakta Apples support för att få fler försök. Efter det tionde misslyckade försöket förstör HSM-klustret deponeringsposten och nyckelringen går förlorad för alltid. Detta skyddar mot automatiserade intrångsförsök i syfte att hämta deponeringsposten. Priset för den säkerheten är att informationen i nyckelringen kan behöva offras.

Dessa policyer är kodade i den fasta HSM-programvaran. De administrativa åtkomstkorten som tillåter att den fasta programvaran ändras har förstörts. Om någon försöker göra ändringar i den fasta programvaran eller komma åt den privata nyckeln raderar HSM-klustret den privata nyckeln. Om detta skulle inträffa får ägaren av varje nyckelring som skyddas av klustret ett meddelande om att deras deponeringspost har gått förlorad. De kan då välja att registrera sig igen.

# Apple Pay

## Säkerhet för Apple Pay i översikt

Med Apple Pay går det att använda iPhone-, iPad-, Mac- och Apple Watch-enheter som stöds till att betala på ett enkelt, säkert och privat sätt i butiker, inuti appar och på webben i Safari. Användare kan också lägga till Apple Pay-kompatibla resekort, studentkort och passerkort i Plånbok. Det är enkelt för användarna och bygger på integrerad säkerhet i både maskin- och programvara.

Apple Pay är dessutom utformat för att skydda användarens personliga information. Apple Pay samlar inte in någon information om transaktioner som kan knytas till användaren. Alla transaktioner vid betalning sker mellan användaren, försäljaren och kortutfärdaren.

## Säkerhet för Apple Pay-komponent

Flera olika maskin- och programvarufunktioner används till att skapa säkra, tillförlitliga betalningar med Apple Pay.

### Secure Element

Secure Element är en certifierad krets av branschstandard som kör plattformen Java Card som uppfyller finanssektorns krav för elektroniska betalningar. Secure Element IC och plattformen Java Card certifieras i enlighet med säkerhetsutvärderingsprocessen EMVCo. När säkerhetsutvärderingen har slutförts med ett godkännande blir unika IC- och plattformscertifikat utfärdade av EMVCo.

Secure Element IC har certifierats i enlighet med Common Criteria-standarden.

### NFC-styrenhet

NFC-styrenheten hanterar protokoll för närfältskommunikation och dirigerar kommunikation mellan approcessorn och Secure Element samt mellan Secure Element och kassaterminalen.

### Plånbok

Appen Plånbok används till att lägga till och hantera kredit-, bank- och butikskort och till att betala med Apple Pay. Användarna kan se sina kort och eventuellt ytterligare information som tillhandahålls av kortutfärdaren, exempelvis kortutfärdarens integritetspolicy, de senaste transaktionerna med mera i Plånbok. Användarna kan också lägga till kort för Apple Pay i:

- Inställningsassistenten och Inställningar för iOS och iPadOS
- Apple Watch-appen för Apple Watch
- Plånbok och Apple Pay i Systeminställningar för Mac-datorer med Touch ID

Dessutom kan användare lägga till och hantera resekort, bonuskort, boardingkort, biljetter, presentkort, studentkort, passerkort och annat i Plånbok.

### Secure Enclave

På iPhone, iPad, Apple Watch, Mac-datorer med Touch ID och Mac-datorer med Apple Silicon som använder Magic Keyboard med Touch ID hanterar Secure Enclave autentiseringsprocessen och tillåter att en betalningstransaktion genomförs.

På Apple Watch måste enheten låsas upp och användaren måste dubbelklicka på sidoknappen. Dubbelklicket identifieras och vidarebefordras direkt till Secure Element, eller Secure Enclave där det är tillgängligt, utan att gå via approcessorn.

### Apple Pay-servrar

Apple Pay-servrarna hanterar inställning av och tillhandahåller kredit-, bank- och resekort, studentkort och passerkort i Plånbok. Servern hanterar också enhetens kontonummer som lagras i Secure Element. De kommunicerar både med enheten och med betalningsnätverkets eller kortutfärdarens servrar. Apple Pay-servrarna ansvarar också för att omkryptera betalningsuppgifterna vid betalningar inuti appar eller på webben.

## Så här skyddar Apple Pay användares köp

### Secure Element

Secure Element innehåller en specialutformad applet som hanterar Apple Pay. Det innehåller även appletar som är certifierade av betalningsnätverk eller kortutfärdare. Kredit- och bankkortsdata eller data från förbetalda kort skickas från betalningsnätverket eller kortutfärdaren till dessa appletar med nycklar som bara är kända av betalningsnätverket eller kortutfärdaren och appletens säkerhetsdomän. Dessa data lagras inom appletarna och skyddas med hjälp av säkerhetsfunktionerna i Secure Element. Under en transaktion kommunicerar terminalen direkt med Secure Element genom NFC-styrenheten (Near Field Communication) via en särskild maskinvarubuss.

### NFC-styrenhet

NFC-styrenheten fungerar som en gateway till Secure Element och ser till att alla kontaktfria betalningstransaktioner utförs med en kassaterminal som befinner sig nära enheten. Endast betalningsförfrågningar som kommer från en terminal som befinner sig inom fältet markeras av NFC-styrenheten som kontaktfria transaktioner.

När en betalning med ett kredit-, bank- eller förbetalt kort (inklusive butikskort) har auktoriserats av kortinnehavaren med Face ID, Touch ID eller en lösenkod, eller på en upplåst Apple Watch genom att dubbelklicka på sidoknappen, dirigerar styrenheten de kontaktlösa svaren som har förberetts av betalningsappletarna i Secure Element enbart till NFC-fältet. Det innebär att betalningsinformation för kontaktlösa betalningstransaktioner hålls inom det lokala NFC-fältet och aldrig exponeras för approcessorn. Betalningsinformation vid köp inuti appar och på webben dirigeras däremot till approcessorn, men skickas inte till Apple Pay-servern förrän den har krypterats av Secure Element.

# Kreditkort, bankkort och förbetalda kort

## Säkerhet för tillägg av kort i översikt

När en användare lägger till ett kredit- eller bankkort eller ett förbetalt kort (även butikskort) i Plånbok skickar Apple kortinformationen tillsammans med annan information om användarens konto och enhet till kortutfärdaren, eller kortutfärdarens auktoriserade tjänsteleverantör (vanligtvis betalningsnätverket). Med hjälp av den här informationen avgör kortutfärdaren (eller dess tjänsteleverantör) om den ska godkänna att kortet läggs till i Plånbok. Som en del av kortbetalningsprocessen använder Apple Pay tre serverstyrda anrop till att skicka och ta emot information från kortutfärdaren eller betalningsnätverket:

- Required Fields
- Check Card
- Link and Provision

Kortutfärdaren eller betalningsnätverket använder de här anropen till att göra det möjligt för kortutfärdaren att verifiera, godkänna och lägga till kort i Plånbok. Dessa klient-server-sessioner använder TLS 1.2 till att överföra data.

De fullständiga kortnumren lagras varken på enheten eller på Apple Pay-servrar. Istället skapas ett unikt enhetskontonummer som sedan krypteras och sparas i Secure Element. Enhetens unika kontonummer krypteras på ett sådant sätt att Apple inte har någon tillgång till det. Enhetens kontonummer är unikt och skiljer sig från de flesta kredit- och bankkortsnummer. Kortutfärdaren eller betalningsnätverket kan förhindra att det används vid betalning med magnetremsa, per telefon eller på webbplatser. Enhetens kontonummer i Secure Element lagras aldrig på Apple Pay-servrarna, säkerhetskopieras inte till iCloud och är isolerat från iOS-, iPadOS- och watchOS-enheter och från Mac-datorer med Touch ID och Mac-datorer med Apple Silicon som använder Magic Keyboard med Touch ID.

Kort som ska användas med Apple Watch tillhandahålls för Apple Pay med Apple Watch-appen på iPhone eller i en kortutfärdares iPhone-app. När du lägger till ett kort på Apple Watch måste klockan vara inom räckvidden för Bluetooth. Kort registreras specifikt för att användas med Apple Watch och har egna enhetskontonummer som lagras i Secure Element på Apple Watch.

När kredit-, bank- eller förbetalda kort (inklusive butikskort) läggs till visas de i en lista med kort medan inställningsassistenten körs på enheter som är inloggade med samma iCloud-konto. De här korten finns kvar i den här listan så länge de är aktiva på minst en enhet. Kort tas bort från den här listan efter att de har varit borttagna från alla enheter under sju dagar. Den här funktionen kräver att tvåfaktorsautentisering är aktiverad på respektive iCloud-konto.

## Lägga till kredit- eller bankkort i Apple Pay

Det går att lägga till kreditkort manuellt i Apple Pay på Apple-enheter.

### Lägga till kredit- eller bankkort manuellt

När du lägger till ett kort manuellt används namnet, kortnumret, sista giltighetsdatumet och CVV-koden för att förenkla tillhandahållandet. Användarna kan ange denna information i Inställningar, Plånbok eller Apple Watch-appen genom att använda enhetens kamera. När kameran tar en bild av kortinformationen försöker Apple att fylla i namnet, kortnumret och sista giltighetsdatum. Bilden sparas aldrig på enheten och lagras inte i bildbiblioteket. När alla fält är ifyllda verifieras alla fält utom CVV-fältet av processen Check Card. De krypteras sedan och skickas till Apple Pay-servern.

Om ett villkors-ID returneras med Check Card-processen hämtar och visar Apple kortutfärdarens villkor för användaren. Om användaren godkänner utfärdarens villkor skickar Apple ID:t för de godkända villkoren samt CVV-koden till processen Link and Provision. Som en del av Link and Provision-processen delar Apple information från enheten med kortutfärdaren eller nätverket. Det omfattar information om (a) användarens kontoaktivitet i iTunes och App Store (t.ex. om användaren har genomfört transaktioner under lång tid i iTunes), (b) användarens enhet (t.ex. telefonnummer, namn och enhetens modell plus eventuella andra Apple-enheter som krävs för att konfigurera Apple Pay) samt (c) ungefär var användaren befinner sig vid den tidpunkt när användaren lägger till kortet (om Platstjänster är aktiverat). Med hjälp av den här informationen avgör kortutfärdaren om den ska godkänna att kortet läggs till i Apple Pay.

Resultatet av processen Link and Provision är att två saker sker:

- Enheten börjar hämta den Plånbok-kupongfil som representerar kredit- eller bankkortet.
- Enheten börjar koppla kortet till Secure Element.

Kupongfilen innehåller URL:er för hämtning av kortbilder, metadata om kortet som kontaktinformation, den relaterade utfärdarens app och funktioner som stöds. Den innehåller också kupongens status, vilket omfattar information om ifall anpassningen av Secure Element är klar eller inte, om kortet för närvarande är spärrat av kortutfärdaren samt om ytterligare verifiering krävs innan kortet kan användas för betalningar med Apple Pay.

### Lägga till kredit- eller bankkort som har sparats i ett iTunes Store-konto

Om användaren har ett kredit- eller bankkort som har sparats i iTunes kan han/hon behöva ange sitt Apple-ID-lösenord igen. Kortnumret hämtas från iTunes och processen Check Card startas. Om kortet har godkänts för användning med Apple Pay hämtar och visar enheten kortutfärdarens villkor och skickar sedan ID:t för villkoren tillsammans med kortets säkerhetskod till Link and Provision-processen. Ytterligare verifiering kan krävas för kort som har sparats i ett iTunes-konto.

### Lägga till kredit- eller bankkort från en kortutfärdares app

När en app registreras för användning med Apple Pay skapas nycklar för appen och för kortutfärdarens server. Nycklarna används till att kryptera kortinformationen som skickas till kortutfärdaren. Det är utformat för att förhindra att informationen läses av Apple-enheten. Flödet liknar det som används för manuellt tillagda kort (vilket beskrivits tidigare), förutom att engångslösenord används i stället för CVV-numret.

### Lägga till kredit- eller bankkort från en kortutfärdares webbplats

En del kortutfärdare erbjuder möjligheten att inleda processen för att lägga till kort för Plånbok direkt från sin webbplats. I det fallet initierar användaren åtgärden genom att välja ett kort som ska läggas till på kortutfärdarens webbplats. Användaren omdirigeras sedan till en fristående Apple-inloggning (som finns i Apples domän) och blir ombedd att logga in med sitt Apple-ID. Efter inloggningen väljer användaren sedan en eller flera enheter där kortet ska läggas till och måste därefter bekräfta tillägget på respektive målenhet.

### Lägga till ytterligare verifiering

En kortutfärdare kan bestämma om ett kredit- eller bankkort behöver verifieras ytterligare. Beroende på vad kortutfärdaren erbjuder kan användaren ha möjlighet att välja mellan olika alternativ för ytterligare verifiering, till exempel SMS, e-post, samtal med kundtjänsten eller en metod i någon tredjepartsapp för att slutföra verifieringen. När det gäller SMS eller e-post får användaren möjlighet att välja bland den kontaktinformation som finns registrerad hos utfärdaren. En kod skickas och måste anges i Plånbok, Inställningar eller Apple Watch-appen. Utfärdaren hanterar kommunikationen vid samtal med kundtjänsten eller vid verifiering med hjälp av en app.

## Betalningsauktorisering med Apple Pay

För enheter med Secure Enclave kan en betalning genomföras endast efter att den har mottagit auktorisering från Secure Enclave. På en iPhone, iPad eller Mac som har Touch ID (eller är parkopplad med ett Magic Keyboard med Touch ID) omfattar det att bekräfta att användaren har autentiserat med biometrisk autentisering eller enhetens lösenkod eller lösenord. Biometrisk autentisering är den förvalda metoden, om sådan är tillgänglig, men autentisering med lösenkoden eller lösenordet kan användas när som helst och erbjuds automatiskt efter tre misslyckade försök att matcha ett fingeravtryck eller (för iPhone och iPad) två misslyckade försök att matcha ett ansikte. Efter fem misslyckade försök krävs lösenkoden efter lösenordet. En lösenkod eller ett lösenord krävs även när biometrisk autentisering inte har konfigurerats eller inte slagits på för Apple Pay. För att en betalning ska kunna genomföras med Apple Watch måste enheten låsas upp med lösenkoden och användaren måste dubbelklicka på sidoknappen.

### Använda en delad parkopplingsnyckel

Secure Enclave och Secure Element kommunicerar via ett seriellt gränssnitt som krypteras och autentiseras baserat på AES och använder kryptografiska anti-replay-värden som skydd mot replay-attacker. Trots att sidorna inte är direkt anslutna till varandra kan de kommunicera säkert med en delad parkopplingsnyckel som tillhandahålls under tillverkningsprocessen. Under den processen genererar Secure Enclave parkopplingsnyckeln från dess UID-nyckel och från Secure Elements unika identifierare. Sedan överförs parkopplingsnyckeln säkert till en HSM-modul (Hardware Security Modul) i fabriken. Sedan infogar HSM parkopplingsnyckeln i Secure Element.

### Auktorisera en säker transaktion

När användaren auktoriserar en transaktion, vilken omfattar en fysisk gest som förmedlas direkt till Secure Enclave, skickar Secure Enclave sedan signerade data om typen av autentisering och detaljer om typen av transaktion (kontaktfri eller inuti appar) till Secure Element, kopplad till ett AR-värde (Authorization Random). AR-värdet genereras i Secure Enclave när en användare först lägger till ett kreditkort och sparas så länge Apple Pay är aktiverat samt skyddas av Secure Enclave-krypteringen och en bakåtspärr. Det levereras på ett säkert sätt till Secure Element genom att använda parkopplingsnyckeln. När Secure Element tar emot ett nytt AR-värde markeras alla tidigare tillagda kort som avslutade.

### Använda ett betalningskryptogram för dynamisk säkerhet

Betalningstransaktioner som har sitt ursprung i betalningsappletar innehåller ett betalningskryptogram tillsammans med ett kontonummer för enheten. Det här kryptogrammet är en engångskod som beräknas med en transaktionsräknare och en nyckel. Transaktionsräknaren ökar i värde för varje ny transaktion. Nyckel tillhandahålls i betalningsappleten under anpassningen och är känd av betalningsnätverket eller kortutfärdaren eller båda. Beroende på betalningsschemat kan också andra data användas vid beräkningen, däribland:

- Ett TUN-nummer (Terminal Unpredictable Number) för NFC-transaktioner
- Ett anti-replay-värde för Apple Pay-servern för transaktioner inuti appar
- Resultat för användarverifikation, till exempel Cardholder Verification Method (CVM)-information

Dessa säkerhetskoder skickas till betalningsnätverket och kortutfärdaren så att utfärdaren kan verifiera alla transaktioner. Längden på dessa säkerhetskoder kan variera beroende på typ av transaktion.

## Betala med kort via Apple Pay

Apple Pay kan användas till att betala för inköp i butiker, i appar och på webbplatser.

### Betala med kort i butiker

Om iPhone eller Apple Watch är på och den upptäcker ett NFC-fält visar den det begärda kortet för användaren (om automatiskt val är aktiverat för det kortet) eller det förvalda kortet som hanteras i Inställningar. Användaren kan även öppna Plånbok och välja ett kort eller göra följande när enheten är låst:

- Dubbelklicka på sidoknappen på enheter med Face ID.
- Dubbelklicka på hemknappen på enheter med Touch ID.
- Använda hjälpmedelsfunktioner som gör att du kan använda Apple Pay från låsskärmen.

Sedan måste användaren autentisera med Face ID, Touch ID eller sin lösenkod innan informationen överförs. När Apple Watch är upplåst aktiveras det förvalda kortet för betalning när användaren dubbelklickar på sidoknappen. Ingen betalningsinformation skickas utan användarens autentisering.

När användaren har autentiserat transaktionen används enhetens unika kontonummer och en transaktionsspecifik dynamisk säkerhetskod för att behandla betalningen. Varken Apple eller användarens enhet skickar det fullständiga kredit- eller bankkortsnumret till försäljare. Apple kan ta emot anonym transaktionsinformation, som ungefärlig tid och plats för transaktionen, vilket hjälper till att förbättra Apple Pay och andra produkter och tjänster från Apple.

## Betala med kort i appar

Apple Pay kan även användas till att utföra betalningar i iOS-, iPadOS-, macOS- och watchOS-appar. När användare betalar i appar med Apple Pay tar Apple emot den krypterade transaktionsinformationen och dirigerar den till utvecklaren eller försäljaren. Innan den informationen skickas vidare till utvecklaren eller försäljaren omkrypterar Apple transaktionen med en utvecklarspecifik nyckel. Apple Pay sparar anonym transaktionsinformation, t.ex. det ungefärliga beloppet. Den här informationen kan inte kopplas till användaren och talar aldrig om vad användaren köper.

När en app initierar en Apple Pay-betalning mottar Apple Pay-servrarna den krypterade transaktionen från enheten innan försäljaren tar emot den. Apple Pay-servrarna krypterar sedan om transaktionen med försäljarens specifika nyckel innan den skickas vidare till försäljaren.

När appen begär en betalning anropar den ett API för att avgöra om enheten stöder Apple Pay och om användaren har ett kredit- eller bankkort som kan utföra betalningar i det betalningsnätverk som försäljaren använder. Appen frågar efter den information den behöver för att behandla och fullfölja transaktionen, till exempel fakturerings- och leveransadressen samt kontaktinformation. Appen ber sedan iOS, iPadOS, macOS eller watchOS att visa Apple Pay-bladet som begär information åt appen och annan nödvändig information, t.ex. vilket kort som ska användas.

Appen förses nu med information om ort och postnummer för slutlig beräkning av fraktkostnaden. All den information som har begärts lämnas inte ut till appen förrän användaren godkänner betalningen med hjälp av Face ID, Touch ID eller enhetens lösenkod. När betalningen auktoriseras överförs informationen som visas på Apple Pay-bladet till försäljaren.

## Betala med kort inuti appklipp

Ett appklipp är en liten del av en app som gör det möjligt för användare att snabbt utföra en uppgift (som att hyra en cykel eller betala för parkering) utan att hämta den fullständiga appen. Om ett appklipp stöder betalningar kan användaren använda Logga in med Apple och sedan betala med Apple Pay. När en användare gör en betalning inuti ett appklipp är alla säkerhets- och integritetsåtgärder samma som när en användare betalar inuti en app.

## Hur användare autentiserar och försäljare verifierar appbetalningar

Användare och försäljare säkerställer säkra appbetalningar genom att överföra information till Apples servrar, Secure Element, enheten och appens API. När användaren inledningsvis auktoriserar en appbetalning erhåller appen ett kryptografiskt anti-replay-värde genom att anropa Apple Pay-servrarna. Servrarna skickar det här värdet och andra transaktionsdata till Secure Element som beräknar betalningsuppgifter som krypteras med en Apple-nyckel. Secure Element returnerar sedan betalningsuppgifterna till Apple Pay-servrarna så att de kan avkryptera dem, verifiera anti-replay-värdet mot det anti-replay-värde som Apple Pay-servrarna ursprungligen skickade och krypterar dem på nytt med den försäljarnyckel som är associerad med försäljarens ID. Apple-servrarna returnerar sedan betalningen till enheten som överlämnar den till appens API så att API:t kan överföra den till försäljarens system för bearbetning. Försäljaren avkrypterar betalningsuppgifterna för att verifiera att den är rätt mottagare till transaktionen.

API:erna kräver en behörighet som anger försäljarens ID-uppgifter. En app kan också bifoga ytterligare data (till exempel ett ordernummer eller kundens identitet) som skickas till Secure Element för signering så att transaktionen inte kan dirigeras om till en annan kund. Det här åstadkoms av apputvecklaren som kan ange applicationData i PKPaymentRequest. En hashkod för dessa data inkluderas i den krypterade betalningsinformationen. Försäljaren är sedan ansvarig för att verifiera att användarens hashkod för applicationData matchar det som ingår i betalningsinformationen.

## Betala med kort på webbplatser

Apple Pay kan användas till att utföra betalningar på webbplatser på iPhone, iPad, Apple Watch och Mac-datorer med Touch ID. Apple Pay-transaktioner kan även inledas på en Mac och sedan slutföras på en Apple Pay-aktiverad iPhone eller Apple Watch som använder samma iCloud-konto.

Apple Pay på webben kräver att alla webbplatser som deltar registreras hos Apple. När domänen har registrerats utförs validering av domännamnet endast efter att Apple har utfärdat ett TLS-klientcertifikat. Webbplatser som stöder Apple Pay måste leverera sitt innehåll via HTTPS. För varje betalningstransaktion måste webbplatserna erhålla en säker och unik säljarsession via en Apple-server genom att använda det TLS-klientcertifikat som Apple har utfärdat. Säljarsessionsdata signeras av Apple. När en säljarsessionssignatur har verifierats kan webbplatsen skicka en förfrågan om användaren har en enhet förberedd för Apple Pay och om ett kredit- eller kontokort eller ett förbetalt kort är aktiverat på enheten. Inga andra detaljer delas. Om användaren inte vill dela den här informationen kan denna avaktivera Apple Pay-förfrågningar i integritetsinställningarna för Safari på iPhone-, iPad- och Mac-enheter.

När en försäljarsession har validerats är alla integritets- och säkerhetsåtgärder samma som när en användare betalar inuti en app.

Om användaren överför betalningsrelaterad information från en Mac till en iPhone eller Apple Watch använder Apple Pay det heltäckande krypterade IDS-protokollet (Apple Identity Service) till att överföra betalningsrelaterad information mellan användarens dator och den auktoriserande enheten. IDS-klienten på en Mac använder användarens enhetsnycklar till att utföra krypteringen så att inga andra enheter kan avkryptera informationen, och nycklarna är inte tillgängliga för Apple. Enhetsupptäckt för Apple Pay-överlämning innehåller typ och unik identifierare för användarens kreditkort tillsammans med vissa metadata. Enhetskontonumret för användarens kort delas inte, och det fortsätter att lagras säkert på användarens iPhone eller Apple Watch. Apple överför också användarens senast använda kontaktuppgifter och leverans- och faktureringsadress säkert via iCloud-nyckelring.

När användaren har auktoriserat en betalning via Face ID, Touch ID, en lösenkod eller genom att dubbelklicka på sidoknappen på Apple Watch överförs en betalningstoken som är unikt krypterad säkert till varje webbplats försäljarcertifikat från användarens iPhone eller Apple Watch till datorn och levereras sedan till försäljarens webbplats.

Endast enheter som finns i närheten av varandra kan begära och slutföra betalningar. Närheten bestäms genom Bluetooth LE-annonsering (BLE).

### Automatiska betalningar och försäljartokens

I iOS 16 eller senare kan appar och webbplatser som erbjuder Apple Pay dra nytta av Apple Pay-försäljartokens som möjliggör säkra betalningar på ett enhetligt sätt på en användares olika enheter. Det uppdaterade Apple Pay-betalningsbladet i iOS 16 optimerar också förauktoriserade betalningsupplevelser. Nya transaktionstyper i Apple Pay-API:t gör det möjligt för app- och webbplatsutvecklare att finjustera upplevelsen i betalningsbladet för abonnemang, återkommande fakturor, avbetalningar och automatisk påfyllning av kortsaldon.

Försäljartokens är inte enhetsspecifika, så återkommande betalningar avbryts inte om användaren tar bort ett kontokort från enheten.

### Betalningar till flera försäljare

I iOS 16 eller senare innehåller Apple Pay möjligheten att ange inköpsbelopp för flera försäljare inuti ett enda Apple Pay-betalningsblad. Det ökar flexibiliteten för kunder som kan göra ett paketköp, till exempel en resa med flyg, hyrbil och hotell, och sedan skicka betalningar till enskilda försäljare.

## Kontaktlösa kuponger i Apple Pay

För att överföra data från kuponger som stöds till kompatibla NFC-terminaler används protokollet Apple Value Added Services (Apple VAS). VAS-protokollet kan implementeras på kontaktlösa terminaler eller i iPhone-appar och NFC används för kommunikationen med de Apple-enheter som stöds. VAS-protokollet fungerar på korta avstånd och kan användas till att lösa in kontaktlösa kuponger fristående eller som en del av en Apple Pay-transaktion.

När enheten hålls nära NFC-terminalen inleder terminalen överföringen av kuponginformationen genom att skicka en kupongförfrågan. Om användaren har en kupong med butikens ID blir användaren ombedd att bekräfta att den ska användas med Face ID, Touch ID eller lösenkod. Kuponginformationen, en tidsstämpel och en slumpmässig ECDH P-256-engångsnyckel används med kupongleverantörens offentliga nyckel för att härleda en krypteringsnyckel för kupongens data som sedan skickas till terminalen.

Från iOS 12.0.1 till och inklusive iOS 13 kan användarna manuellt välja en kupong innan de lämnar över den till försäljarens NFC-terminal. I iOS 13.1 och senare kan kupongleverantörerna konfigurera att manuellt valda kuponger antingen ska kräva användarautentisering eller kunna användas utan autentisering.

## Göra kort oanvändbara med Apple Pay

Kredit- och bankkort och förbetalda kort som lagts till i Secure Element kan bara användas om Secure Element tar emot en auktorisering med samma parkopplingsnyckel och AR-värde (Authorization Random) som när kortet lades till. När Secure Element tar emot ett nytt AR-värde markeras alla tidigare tillagda kort som avslutade. Detta innebär att operativsystemet kan instruera Secure Enclave att göra korten oanvändbara genom att markera deras kopia av AR-värdet som ogiltigt under följande omständigheter:

| Metod | Enhet |
|---|---|
| Lösenkoden avaktiveras. | iPhone, iPad, Apple Watch |
| Lösenordet avaktiveras. | Mac |
| Användaren loggar ut från iCloud. | iPhone, iPad, Mac, Apple Watch |
| Användaren väljer Radera allt innehåll och inst. | iPhone, iPad, Mac, Apple Watch |
| Enheten återskapas från återställningsläge. | iPhone, iPad, Mac, Apple Watch |
| Tar bort parkoppling | Apple Watch |

### Spärra, ta bort och radera kort

Användaren kan spärra Apple Pay på iPhone, iPad och Apple Watch genom att försätta enheten i förlorat läge i Hitta. Användaren har också möjlighet att ta bort och radera sina kort från Apple Pay via Hitta, på iCloud.com eller direkt på enheten i Plånbok. Kort kan tas bort i iCloud-inställningarna på Apple Watch, i Apple Watch-appen på iPhone eller direkt på klockan. Möjligheten att betala med kort från enheten stängs av eller tas bort från Apple Pay av kortutfärdaren eller respektive betalningsnätverk, även om enheten är nedkopplad och inte ansluten till ett mobilnät eller ett Wi-Fi-nätverk. Användaren kan även ringa kortutfärdaren för att stänga av eller ta bort kort från Apple Pay.

När en användare raderar hela enheten, antingen med Radera allt innehåll och inst., i Hitta eller genom att återskapa enheten (iPhone, iPad, Mac och Apple Watch) uppmanas Secure Element att markera alla kort som avslutade. Effekten av detta är att korten omedelbart blir oanvändbara, i väntan på att Apple Pay-servrarna kan kontaktas och helt radera korten från Secure Element. Oberoende av detta markerar Secure Element AR-värdet som ogiltigt så att inga betalningsgodkännanden för tidigare registrerade kort längre är möjliga. När enheten är online försöker den kontakta Apple Pay-servrarna för att säkerställa att alla kort i Secure Element har raderats.

# Säkerhet och Apple Card

På iPhone- och Mac-modeller som stöds kan en användare ansöka om ett Apple Card på ett säkert sätt.

## Apple Card-ansökan

I iOS 12.4 eller senare, macOS 10.14.6 eller senare och watchOS 5.3 eller senare kan Apple Card användas med Apple Pay till att betala i butiker, appar och på webben.

För att ansöka om ett Apple Card måste användaren vara inloggad på sitt iCloud-konto på en Apple Pay-kompatibel iPhone eller iPad och ha ställt in tvåfaktorsautentisering för iCloud-kontot. Användaren kan också ansöka på apply.applecard.apple efter inloggning till sitt Apple-ID. När ansökan är godkänd blir Apple Card tillgängligt i Plånbok eller under Inställningar > Plånbok och Apple Pay på alla kompatibla enheter där användaren har loggat in med sitt Apple-ID.

När en användare ansöker om ett Apple Card verifieras användaridentiteten på ett säkert sätt av de identitetsleverantörer som Apple samarbetar med och delas sedan med Goldman Sachs Bank USA för identifikation och kreditbedömning.

Information som personnummer eller ID-dokumentbilder som tillhandahålls under ansökan skickas på ett säkert sätt till de identitetsleverantörer som Apple samarbetar med och/eller Goldman Sachs Bank USA och krypteras med deras respektive nycklar. Apple kan inte avkryptera den här informationen.

Den inkomstinformation som anges under ansökan, och den bankkontoinformation som används för fakturabetalningar, överförs till Goldman Sachs Bank USA och är krypterad med deras nyckel. Bankkontoinformationen sparas i nyckelringen. Apple kan inte avkryptera den här informationen.

När Apple Card läggs till i Plånbok kan samma information som en användare anger när den lägger till ett kredit- eller bankkort komma att delas med Apples partnerbank Goldman Sachs Bank USA och med Apple Payments Inc. Den här informationen används endast vid felsökning samt i syfte att förhindra bedrägerier och uppfylla lagstadgade krav.

I iOS 14.6 eller senare, iPadOS 14.6 eller senare och watchOS 7.5 eller senare kan samordnaren i en iCloud-familj som har ett Apple Card dela kortet med sina iCloud-familjemedlemmar som är 13 år eller äldre. Användaren måste autentisera sig för att bekräfta sin inbjudan. Plånbok använder en nyckel i Secure Enclave till att beräkna en signatur som sammankopplar ägaren och den inbjudna. Den signaturen valideras på Apple-servrar.

Samordnaren kan även välja att ställa in en transaktionsgräns för deltagarna. Deltagarkort kan också låsas för att pausa transaktioner när som helst genom Plånbok. När en delägare eller deltagare över 18 år tackar ja till en inbjudan och ansöker går de igenom samma ansökningsprocess som definieras i avsnittet om Apple Card-ansökan i Plånbok.

## Apple Card-användning

Ett fysiskt kort kan beställas från Apple Card i Plånbok. När användaren har fått det fysiska kortet aktiveras det med den NFC-tagg som finns i kuvertet till det fysiska kortet. Taggen är unik för varje enskilt kort och kan inte användas till att aktivera en annan användares kort. Kortet kan även aktiveras manuellt i Plånbok-inställningarna. Dessutom kan användaren också när som helst låsa eller låsa upp det fysiska kortet via Plånbok.

### Apple Card-betalningar och Apple Wallet-kuponginformation

Betalningar från Apple Card-kontot kan göras i en webbläsare eller Plånbok i iOS med Apple Cash och ett bankkonto. Fakturabetalningar kan schemaläggas som återkommande eller som engångsbetalningar vid ett visst datum med Apple Cash och ett bankkonto. Vid en betalning görs ett anrop till Apple Pay-servrarna för att få ett kryptografiskt anti-replay-värde som liknar Apple Cash. Anti-replay-värdet överförs tillsammans med informationen om betalningsinställningar till Secure Element som beräknar en signatur. Signaturen returneras sedan till Apple Pay-servrarna. Betalningens autentisering, integritet och korrekthet verifieras genom signaturen och anti-replay-värdet av Apple Pay-servrar och beställningen skickas till Goldman Sachs Bank USA för bearbetning.

Apple Card-numret hämtas av Plånbok genom att presentera ett certifikat. Apple Pay-servern validerar certifikatet för att bekräfta att nyckeln genererades i Secure Enclave. Sedan använder det den här nyckeln till att kryptera Apple Card-numret innan det returneras till Plånbok så att endast den iPhone som begärde Apple Card-numret kan avkryptera det. Efter avkrypteringen sparas Apple Card-numret i iCloud-nyckelring.

För att visa Apple Card-nummerinformationen i kupongen i Plånbok krävs användarautentisering med Face ID, Touch ID eller en lösenkod. Användaren kan byta alternativ i kortinformationen, varpå det gamla alternativet avaktiveras.

### Avancerat bedrägeriskydd

I iOS 15 eller senare och iPadOS 15 eller senare kan Apple Card-användare aktivera Avancerat bedrägeriskydd i Plånbok. När det är aktiverat uppdateras kortets säkerhetskod med några dagars mellanrum.

## Säkerhet och Apple Cash

I iOS 11.2 och senare, iPadOS 13.1 och senare och watchOS 4.2 och senare kan Apple Cash användas på en iPhone, iPad eller Apple Watch till att skicka, ta emot och begära pengar från andra användare. När en användare tar emot pengar läggs de till i ett Apple Cash-konto som kan nås i Plånbok eller i Inställningar > Plånbok och Apple Pay på valfri kompatibel enhet där användaren har loggat in med sitt Apple-ID.

I iOS 14, iPadOS 14 och watchOS 7 kan samordnaren för en iCloud-familj som har verifierat sin identitet med Apple Cash aktivera Apple Cash för familjemedlemmar som är yngre än 18 år. Alternativt kan samordnaren begränsa möjligheten för dessa användare att skicka pengar till endast familjemedlemmar eller endast kontakter. Om en familjemedlem som är yngre än 18 år går igenom en Apple-ID-kontoåterställning måste familjesamordnaren manuellt återaktivera Apple Cash-kortet för den användaren. Om familjemedlemmar som är yngre än 18 år upphör att ingå i iCloud-familjen överförs deras Apple Cash-saldo automatiskt till samordnarens konto.

När användaren ställer in Apple Cash kan samma information som när användaren lägger till ett kredit- eller bankkort delas med vår partnerbank Green Dot Bank och med Apple Payments Inc. som är ett helägt dotterbolag som skapades för att skydda användarens integritet genom att lagra och bearbeta information separat från övriga Apple och på ett sätt som övriga Apple inte känner till. Den här informationen används endast vid felsökning samt i syfte att förhindra bedrägerier och uppfylla lagstadgade krav.

## Använda Apple Cash i iMessage

För att göra betalningar mellan personer och använda Apple Cash måste användaren vara inloggad på sitt iCloud-konto på en Apple Cash-kompatibel enhet och ha ställt in tvåfaktorsautentisering för iCloud-kontot. Förfrågningar om pengar och överföringar mellan användare inleds i appen Meddelanden eller genom att fråga Siri. När en användare försöker skicka pengar visas Apple Pay-bladet av iMessage. Apple Cash-saldot används alltid först. Om det behövs dras resterande belopp från ett annat kredit- eller bankkort som användaren har lagt till i Plånbok.

## Använda Apple Cash i butiker, appar och på webben

Apple Cash-kortet i Plånbok kan användas med Apple Pay till att utföra betalningar i butiker, i appar och på webben. Pengarna på Apple Cash-kontot kan även överföras till ett bankkonto. Utöver att ta emot pengar från en annan användare kan pengar läggas till på Apple Cash-kontot från ett bankkort eller förbetalt kort i Plånbok.

Apple Payments Inc. lagrar och kan använda användarens transaktionsinformation vid felsökning samt i syfte att förhindra bedrägerier och uppfylla lagstadgade krav när en transaktion har genomförts. Övriga Apple vet inte vem eller vilka användaren skickar pengar till, tar emot pengar från eller var användaren har betalat med sitt Apple Cash-kort.

När användaren skickar pengar med Apple Pay, lägger till pengar på ett Apple Cash-konto eller överför pengar till ett bankkonto görs ett anrop till Apple Pay-servrarna för att få ett kryptografiskt anti-replay-värde som liknar värdet som returneras för Apple Pay inuti appar. Anti-replay-värdet och andra transaktionsdata överförs till Secure Element som beräknar en betalningssignatur. Signaturen returneras sedan till Apple Pay-servrarna. Transaktionens autenticitet, integritet och riktighet verifieras av Apple Pay-servrarna via betalningssignaturen och anti-replay-värdet. Därefter inleds pengaöverföringen och användaren får ett meddelande om den slutförda transaktionen.

Om transaktionen involverar:

- Ett bankkort för att lägga till pengar i Apple Cash
- Att tillhandahålla mer pengar om Apple Cash-saldot är otillräckligt

En krypterad betalningsuppgift produceras också och skickas till Apple Pay-servrar, på ett liknande sätt som Apple Pay fungerar i appar och på webbplatser.

När saldot på Apple Cash-kontot överskrider ett visst belopp, eller om ovanlig aktivitet upptäcks, uppmanas användaren att verifiera sin identitet. Informationen som används till att verifiera användarens identitet – t.ex. personnummer eller svar på frågor (exempelvis namnet på en gata som användaren har bott på) – skickas till Apples partner på ett säkert sätt och krypteras med partnerns nyckel. Apple kan inte avkryptera den här informationen. Användaren blir uppmanad att verifiera sin identitet igen om denna utför en Apple-ID-kontoåterställning innan han eller hon återfår tillgången till sitt Apple Cash-saldo.

# Säkerhet för Tap to Pay on iPhone

Med Tap to Pay on iPhone i iOS 15.4 eller senare kan försäljare ta emot Apple Pay och andra kontaktlösa betalningar genom att använda iPhone och en partneraktiverad iOS-app. Med den här tjänsten kan användare med iPhone-enheter som stöds ta emot kontaktlösa betalningar och NFC-aktiverade Apple Pay-kuponger på ett säkert sätt. Med Tap to Pay on iPhone behöver försäljare ingen ytterligare maskinvara för att ta emot kontaktlösa betalningar.

Tap to Pay on iPhone är utformat för att skydda betalarens personliga information. Tjänsten samlar inte in transaktionsinformation som kan knytas till betalaren. Kontokortsinformation, som kredit-/bankkortsnumret (PAN), skyddas av Secure Element och är inte synlig för försäljarens enhet. Kontokortsinformation stannar mellan försäljarens betaltjänsteleverantör och betalaren och kortutfärdaren. Utöver detta samlar Tap to Pay-tjänsten inte in betalarens namn, adresser eller telefonnummer.

Tap to Pay på iPhone har utvärderats externt av ett ackrediterat säkerhetslaboratorium och godkänts för användning av alla godkända betalningsnätverk i territorierna där det är tillgängligt.

## Säkerhet för kontaktlösa betalningskomponenter

- *Secure Element:* Secure Element är värd för betalningskärnorna som läser och skyddar kontaktlösa betalningskortdata.
- *NFC-styrenhet:* NFC-styrenheten hanterar NFC-protokoll (Near Field Communication) och dirigerar kommunikation mellan approcessorn och Secure Element samt mellan Secure Element och det kontaktlösa betalningskortet.
- *Tap to Pay on iPhone-servrar:* Tap to Pay on iPhone-servrarna hanterar inställning av och tillhandahåller betalningskärnorna i enheten. Servrarna övervakar även säkerheten i Tap to Pay på iPhone-enheter på ett sätt som är kompatibelt med Contactless Payments on COTS (CPoC)-standarden från Payment Card Industry Security Standards Council (PCI SSC) och är PCI DSS-kompatibelt.

## Så här läser Tap to Pay av kreditkort, bankkort och förbetalda kort

### Så här tillhandahålls säkerhet i Tap to Pay

Vid första användningen av Tap to Pay on iPhone med en behörig app fastställer Tap to Pay on iPhone-servrarna om enheten uppfyller kompatibilitetskrav som enhetsmodell, iOS-version och ifall en lösenkod har ställts in. När verifieringen har genomförts hämtas appleten för betalningsmottagande från Tap to Pay on iPhone-servrarna och installeras i Secure Element tillsammans med tillhörande betalningskärnkonfiguration. Den här åtgärden utförs säkert mellan Tap to Pay on iPhone-servrarna och Secure Element. Secure Element validerar integriteten och autenticiteten hos dessa data inför installationen.

### Så här läser Tap to Pay kort säkert

När en Tap to Pay on iPhone-app begär en kortläsning från ramverket ProximityReader visas ett blad (som styrs av iOS) där användaren blir uppmanad att trycka på ett kontokort. Inga appar kan läsa av sensorer som kan avslöja någon del av känsliga kortdata medan tryckskärmen är aktiv. iOS initierar kontokortsläsaren och begär sedan att betalningskärnorna i Secure Element initierar en kortläsning.

I det här stadiet tar Secure Element kontroll över NFC-styrenheten i läsarläge. I det här läget kan kortdata endast utväxlas mellan betalningskortet och Secure Element via NFC-styrenheten. Betalningskort kan endast läsas i det här läget.

När appleten för betalningsmottagande i Secure Element har slutfört betalningskortläsningen krypterar den och signerar kortdata. Kontokortsdata förblir krypterade och autentiserade tills de når betaltjänsteleverantören. Endast betaltjänsteleverantören som användes av appen till att begära kortläsningen kan avkryptera dessa kontokortsdata. Betaltjänsteleverantören måste begära krypteringsnyckeln för kontokortsdata från Tap to Pay on iPhone-servrarna. Tap to Pay på iPhone-servrarna utfärdar avkrypteringsnycklar till betaltjänsteleverantören efter validering av dataintegritet och -autenticitet och efter verifiering att kortläsningen skedde inom 60 sekunder efter förfrågan om avkrypteringsnyckeln för betalningskortdata.

Den här modellen säkerställer att betalningskortdata inte kan avkrypteras av någon annan än betaltjänsteleverantören som bearbetar transaktionen åt försäljaren.

### Använda PIN-koden till att auktorisera transaktioner

PIN-kodsinmatning, som finns i iOS 16.0 eller senare, gör det möjligt för betalaren att auktorisera en transaktion genom att mata in sin PIN-kod på försäljarens enhet. Skärmen för PIN-kodsinmatning kan lösas ut omedelbart efter trycket baserat på informationen som utväxlas med betalningskortet. Betaltjänsteleverantören kan också lösa ut PIN-kodsskärmen genom att tillhandahålla en signerad token som endast är giltig för en transaktion.

Mekanismen för PIN-kodsinmatning har utvärderats externt av ett ackrediterat säkerhetslaboratorium och godkänts för användning av alla godkända betalningsnätverk i territorierna där det är tillgängligt. Skärmen för PIN-kodsinmatning skyddas mot skärmavbildning och bildskärmsdubblering och inga appar kan läsa av någon sensor som kan avslöja någon del av PIN-kodsvärdet så länge skärmen för PIN-kodsinmatning är aktiv.

Siffrorna i PIN-koden som anges registreras säkert av Secure Element. Med dessa PIN-kodssiffror skapar Secure Element ett krypterat PIN-kodsblock som är kompatibelt med betalningsbranschstandard. Apple tillhandahåller det krypterade PIN-kodsblocket på ett säkert sätt från dess PCI-PIN-kodskompatibla back-end till betaltjänsteleverantören för vidare bearbetning.

PIN-kodsvärdet blir:

- Aldrig tillgängligt för försäljaren på dennas enhet
- Aldrig avkrypterat av Apple vid någon tidpunkt
- Aldrig lagrat av Apple

# Använda Plånbok

## Åtkomst med Plånbok

Användare kan lagra flera typer av nycklar i Plånbok på iPhone- och Apple Watch-enheter som stöds. När en användare når en dörr kan rätt nyckel till och med visas automatiskt (om expressläge stöds av nyckeln och har aktiverats) så att dörren smidigt kan öppnas med ett tryck via NFC.

### Enkelhet för användare

#### Expressläge

När en nyckel läggs till i Plånbok aktiveras expressläget som förval. Nycklar i expressläge fungerar med kompatibla terminaler utan Face ID, Touch ID, lösenkodsautentisering eller dubbelklick på sidoknappen på Apple Watch. Användare kan avaktivera den här funktionen om de stänger av expressläget genom att trycka på merknappen på framsidan av kortet som representerar nyckeln i Plånbok. Om de vill slå på expressläget igen måste de använda Face ID, Touch ID eller en lösenkod.

#### Nyckeldelning

I iOS 16 eller senare är nyckeldelning tillgänglig för vissa nyckeltyper.

Användare kan dela tillgången till en nyckel (exempelvis en hem- eller bilnyckel) med säkerhet och integritet som genomdrivs från nyckelägarens iPhone till den inbjudna nyckelmottagarens iPhone. Nycklar delas genom att trycka på delningssymbolen på nyckeln i Plånbok och kan delas med metoder som visas på delningsbladet. Nyckelägare kan också välja åtkomstnivå och giltig tidsperiod för varje delad nyckel. Nyckelägaren har översikt över alla nycklar den har delat och kan återkalla åtkomsten för valfria delade nycklar, inklusive tillfällen när den första nyckelmottagaren i sin tur delar nyckeln igen till en annan användare.

En nyckeldelningsinbjudan lagras anonymiserat och säkert av en dedikerad server inuti en brevlåda och skyddas med en AES 128- eller 256-krypteringsnyckel. Krypteringsnyckeln delas aldrig med servern eller någon annan, förutom den avsedda nyckelmottagaren, och bara nyckelmottagaren kan avkryptera sin inbjudan. När brevlådan skapas tillhandahåller nyckelägarens iPhone en enhetsfordran som är kopplad till endast den brevlådan av servern. När nyckelmottagarens iPhone ansluter till denna brevlåda för första gången presenterar den en enhetsfordran för nyckelmottagare. Endast nyckelägarens och nyckelmottagarens iPhone-enheter som presenterar giltiga enhetsfordringar kan komma åt den brevlådan. Varje iPhones enhetsfordran har ett unikt UUID-värde i enlighet med RFC4122.

Som en ytterligare säkerhetsåtgärd kan nyckelägaren slå på en sexsiffrig slumpmässigt genererad aktiveringskod som krävs på nyckelmottagarens iPhone. Antalet kodförsök genomdrivs och valideras av antingen nyckelägaren eller en partnerserver. Den här aktiveringskoden måste förmedlas av nyckelägaren till nyckelmottagaren och nyckelmottagaren måste ange koden vid uppmaning för validering av antingen nyckelägaren eller en partnerserver.

När en inbjudan har tagits emot av nyckelmottagaren raderas den omedelbart från servern av den iPhone som tar emot den. Brevlådan som innehåller en nyckeldelningsinbjudan har även den en begränsad livslängd som ställs in när den skapas och genomdrivs av servern. Inbjudningar som löper ut raderas automatiskt av servern.

Beroende på originaltillverkaren kan nycklar även delas med enheter som inte kommer från Apple, men deras metod för säker nyckeldelning kan vara annorlunda än Apples.

## Integritet och säkerhet

Åtkomstnycklar i Plånbok drar full nytta av de inbyggda integritets- och säkerhetsfunktionerna i iPhone och Apple Watch. När eller var någon använder nycklarna i Plånbok delas aldrig med Apple eller lagras på Apples servrar och ID-handlingar lagras säkert inuti Secure Element i enheter som stöds. Secure Element innehåller särskilt utformade appletar som hanterar åtkomstnycklar på ett säkert sätt så att de inte kan extraheras eller läcka.

Innan nycklar får läggas till måste en användare vara inloggad på sitt iCloud-konto på en kompatibel iPhone och tvåfaktorsautentisering måste vara aktiverad för iCloud-kontot, med undantag för studentkort (som inte kräver tvåfaktorsautentisering för att aktiveras).

När en användare inleder tilläggsprocessen sker denna i liknande steg som när kredit- och bankkort läggs till, exempelvis länkning och tillhandahållande. Under en transaktion kommunicerar läsaren med Secure Element genom NFC-styrenheten (Near Field Communication) via en upprättad säker kanal.

Antalet enheter, inklusive iPhone och Apple Watch, där en nyckel kan läggas till definieras och styrs av respektive partner och kan variera mellan olika partners. Ett sådant tillvägagångssätt tillåter att enskilda partners har kontroll över det högsta antalet nycklar som läggs till för olika enhetstyper så att det passar deras specifika behov. För detta syfte tillhandahåller Apple partners med enhetstyp och anonymiserade enhetsidentifierare. Identifierare skiljer sig åt mellan olika partners av integritets- och säkerhetsskäl.

Partners får också användaridentifierare, som är anonymiserade och unika för varje partner, så att de säkert kan koppla nyckeln till användarens iCloud-konto under det inledande tillägget. Den här åtgärden skyddar nycklar från att läggas till av en annan användare ifall ett användarkonto som skapades med partnern har blivit äventyrat, till exempel under ett kontokapningsscenario.

Nycklar kan avaktiveras eller tas bort genom att:

- Fjärradera enheten med Hitta.
- Aktivera förlorat läge med Hitta.
- Ta emot ett MDM-fjärraderingskommando.
- Användaren tar bort alla kort från sina Apple-ID-kontosidor.
- Ta bort alla kort från iCloud.com.
- Ta bort alla kort från Plånbok.
- Ta bort kortet i utfärdarens app.

När en användare har iOS 15.4 eller senare och dubbelklickar på sidoknappen på en iPhone med Face ID, eller dubbelklickar på hemknappen på en iPhone med Touch ID, visas inte detaljer om dess kuponger och åtkomstnycklar förrän han eller hon autentiserar sig på enheten. Användaren måste autentisera med Face ID, Touch ID eller lösenkod innan specifik information om kuponger, exempelvis detaljer om en hotellbokning, visas i Plånbok.

# Åtkomstnyckeltyper

Det finns olika typer av åtkomst från Plånbok, exempelvis hotellnycklar, passerkort, studentkort, hemnycklar och bilnycklar.

## Hotell

Med hotellrumsnycklar i Plånbok är det enkelt att göra allt från incheckning till utcheckning kontaktlöst och samtidigt öka integriteten och säkerheten för gästerna jämfört med traditionella hotellnyckelkort i plast. Hotellgäster på medverkande anläggningar kan trycka för att låsa upp med rumsnycklar i Plånbok på sin kompatibla iPhone och Apple Watch Series 4 eller senare.

Funktionerna i Plånbok är särskilt utformade för att göra upplevelsen smidigare för kunden:

- Tillägg av en kupong i Plånbok från hotellets app inför vistelsen
- Kuponger för att inleda incheckning och rumstilldelning från Plånbok
- Nyckeluppdateringar vid förlängning eller ändring av pågående vistelser
- Stöd för flera rumsnycklar för en enda kupong i Plånbok
- Automatisk arkivering av utgångna nycklar i Plånbok

## Disney MagicMobile-kort

Användare kan lägga till ett Disney MagicMobile-kort i Plånbok på iPhone eller Apple Watch och använda det i entrén till medverkande Disney-temaparker. MagicMobile-kort kan användas i entrén till parken liksom vid en rad andra läsare i hela parkerna.

När användare vill lägga till ett Disney MagicMobile-kort måste tvåfaktorsautentisering vara aktiverat för deras iCloud-konto och användare måste ha biljetter eller bokningar till en medverkande temapark som är associerad med ett giltigt My Disney Experience-konto. Från My Disney Experience-appen på iPhone kan användaren välja ett eller flera kort att lägga till i Plånbok. Om användaren har en parkopplad Apple Watch läggs de valda korten automatiskt till på både användarens iPhone och dennas parkopplade Apple Watch. Expressläge slås på som förval för kort som läggs till på både iPhone- och Apple Watch-enheter. När expressläge slås på blir det också påslaget för alla MagicMobile-kort som för närvarande finns på enheten så att de enkelt kan användas.

Flera kort kan läggas till på en enda enhet så att användare kan hantera korten för alla medlemmar i sin grupp. Användare kan också välja att använda My Disney Experience-appen till att dela kort med andra användare. På det sättet kan mottagare lägga till de delade korten i Plånbok på sina enheter.

## Passerkort

Passerkort för medarbetare i partnerföretag kan läggas till i Plånbok på iPhone och Apple Watch så att medarbetare i hela världen får kontaktlös tillgång till sina arbetsplatser. När medarbetare ska lägga till ett passerkort måste de ha flerfaktorsautentisering aktiverad för kontot som används vid inloggning till den app som arbetsgivaren tillhandahåller.

Medarbetares passerkort drar nytta av Apples åtkomstfunktioner som gör det möjligt för användare att:

- Automatiskt lägga till ett passerkort på sin parkopplade Apple Watch via en pushfunktion som inte kräver att en partnerapp installeras
- Smidigt komma åt kontorsutrymmen med expressläge
- Få tillgång till arbetsplatsen även efter att iPhones batteri har tömts

## Studentkort

I iOS 12 och senare kan studenter, lärare och personal på campus som deltar lägga till sina student- och personalkort i Plånbok på iPhone- och Apple Watch-modeller som stöds så att de kan öppna dörrar och betala överallt där kortet gäller.

En användare lägger till sitt studentkort i Plånbok via en app som tillhandahålls av kortutfärdaren eller skolan som deltar. Den tekniska processen som används är samma som den som beskrivs i Lägga till kredit- eller bankkort från en kortutfärdares app. Utfärdande appar måste dessutom ha stöd för tvåfaktorsautentisering för de konton som bevakar tillgången till deras studentkort. Ett kort kan ställas in samtidigt på en användares iPhone och en parkopplad Apple Watch.

## Flerfamiljshem

Hyresgäster och personal i partnerbyggnader som stöds kan använda sin hemnyckel i Plånbok till att öppna porten, lägenhetsdörren och gemensamhetsutrymmen. Hemnyckeln kan läggas till och aktiveras från partnerns app. För partners som stöder friktionslöst tillägg kan fastighetsförvaltare skicka en länk till hyresgäster så att de kan inleda tillägget via den kanal de föredrar (t.ex. mejl eller SMS) så att hyresgästen bara behöver klicka på länken för att hämta nyckeln. Appklipp skapar också en säker och smidig upplevelse som gör det möjligt att lägga till en nyckel utan att installera en partners app. Mer information finns i Apple Support-artikeln Använda appklipp på iPhone.

En flerfamiljshemnyckel kan också användas i expressläge och delas säkert med familjemedlemmar och vänner. Mer information finns i Nyckeldelning.

## Hemnyckel

En hemnyckel i Plånbok kan användas med NFC-kompatibla dörrlås som stöds genom att helt enkelt trycka på en iPhone eller Apple Watch. Mer information om hur en användare kan ställa in och använda en hemnyckel finns i Apple Support-artikeln Låsa upp dörren med en hemnyckel på iPhone.

När en användare ställer in en hemnyckel får alla som är bosatta i hemmet automatiskt den nya hemnyckeln. Om ägaren till ett hem utöver detta vill dela en hemnyckel, eller ta bort en medlem i ett delat hem, kan ägaren använda appen Hem till att hantera inbjudningar och medlemmar. När en användare väljer att tacka ja till en inbjudan att ansluta till ett hem med en hemnyckel inleder detta tilläggningen av hemnyckeln i Plånbok på dennas enhet. Ifall en användare väljer att lämna ett hem, eller om hemmets ägare drar tillbaka användarens åtkomst, leder dessa åtgärder till att hemnyckeln tas bort från Plånbok.

### Bilnyckel

I iPhone-enheter och parkopplade Apple Watch-enheter som stöds går det att lagra bilnycklar digitalt i Plånbok. Bilnycklar visas som kuponger (skapade av Apple på uppdrag av biltillverkaren) i Plånbok och stöder hela livscykeln för Apple Pay-kort (förlorat läge i iCloud, fjärradering, lokal radering av kuponger och Radera allt innehåll och alla inst.). I likhet med vanliga Apple Pay-kort kan delade bilnycklar raderas från ägarens iPhone och Apple Watch samt från gränssnittet i bilen (HMI, Human Machine Interface).

Bilnycklar kan till exempel användas till att låsa upp och låsa fordonet, öppna och stänga bagageluckan, slå på och stänga av alarmet, starta motorn eller försätta fordonet i körläge. ”Standardtransaktionen” erbjuder gemensam autentisering och är obligatorisk för motorstart. Upplåsnings- och låsningstransaktioner kan använda den ”snabba transaktionen” om det krävs av prestandaorsaker.

Nycklar skapas genom att ansluta (eller parkoppla) en iPhone till en bil som har stöd för funktionen och som användaren äger. Alla nycklar skapas inuti Secure Element baserat på en ECC-OBKG (elliptic curve (NIST P-256) on-board key generation) och de privata nycklarna lämnar aldrig Secure Element. Kommunikation mellan enheter och bilen sker antingen via NFC eller en kombination av Bluetooth LE och Ultra Wideband (UWB). Nyckelhanteringen använder ett Apple-till-biltillverkarserver-API med gemensamt autentiserat TLS. När nyckeln har parkopplats med en iPhone kan också alla Apple Watch-enheter som är parkopplade med denna iPhone ta emot nyckeln. När en nyckel raderas antingen i fordonet eller på enheten kan den inte återskapas. Nycklar på enheter som försvinner eller blir stulna kan spärras och öppnas igen, men det krävs en ny parkoppling eller delning om de ska läggas till på en ny enhet.

Bilnycklar kan också användas i expressläge och delas säkert med familjemedlemmar och vänner. Mer information finns i Nyckeldelning. Mer information om säkerhet och integritet för digitala bilnycklar finns i Säkerhet för bilnycklar i iOS.

### Skoternyckel

I iOS 17 eller senare och i vissa länder eller regioner med partners som stöds kan användare få en skoternyckel från partnerappen direkt i Plånbok på en iPhone och parkopplad Apple Watch som stöds för följande ändamål:

- Tryck för att låsa eller låsa upp skotern
- Tryck för att låsa eller låsa upp skoterns bagageutrymme (om sådant finns)

En dedikerad applet i Secure Element hanterar kryptografiska ID-handlingar som är associerade till skoternyckeln på ett säkert sätt och tillåter säkra transaktioner med skotern.

På baksidan av kortet kan användare komma åt mer information om sin skoter, till exempel de sista fyra siffrorna i fordonsidentifieringsnumret och dess registreringsskylt. Sådan information kan betraktas som privat och är endast tillgänglig efter autentisering med biometri eller enhetens lösenkod.

Skoternyckeln kan också användas i expressläge och delas säkert med familjemedlemmar och vänner. Mer information finns i Nyckeldelning.

## Säkerhet för bilnycklar i iOS

Utvecklare kan lägga till stöd för säkra nyckellösa metoder att låsa upp och köra en bil i en iPhone som stöds och en parkopplad Apple Watch.

### Parkoppling utförd av ägaren

Ägaren måste bevisa att han eller hon äger fordonet (metoden är olika för olika biltillverkare) och kan starta parkopplingsprocessen i biltillverkarens app genom att använda en e-postlänk som biltillverkaren skickat eller via fordonets meny. Oavsett metod måste ägaren alltid ange ett hemligt engångslösenord för parkoppling på iPhone. Lösenordet används till att generera en säker parkopplingssignal med protokollet SPAKE2+ med NIST P-256-kurvan. När appen eller e-postlänken används överförs lösenordet automatiskt till iPhone medan det måste anges manuellt när parkopplingen startas från fordonet.

### Nyckeldelning

Ägarens parkopplade iPhone kan dela nycklar till behöriga familjemedlemmars och vänners iPhone-enheter (och deras parkopplade Apple Watch-enheter) genom att skicka enhetsspecifika inbjudningar via iMessage och IDS (Apple Identity Service). Alla delningskommandon skickas med IDS-funktionen och heltäckande kryptering. Ägarens parkopplade iPhone förhindrar att IDS-kanalen ändras under delningsprocessen för att skydda mot vidarebefordring av inbjudningen.

När inbjudan har accepterats skapar familjemedlemmens eller vännens iPhone en digital nyckel och skickar certifikatkedjan för tillverkning av nyckeln tillbaka till ägarens parkopplade iPhone för att verifiera att nyckeln skapades på en autentisk Apple-enhet. Ägarens parkopplade iPhone signerar den publika ECC-nyckeln för den andra familjemedlemmens eller vännens iPhone och skickar tillbaka signaturen till familjemedlemmens eller vännens iPhone. Signeringsåtgärden i ägarenheten kräver användarautentisering (Face ID, Touch ID eller lösenkod) och en säker bekräftelse av användarens avsikt enligt beskrivningen i Användningsområden för Face ID och Touch ID. Auktoriseringen begärs när inbjudan skickas. Den lagras i Secure Element och används när vännens enhet skickar tillbaka begäran om signering. Nyckelbehörigheter överförs till bilen antingen online via bilens OEM-server eller under den första användningen av den delade nyckeln i bilen.

### Nyckelradering

Det går att radera nycklar i nyckelhållarenheten via ägarenheten och i fordonet. Nycklar som raderas på nyckelhållarens iPhone försvinner omedelbart, även om nyckelhållaren använder nyckeln. Därför visas en skarp varning innan den raderas. Det kan vara möjligt att radera nycklar i bilen när som helst eller bara när bilen är uppkopplad.

I båda fallen rapporteras raderingen på nyckelhållarens enhet eller i fordonet till en KIS-server (Key Inventory Server) hos biltillverkaren som i sin tur registrerar utfärdade nycklar för ett fordon av försäkringsskäl.

Ägaren kan begära en radering från baksidan av ägarkupongen. Begäran skickas först till biltillverkaren så att nyckeln tas bort i fordonet. Villkoren för att ta bort nyckeln från fordonet fastställs av biltillverkaren. Biltillverkarens server skickar en begäran om fjärradering till nyckelhållarens enhet först när nyckeln tas bort i fordonet.

När en nyckel avslutas i en enhet skapar appleten som hanterar de digitala bilnycklarna en kryptografiskt signerad avslutningsattest. Den används som bevis på att nyckeln har raderats av biltillverkaren och till att ta bort nyckeln från KIS.

## Standardtransaktioner via NFC

För bilar som använder en NFC-nyckel upprättas en säker kanal mellan en läsare och iPhone genom att tillfälliga nyckelpar genereras på läsarens sida och iPhone-sidan. Med en nyckelavtalsmetod kan en delad hemlighet härledas på båda sidor och användas till att generera en delad symmetrisk nyckel med Diffie-Hellman, en nyckelhärledningsfunktion och signaturer från den långvariga nyckeln som skapats under parkoppling.

Den tillfälliga, offentliga nyckeln som genereras på fordonssidan signeras med läsarens långvariga, privata nyckel, vilket resulterar i en autentisering av läsaren av iPhone. Från iPhones håll är det här protokollet utformat för att förhindra att integritetskänsliga data avslöjas för en motspelare som avlyssnar kommunikationen.

Till sist använder iPhone den upprättade säkra kanalen till att kryptera sin identifierare för den offentliga nyckeln samt signaturen som beräknats efter en läsares datahärledda kontrolltal och andra ytterligare appspecifika data. Läsarens verifiering av iPhone-signaturen gör det möjligt för läsaren att autentisera enheten.

## Snabba transaktioner

iPhone genererar ett kryptogram baserat på en hemlighet som tidigare delats under en standardtransaktion. Detta kryptogram gör det möjligt för fordonet att snabbt autentisera enheten i prestandakänsliga situationer. Ett alternativ till detta är att en säker kanal upprättas mellan fordonet och enheten genom att sessionsnycklar härleds från en hemlighet som har delats under en standardtransaktion och ett nytt par med tillfälliga nycklar. Möjligheten för fordonet att upprätta den säkra kanalen autentiserar fordonet för iPhone.

## Standardtransaktioner via BLE/UWB

För bilar som använder en UWB-nyckel upprättas en Bluetooth LE-session mellan bilen och iPhone. I likhet med NFC-transaktionen härleds en delad hemlighet på båda sidorna och används vid upprättandet av en säker session. Den här sessionen används därefter till att härleda och samtycka till en URSK (UWB Ranging Secret Key) Denna URSK skickas sedan till UWB-radiomottagare i användarens enhet och i bilen för att aktivera en noggrann platsbestämning för användarens enhet i förhållande till en viss plats nära eller inuti bilen. Bilen använder sedan enhetens plats till att avgöra ifall den tillåter att bilen blir upplåst eller startar. URSK:er har en fördefinierad TTL. För att undvika störningar av intervallbestämning när en TTL löper ut kan URSK:er förhärledas i enhetens SE och bilens HSM/SE när BLE är anslutet men säker intervallbestämning inte är aktiv. Det här undviker behovet för en standardtransaktion att härleda en ny URSK i en tidskritisk situation. En förhärledd URSK kan överföras väldigt snabbt till UWB-radiomottagare i bilen och enheten för att undvika störningar av UWB-intervallbestämningen.

## Integritet

Biltillverkarens nyckellagerserver (key inventory server, KIS) lagrar inte enhetens ID, SEID eller Apple-ID. Den lagrar endast en muterbar identifierare – t.ex. instansens CA-identifierare. Identifieraren är inte bunden till privata data på enheten eller servern, och den raderas när användaren rensar sin enhet helt (genom att använda Radera allt innehåll och inst.).

# Lägga till rese- och eMoney-kort i Plånbok

På flera platser i världen kan användare lägga till rese- och eMoney-kort som stöds i Plånbok på kompatibla iPhone- och Apple Watch-modeller. Beroende på kollektivtrafikföretaget kan det här göras genom att överföra saldot eller pendlingsbiljetten (eller båda) från ett fysiskt kort till dess digitala representation i Plånbok, eller genom att lägga till ett nytt rese- eller eMoney-kort från Plånbok eller kortutfärdarens app. När resekort läggs till i Plånbok kan användare åka kollektivt genom att helt enkelt hålla iPhone eller Apple Watch nära resekortläsaren. Vissa resekort kan även användas till att betala med.

## Så här fungerar rese- och eMoney-kort

Tillagda rese- och eMoney-kort associeras med en användares iCloud-konto. Om användaren lägger till fler än ett kort i Plånbok kanske Apple eller kortutfärdaren kan länka användarens personliga uppgifter och den associerade kontoinformationen mellan korten. Rese- och eMoney-kort och transaktioner skyddas av en uppsättning hierarkiska kryptografiska nycklar.

När saldot från ett fysiskt kort överförs till Plånbok måste användaren ange information som är specifik för det aktuella kortet. Användare kan även behöva ange personuppgifter som bevis för kortinnehavet. När biljetter överförs från iPhone till Apple Watch måste båda enheterna vara uppkopplade.

Saldot kan fyllas på med pengar från kredit-, bank- och förbetalda kort via Plånbok eller från rese- eller eMoney-kortutfärdarens app. Om du vill veta mer om säkerheten vid påfyllning av saldot när Apple Pay används läser du Betala med kort i appar. Om du vill veta hur resekortet tillhandahålls inuti kortutfärdarens app läser du Lägga till kredit- eller bankkort från en kortutfärdares app.

Om det finns stöd för tillägg från ett fysisk kort har utfärdaren av rese- eller eMoney-kortet de kryptografiska nycklar som behövs för att autentisera det fysiska kortet och verifiera användarens angivna data. När alla data har verifierats kan systemet skapa ett enhetskontonummer för Secure Element och aktivera den tillagda biljetten i Plånbok tillsammans med det överförda saldot. Vissa fysiska kort avaktiveras när överföringsprocessen från det fysiska kortet är klar.

Om saldot för resekort lagras på enheten krypteras saldot på resekortet och lagras i en utsedd applet i Secure Element vid slutet av båda processerna. Kollektivtrafikföretaget har nycklarna som krävs för att utföra kryptografiska åtgärder gällande kortdata för saldotransaktioner.

Som förval drar användare med resekort nytta av den smidiga expressreseupplevelsen som innebär att de kan betala och åka utan att använda Touch ID, Face ID eller en lösenkod. Information som senast besökta stationer, transaktionshistorik och ytterligare biljetter kan kontrolleras via en kontaktlös kortläsare i närheten med Expressläge aktiverat. Användare kan slå på auktoriseringskrav för Face ID, Touch ID eller lösenkod i Plånbok och Apple Pay-inställningarna genom att avaktivera Expressresor. Expressläge stöds inte för eMoney-kort.

I likhet med andra Apple Pay-kort kan användare spärra eller ta bort eMoney-kort genom att:

- Fjärradera enheten med Hitta.
- Aktivera förlorat läge med Hitta.
- Ange ett MDM-fjärraderingskommando
- Användaren tar bort alla kort från sina Apple-ID-kontosidor.
- Ta bort alla kort från iCloud.com.
- Ta bort alla kort från Plånbok.
- Ta bort kortet i utfärdarens app.

Apple Pay-servrar meddelar kortföretaget om att spärra eller avaktivera dessa kort. Om en användare tar bort ett rese- eller eMoney-kort kan saldot återvinnas genom att användaren lägger tillbaka kortet på en enhet som är inloggad med samma Apple-ID. Om en enhet är nedkopplad, avstängd eller oanvändbar kanske det inte går att återvinna saldot.

## Lägga till rese- och eMoney-kort på en familjemedlems Apple Watch

I iOS 15 eller senare och watchOS 8 eller senare kan samordnaren i en iCloud-familj lägga till rese- och eMoney-kort på familjemedlemmars Apple Watch-enheter genom iPhones Apple Watch-app. När ett av de här korten läggs till på en familjemedlems Apple Watch måste klockan finnas i närheten och vara ansluten till samordnarens iPhone via Wi-Fi eller Bluetooth. Familjemedlemmar måste ha autentisering med tvåfaktorsautentisering aktiverad för sitt Apple-ID för att det här ska fungera.

Familjemedlemmar kan skicka en förfrågan om att lägga till pengar på ett rese- eller eMoney-kort från sin Apple Watch via iMessage. Innehåller i meddelandet skyddas av heltäckande kryptering i enlighet med beskrivningen i iMessage-säkerhet i översikt. Pengar kan läggas till på ett kort på en familjemedlems Apple Watch på distans med en Wi-Fi- eller mobilanslutning. Enheterna behöver inte vara nära varandra.

*Obs!* Den här funktionen är kanske inte tillgänglig i alla länder eller regioner.

## Kredit- och bankkort

I vissa städer accepterar kollektivtrafikläsare EMV-kort (smarta kort) för betalning av resor. När användaren håller ett kredit- eller bankkort av typen EMV vid en sådan läsare krävs användarautentisering på det sätt som beskrivs under Betala med kredit- eller bankkort i butiker.

I iOS 12.3 eller senare kan vissa befintliga EMV-kreditkort eller bankkort i Plånbok aktiveras för expressresor. Med expressresor kan användare betala för en resa med kollektivtrafikföretag som stöds utan att det krävs Face ID, Touch ID eller en lösenkod. När en användare lägger till ett EMV-kreditkort eller -bankkort aktiveras det första kortet som lagts till i Plånbok för expressresor. Användaren kan trycka på merknappen på framsidan av kortet i Plånbok och avaktivera Expressresor för det kortet genom att välja Inga vid Inställningar för expressresor. Användaren kan också välja ett annat kredit- eller bankkort som expressresekort i Plånbok. Face ID, Touch ID eller lösenkod krävs för att återaktivera eller välja ett annat kort för expressresor.

Apple Card och Apple Cash fungerar för expressresor.

# ID:n i Plånbok

## ID:n i Plånbok

På iPhone 8 eller senare med iOS 15.4 eller senare och Apple Watch Series 4 eller senare med watchOS 8.4 eller senare kan användare lägga till sitt myndighetsutfärdade ID-kort eller körkort i Plånbok och sedan trycka på sin iPhone eller Apple Watch för att smidigt och säkert visa det på medverkande platser.

*Obs!* Den här funktionen är endast tillgänglig i medverkande delstater i USA.

ID:n i Plånbok använder säkerhetsfunktioner som är inbyggda i maskinvaran och programvaran i användares enheter som hjälp att skydda deras identitet och hålla deras personliga information säker.

### Lägga till ett körkort eller myndighetsutfärdat ID-kort i Plånbok

På iPhone kan användare helt enkelt trycka på lägg till-knappen (+) överst på skärmen i Plånbok för att börja lägga till sitt körkort eller ID-kort. Om användare har en parkopplad Apple Watch under inställningen blir de uppmanade att också lägga till körkortet eller ID-kortet i Plånbok på Apple Watch.

Användare blir först ombedda att skanna framsidan och baksidan av det fysiska körkortet eller ID-kortet med sin iPhone. iPhone utvärderar kvaliteten och typen av bilder för att säkerställa att bilderna uppfyller kraven hos den utfärdande myndigheten. ID-bilderna krypteras med den utfärdande myndighetens nyckel på enheten och skickas sedan till den utfärdande myndigheten.

Därefter blir användaren ombedd att utföra en serie ansikts- och huvudrörelser. Dessa rörelser utvärderas av användarens enhet och av Apple för att minska risken att någon annan använder en bild, video eller mask till att försöka lägga till någon annans ID i Plånbok. Resultatet från analysen av rörelserna skickas sedan till den utfärdande myndigheten, men inte själva videon med rörelser.

För att säkerställa att personen som lägger till ett ID-kort i Plånbok är samma person som ID-kortet tillhör blir användare uppmanade att ta en selfie. Innan användarens bild skickas in till den utfärdande myndigheten jämför Apples servrar och användarens enhet bilden med utseendet på personen som utförde serien med ansikts- och huvudrörelser som ett sätt att säkerställa att bilden som skickas in är av en levande person med samma utseende som den på ID-kortet. När jämförelsen är klar krypteras bilden på enheten och skickas sedan till den utfärdande myndigheten där bilden jämförs med den bild som är arkiverad för ID-kortet.

Slutligen blir användare uppmanade att autentisera med Face ID eller Touch ID. Användarens enhet kopplar dessa matchade biometriska data från Face ID eller Touch ID till ID-kortet för att se till att endast personen som lade till ID-kortet på iPhone kan visa det. Annan registrerad biometrisk information kan inte användas till att auktorisera visning av ID-kortet. Detta sker endast på enheten och skickas inte till den utfärdande myndigheten.

Den utfärdande myndigheten får information som krävs för att skapa det digitala ID-kortet. Detta omfattar bilder av fram- och baksidan på användarens ID-kort, data som har lästs från PDF417-streckkoden samt den selfie som användaren tog som en del av ID-verifieringsprocessen. Den utfärdande delstaten får även ett ensiffrigt värde, som används till att förhindra bedrägerier, som baseras på användarens enhetsanvändningsmönster, inställningsdata och information om dennas personliga Apple-ID. Slutligen är det upp till den utfärdande delstaten att godkänna eller neka att ID-kortet läggs till i Plånbok.

När den utfärdande delstatsmyndigheten har godkänt att ID-kortet eller körkortet läggs till i Plånbok genereras ett nyckelpar i Secure Element av iPhone som förankrar användarens ID-kort till den specifika enheten. Om det läggs till på Apple Watch genereras ett nyckelpar i Secure Element på Apple Watch.

När ID-kortet finns på iPhone lagras informationen som återspeglas på användarens ID-kort i Plånbok i ett krypterat format som skyddas av Secure Enclave.

### Använda ett körkort eller myndighetsutfärdat ID-kort i Plånbok med en ID-läsare

För att kunna använda sitt ID-kort i Plånbok måste användare autentisera med Face ID eller Touch ID på den enhet som är kopplad till ID-kortet i Plånbok innan iPhone visar informationen för ID-läsaren.

För att kunna använda sitt ID-kort i Plånbok på Apple Watch måste användare låsa upp sin iPhone med det kopplade Face ID-utseendet eller Touch ID-fingeravtrycket varje gång de tar på sig sin Apple Watch. Sedan kan de använda sitt ID-kort i Plånbok utan att autentisera tills de tar av sig Apple Watch igen. Den här funktionen drar nytta av funktionerna för automatisk upplåsning som beskrivs i Systemsäkerhet för watchOS.

När användare håller sin iPhone eller Apple Watch i närheten av ID-läsaren, eller när de delar sitt ID inuti en app, ser de ett meddelande på enheten som visar vilken specifik information som efterfrågas, av vem och ifall den avser att lagra den. Efter auktorisering med det Face ID eller Touch ID som är kopplat överförs den efterfrågade ID-informationen från enheten.

**Viktigt:** Användare behöver inte låsa upp, visa eller lämna över sin enhet för att kunna visa sitt ID.

Ifall användare har aktiverat en hjälpmedelsfunktion som röststyrning, reglagestyrning eller AssistiveTouch istället för Face ID eller Touch ID kan de använda sin lösenkod till att komma åt och visa sin information.

Överföringen av ID-data till ID-läsaren följer standarden ISO/IEC 18013-5, vilken innehåller flera säkerhetsmekanismer som kan upptäcka, förhindra och begränsa säkerhetsrisker. Dessa utgörs av ID-dataintegritet och bedrägeriskydd, enhetskoppling, informerat samtycke och konfidentialitet för användardata via radiolänkar.

### Använda ett körkort eller myndighetsutfärdat ID-kort i Plånbok med iOS-appar

Användare kan också dela information om sitt körkort eller myndighetsutfärdade ID-kort i Plånbok med iOS-appar. När en användare delar sitt ID med en app hämtar och validerar Plånbok ett krypteringscertifikat som är registrerat hos apputvecklaren.

Detta certifikat används till att kryptera informationen som användaren har samtyckt till att dela. Informationen krypteras av Plånbok med HPKE och blir aldrig tillgänglig för Apple. Plånbok skickar då och då förfrågningar till Apples servrar för att verifiera att ID:t fortfarande är giltigt. Om en kontroll inte har utförts nyligen kan detta inträffa när användaren delar sitt ID med en app.

## Säkerhet för ID:n i Plånbok

Följande funktioner bidrar till att öka säkerheten när ID-kort används i Plånbok.

### ID-dataintegritet och bedrägeriskydd

ID:n i Plånbok använder en signatur från utfärdaren för att tillåta alla läsare som överensstämmer med ISO/IEC 18013-5 att verifiera en användares ID i Plånbok. Dessutom skyddas alla dataelement i ID-kort i Plånbok individuellt mot bedrägerier. Detta tillåter att ID-läsaren begär en specifik delmängd av dataelementen som visas på ID:t i Plånbok och att ID:t i Plånbok svarar med samma delmängd, vilket innebär att endast begärda data delas och att användarens integritet respekteras i största möjligaste mån.

### Enhetskoppling

ID:n som autentiseras i Plånbok använder en enhetssignatur till att skydda mot kloning av ett ID och replay av en ID-presentation. Plånbok lagrar den privata nyckeln för ID-autentisering i iPhones Secure Element så att ID-kortet kopplas till samma enhet som den utfärdande myndigheten skapade ID-kortet för.

### Informerat samtycke

ID:n i Plånbok kan använda autentisering för att identifiera läsaren i enlighet med protokollet som definieras i standarden ISO/IEC 18013-5. Om läsaren har ett eget certifikat som är betrott av Plånbok visas en symbol under presentationen som försäkrar användaren om att den interagerar med den avsedda parten.

### Konfidentialitet för användardata via radiolänkar

Sessionskryptering säkerställer att all personligt identifierbar information (PII) som utväxlas mellan ID-kortet i Plånbok och ID-läsaren är krypterad. Krypteringen utförs av applagret. Sessionskrypteringens säkerhet är därför inte beroende av den säkerhet som tillhandahålls av överföringslagret (till exempel NFC, Bluetooth och Wi-Fi).

### ID:n i Plånbok håller användarnas information privat

ID:n i Plånbok följer processen ”device retrieval” som beskrivs i ISO/IEC 18013-5. Enhetshämtning undanröjer behovet att göra serveranrop under visning och skyddar därmed användare från att spåras av Apple och utfärdaren.

### Säkerhet för ID-verifierare

I iOS 17 eller senare kan företag och organisationer i USA använda iPhone till att smidigt och säkert läsa mobila ID:n som användare visar, förutsatt att de överensstämmer med ISO 18013-5, utan någon extern maskinvara. ID-verifierare kan användas på två olika sätt beroende på hur verifieringen ska ske:

- *ID Verifier Display Only:* Det här gör det möjligt för ett iOS-gränssnitt att visa data som namn, ålder, ID-bild och ålder över *N* för fall där bara en visuell bekräftelse krävs. Tjänsten tillåter inte insamling av *personligt identifierbar information* (PII) som kan kopplas tillbaka till den som visar ID:t.

- *ID Verifier Data Transfer:* Det här gör det möjligt för appar att efterfråga ytterligare dataelement, som födelsedatum och adress, i syfte att uppfylla lagstadgade verifieringskrav. Tillgången till API:t ID Verifier Data Transfer styrs med hjälp av behörigheter och appar måste överensstämma med krav på hur data får användas. En app måste till exempel uppvisa ett lagenligt krav för att efterfråga ID-data. Appar måste också upprätthålla en integritetspolicy som detaljerat beskriver hur begärda ID-data bearbetas, lagras eller används på andra sätt.

### Läsa ett mobilt ID

ID-verifierare följer protokollet som definieras i standarden ISO/IEC 18013-5. När en app som använder API:t ID Verifier begär att få läsa ett mobilt ID visas ett blad (som styrs av iOS) där innehavaren av det mobila ID:t uppmanas att hålla sin enhet nära ID-läsaren. Denna inledande NFC-anslutning (enligt definitionen i standarden ISO/IEC 18013-5 kan en QR-kod användas till att inleda en Bluetooth-överlämningsprocess istället för NFC) upprättar en säker Bluetooth Low Energy (BLE)-anslutning mellan båda enheterna. Vid den tidpunkten kan innehavaren av det mobila ID:t granska informationen som efterfrågas på sin enhet. När innehavaren av det mobila ID:t ger sitt godkännande överförs efterfrågade ID-data till läsarenheten. Appar som använder API:t ID Verifier Data Transfer får svarsdata för bearbetning medan appar som använder API:t ID Verifier Display Only direkt ser data som visas av iOS.

Standarden ISO/IEC 18013-5 innehåller flera säkerhetsmekanismer som kan upptäcka, förhindra och begränsa säkerhetsrisker. Bland dessa utför ID-verifierare både validering av utfärdarsignatur och enhetssignatur. Dessutom stöder ID-verifierare läsarautentisering enligt protokollet som definieras i standarden ISO/IEC 18013-5. Appar kan välja att visa en symbol och ett namn som en försäkran om att ID-innehavaren interagerar med den avsedda parten genom att använda läsarens certifikat.

### Validering av utfärdare och enhet

Som ett skydd mot förfalskning validerar ID-verifierare signaturen för Mobile Security Object av den betrodda utfärdaren av det mobila ID:t. ID Verifier Data Transfer tillhandahåller också ett API som gör att appar kan utföra sin egen signaturvalidering, istället för iOS, om så önskas. Som en försäkran till företaget eller organisationen om att det mobila ID:t inte har kopierats från en enhet till en annan validerar ID-verifierare signaturen via sessionsdata.

### Läsarautentisering

Vid tidpunkten för ID-presentation signeras ID-verifierares läsarförfrågan av den privata nyckel som är associerad till läsarautentiseringscertifikatet som är kopplat till Apples rotcertifikatutfärdare (CA) som innehåller relevanta anpassade x509-tillägg för att indikera till innehavaren om företaget avser att lagra data. Om en app vill visa namn och symbol för ID-innehavaren måste appadministratören registrera den i Apple Business Register och tillhandahålla korrekt varumärkesinformation. När den skickade informationen har verifierats vid tidpunkten för transaktionen tillhandahåller läsarautentiseringscertifikatet information om entiteten från Apple Register till ID-innehavaren via läsarautentiseringscertifikatet.

# iMessage

## iMessage-säkerhet i översikt

Apples iMessage är en meddelandetjänst för iPhone- och iPad-enheter, Apple Watch och Mac-datorer. iMessage stöder text och bilagor som bilder, kontakter, platser, länkar och bilagor direkt i ett meddelande, t.ex. en tummen upp-symbol. Meddelanden visas på alla användarens registrerade enheter så att konversationer kan fortsätta från en enhet till en annan. iMessage använder APNs (Apples tjänst för pushnotiser). Apple loggar inte innehållet i meddelanden eller bilagor, och de skyddas av heltäckande kryptering så att ingen utom sändaren och mottagaren kan komma åt dem. Apple kan inte avkryptera informationen.

När en användare aktiverar iMessage på en enhet så genererar enheten kryptering och signerar nyckelpar för användning med tjänsten. För kryptering finns en 1280-bitars RSA-krypteringsnyckel och även en 256-bitars EC-krypteringsnyckel på NIST P-256-kurvan. För signaturer används 256-bitars signeringsnycklar av typen ECDSA (Elliptic Curve Digital Signature Algorithm). De privata nycklarna sparas i enhetens nyckelring och är endast tillgängliga efter den första upplåsningen. De privata nycklarna skickas till Apples ID-tjänst (IDS). Där kopplas de till användarens telefonnummer eller e-postadress tillsammans med enhetens APNs-adress.

När användarna aktiverar ytterligare enheter för användning med iMessage läggs deras publika nycklar för kryptering och signering, APNs-adresser och kopplade telefonnummer till i katalogtjänsten. Användarna kan också lägga till fler e-postadresser som verifieras genom att skicka en bekräftelselänk. Telefonnummer verifieras av operatörsnätet och SIM-kortet. I en del operatörsnät måste SMS användas (en bekräftelsedialogruta visas för användaren om SMS:et inte är kostnadsfritt). Verifiering av telefonnumret kan krävas för flera systemtjänster utöver iMessage, exempelvis FaceTime och iCloud. Alla användarens registrerade enheter visar ett varningsmeddelande när en ny enhet, ett nytt telefonnummer eller en ny e-postadress läggs till.

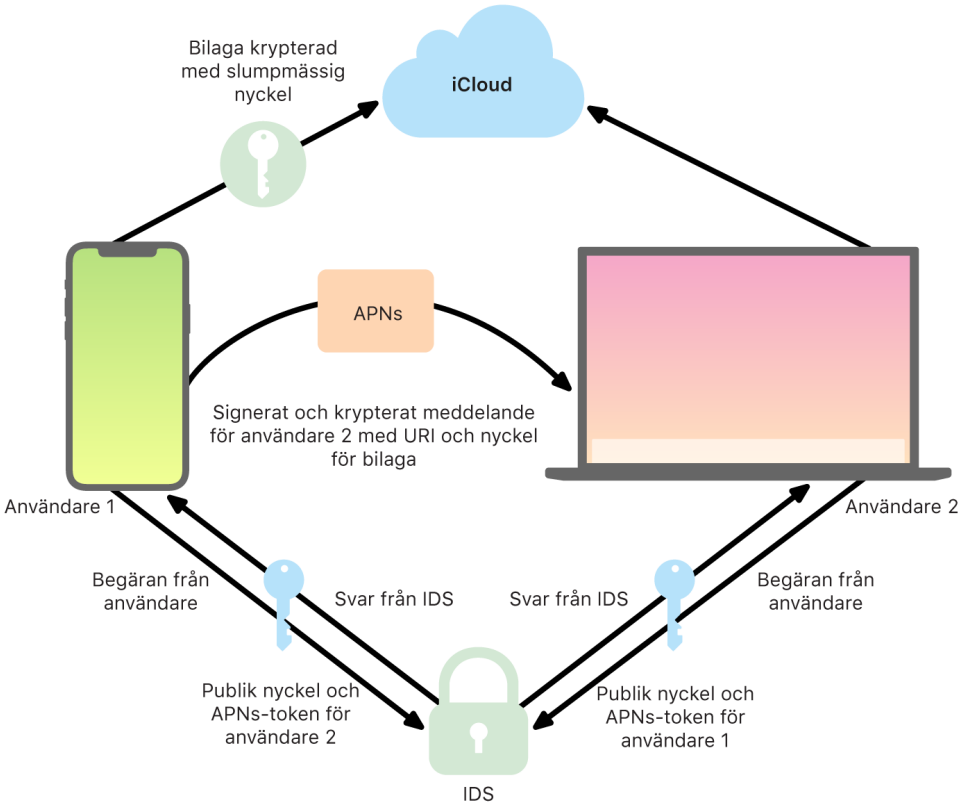
## Hur iMessage skickar och tar emot meddelanden på ett säkert sätt

Användarna startar en ny iMessage-konversation genom att ange en adress eller ett namn. Om de skriver in ett telefonnummer eller en e-postadress kontaktar enheten IDS (Apple Identity Service) och hämtar de publika nycklarna och APNs-adresserna för alla enheter som är kopplade till mottagaren. Om användaren skriver in ett namn använder enheten först appen Kontakter till att samla in telefonnummer och e-postadresser som är kopplade till det namnet. Sedan hämtar enheten de publika nycklarna och APNs-adresserna från IDS.

Användarens utgående meddelande krypteras enskilt för var och en av mottagarens enheter. De publika krypteringsnycklarna och signeringsnycklarna för de mottagande enheterna hämtas från IDS. För varje mottagande enhet genererar den avsändande enheten ett slumpmässigt 88-bitarsvärde och använder det som en HMAC-SHA256-nyckel för att konstruera ett 40-bitarsvärde härlett från avsändarens och mottagarens publika nyckel och texten. Hoplänkningen av 88-bitars- och 40-bitarsvärdet ger en 128-bitarsnyckel som krypterar meddelandet med AES i CTR-läge (Counter). 40-bitarsvärdet används av mottagarsidan till att verifiera integriteten för den avkrypterade texten. Denna AES-nyckel per meddelande krypteras med RSA-OAEP till den publika nyckeln för den mottagande enheten. Kombinationen av den krypterade meddelandetexten och den krypterade meddelandenyckeln bearbetas med hashfunktionen SHA-1, och hashvärdet signeras med ECDSA (Elliptic Curve Digital Signature Algorithm) med hjälp av den avsändande enhetens privata signeringsnyckel. I iOS 13 eller senare och iPadOS 13.1 eller senare kan enheter använda ECIES-kryptering (Elliptic Curve Integrated Encryption Scheme) istället för RSA-kryptering.

De resulterande meddelandena, ett för varje mottagande enhet, består av den krypterade meddelandetexten, den krypterade meddelandenyckeln och avsändarens digitala signatur. Dessa skickas sedan till APNs för leverans. Metadata som tidsstämpel och APNs-routinginformation krypteras inte. Kommunikation med APNs krypteras med en så kallad FS-TLS-kanal.

APNs kan endast vidarebefordra meddelanden på upp till 4 eller 16 KB beroende på iOS- eller iPadOS-version. Om meddelandet är för långt, eller om en bild eller annan bilaga ingår, krypteras bilagan med hjälp av AES i CTR-läge med en slumpgenererad 256-bitarsnyckel och överförs till iCloud. Bilagans AES-nyckel, dess URI (Uniform Resource Identifier) och ett SHA-1-hashvärde för dess krypterade form skickas sedan till mottagaren som innehållet i ett iMessage. Deras konfidentialitet och integritet skyddas genom normal iMessage-kryptering (se diagrammet nedan).




När det gäller gruppkonversationer upprepas processen för varje mottagare och deras enheter.

På den mottagande sidan får varje enhet en kopia av meddelandet från APNs och hämtar bilagan från iCloud när det behövs. Avsändarens inkommande telefonnummer eller e-postadress matchas mot mottagarens kontakter så att ett namn kan visas om möjligt.

I likhet med alla pushnotiser raderas meddelandet från APNs när det har levererats. Men till skillnad från andra pushnotiser ställs iMessage-meddelanden i kö för leverans till enheter som är offline. Meddelanden lagras på Apple-servrar i upp till 30 dagar.

## Säker namn- och bilddelning i iMessage

Med funktionen för namn- och bilddelning i iMessage kan en användare dela ett namn och en bild via iMessage. Användaren kan välja informationen från Mitt kort eller anpassa namnet och lägga till valfri bild. Funktionen för namn- och bilddelning i iMessage använder ett tvåstegssystem för att distribuera namnet och bilden.

Informationen delas upp i fält. Varje fält krypteras och autentiseras separat och autentiseras även tillsammans med den process som beskrivs nedan. Det finns tre fält:

- Namn
- Bild
- Bildfilnamn

Ett av de första stegen för att skapa data är att generera en slumpmässig 128-bitars postnyckel på enheten. Den postnyckeln härleds sedan med HKDF-HMAC-SHA256 för att skapa tre delnycklar: Nyckel 1:Nyckel 2:Nyckel 3 = HKDF(postnyckel, ”smeknamn”). För varje fält genereras ett slumpmässigt 96-bitars IV (Initialization Vector) och alla data krypteras med AES-CTR och Nyckel 1. Därefter beräknas en MAC (message authentication code) med HMAC-SHA256 genom användning av Nyckel 2 och täcker fältnamnet, fält-IV och fält-kodtext. Slutligen länkas uppsättningen av enskilda fält-MAC-värden och deras MAC beräknas med HMAC-SHA256 genom användning av Nyckel 3. Det aktuella 256-bitars MAC-värdet lagras tillsammans med de data som har krypterats. De första 128 bitarna av denna MAC används som RecordID.

Den här krypterade posten lagras sedan i den publika CloudKit-databasen under RecordID. Den här posten ändras aldrig, och när användaren väljer att ändra sitt namn och sin bild genereras en ny krypterad post varje gång. När användare 1 väljer att dela sitt namn och sin bild med användare 2 skickas postnyckeln tillsammans med det RecordID som finns i användarens iMessage-nyttolast som är krypterad.

När användare 2:s enhet tar emot den aktuella iMessage-nyttolasten ser den att nyttolasten innehåller ett RecordID med smeknamn och bild samt en nyckel. Användare 2:s enhet anropar sedan den publika CloudKit-databasen för att hämta det krypterade namnet och den krypterade bilden från RecordID och skickar informationen via iMessage.

När meddelandet har hämtats avkrypterar användare 2:s enhet nyttolasten och verifierar signaturen med hjälp av RecordID. Om verifieringen godkänns får användare 2 se namnet och bilden och kan välja att lägga till informationen bland sina kontakter eller använda den för Meddelanden.

# Säkra Apple Messages for Business

Apple Messages for Business är en meddelandetjänst som gör det möjligt för användare att kommunicera med företag i appen Meddelanden. Med Apple Messages for Business är det användaren som styr konversationen. Användare kan också radera konversationen och blockera företaget från att skicka meddelanden i framtiden. Av integritetsskäl får företaget inte användarens telefonnummer, e-postadress eller iCloud-kontoinformation. Istället genereras en anpassad unik identifierare som kallas ett *anonymt ID* av Apple Identity Service (IDS) och delas med företaget. Det anonyma ID:t är unikt för relationen mellan användarens Apple-ID och företagets företags-ID. Användarna har olika anonyma ID:n för varje företag de har kontakt med via Apple Messages for Business. Användaren bestämmer om och när personligt identifierande information ska delas med företaget och Apple Messages for Business lagrar aldrig konversationshistoriken.

Apple Messages for Business har stöd för hanterade Apple-ID:n från Apple Business Manager och avgör om de är aktiverade för iMessage och FaceTime i Apple School Manager.

Meddelanden som skickas till företaget krypteras mellan användarens enhet och Apples meddelandeservrar och använder samma säkerhet och Apple-meddelandeservrar som iMessage. Apples meddelandeservrar avkrypterar dessa meddelanden i RAM och vidarebefordrar dem till företaget via en krypterad länk med TLS 1.2. Meddelanden lagras aldrig i okrypterad form under överföringen genom Apple Messages for Business-tjänsten. Företags svar skickas också med TLS 1.2 till Apples meddelandeservrar. Där krypteras de med vardera mottagarenhets unika, offentliga nycklar.

Om användarens enhet är ansluten levereras meddelandet direkt och cachelagras inte på Apples meddelandeservrar. Om en användares enhet är nedkopplad cachelagras det krypterade meddelandet i upp till 30 dagar för att användaren ska kunna ta emot meddelandet när enheten ansluts igen. Så fort enheten ansluts igen levereras meddelandet och raderas från cachen. Efter 30 dagar går ett cachelagrat meddelande som inte levererats ut och raderas permanent.

# FaceTime-säkerhet

FaceTime är Apples video- och röstsamtalstjänst. I likhet med iMessage använder FaceTime APNs (Apples pushnotistjänst) till att upprätta en första anslutning till användarens registrerade enheter. Ljud- och videoinnehållet i FaceTime-samtal skyddas av heltäckande kryptering så att ingen utom sändaren och mottagaren kan komma åt det. Apple kan inte avkryptera informationen.

Den inledande FaceTime-anslutningen sker via Apples serverinfrastruktur som vidarebefordrar datapaket mellan användarnas registrerade enheter. Med hjälp av APNs och STUN-meddelanden (Session Traversal Utilities for NAT) via reläanslutningen verifierar enheterna sina identitetscertifikat och upprättar en delad hemlighet för varje session. Den delade hemligheten användas till att härleda sessionsnycklar för mediekanaler som strömmas med hjälp av SRTP-protokollet (Secure Real-time Transport Protocol). SRTP-paket krypteras med AES256 i Counter Mode och autentiseras med HMAC-SHA1. Efter den inledande anslutningen och säkerhetsinställningen använder FaceTime STUN och ICE (Internet Connectivity Establishment) till att om möjligt upprätta en P2P-anslutning mellan enheter.

FaceTime-gruppsamtal utökar FaceTime så att det stöder upp till 33 samtidiga deltagare. I likhet med klassiska tvåvägssamtal via FaceTime är gruppsamtalen heltäckande krypterade mellan de inbjudna deltagarnas enheter. FaceTime-grupper återanvänder en stor del av infrastrukturen och utformningen hos tvåvägs-FaceTime, men gruppsamtalen har även en ny mekanism för nyckelupprättning byggd ovanpå den autentisering som tillhandahålls av IDS (Apple Identity Service). Det här protokollet tillhandahåller forward secrecy, vilket innebär att en enhet vars säkerhet äventyras inte kommer att läcka innehåll från tidigare samtal. Sessionsnycklarna paketeras via AES-SIV och distribueras bland deltagarna genom att använda en ECIES (Elliptic Curve Integrated Encryption Scheme)-konstruktion med tillfälliga P-256 ECDH-nycklar.

När ett nytt telefonnummer eller en ny e-postadress läggs till i ett pågående FaceTime-gruppsamtal upprättar aktiva enheter nya medienycklar och delar aldrig de tidigare använda nycklarna med de nyligen inbjudna enheterna.

# Hitta

## Hitta och säkerhet

Hitta-appen för Apple-enheter bygger på en grund av avancerad kryptografi med publika nycklar.

### Översikt

Hitta är en kombination av Hitta min iPhone och Hitta mina vänner i iOS, iPadOS och macOS. Hitta kan hjälpa användare att hitta en saknad enhet, till och med en Mac som inte är ansluten till internet. En uppkopplad enhet kan helt enkelt rapportera sin plats till användaren via iCloud. Hitta fungerar i frånkopplat läge genom att skicka ut Bluetooth-signaler med kort räckvidd från den saknade enheten som kan upptäckas av andra Apple-enheter i närheten. Enheterna i närheten kan sedan förmedla platsen för den saknade enheten till iCloud så att användare kan hitta den i appen Hitta samtidigt som integriteten och säkerheten tryggas för alla inblandade användare. Hitta fungerar till och med för en Mac som är frånkopplad och i viloläge.

Genom att använda Bluetooth och de många hundra miljoner iOS-, iPadOS- och macOS-enheterna i aktiv användning över hela världen kan en användare hitta en saknad enhet, även om den inte kan ansluta till vare sig ett Wi-Fi-nätverk eller mobilnätverk. Alla iOS-, iPadOS- och macOS-enheter med Hitta i nedkopplat läge aktiverat i inställningarna för Hitta kan fungera som en ”upptäckarenhet”. Det här innebär att enheten kan upptäcka närvaron av en annan saknad och nedkopplad enhet via Bluetooth, och sedan använda sin egen nätverksanslutning till att rapportera en ungefärlig plats tillbaka till ägaren. När Hitta i nedkopplat läge är aktiverat på en enhet innebär det också att den kan hittas av andra deltagare på samma sätt. Hela den här interaktionen är heltäckande krypterad, anonym och utformad för att vara både batteri- och datasnål. Batteritiden och mobildataanvändningen är minimal och användarens integritet skyddas bättre.

*Obs!* Hitta kanske inte är tillgänglig i alla länder eller regioner.

### Heltäckande kryptering

Hitta bygger på en grund av avancerad kryptografi med publika nycklar. När Hitta i nedkopplat läge är aktiverat i inställningarna för Hitta genereras ett privat EC P-224-krypteringsnyckelpar som skrivs {d,P} direkt på enheten där *d* är den privata nyckeln och *P* är den publika nyckeln. Dessutom initieras en 256-bitars hemlig $SK_0$ och en räknare $_i$ till noll. Det här privata nyckelparet och hemligheten skickas aldrig till Apple och synkroniseras endast till användarens andra enheter med heltäckande kryptering via iCloud-nyckelring. Hemligheten och räknaren används till att härleda aktuell symmetrisk nyckel-$SK_i$ med följande rekursiva konstruktion: $SK_i = KDF(SK_{i-1}, \text{”update”})$.

Baserat på nyckel-$SK_i$, beräknas två stora heltal $u_i$ och $v_i$ med $(u_i,v_i) = KDF(SK_i, \text{”diversify”})$. Både den privata P-224-nyckeln kallad *d* och motsvarande publika nyckel kallad *P* härleds sedan med en affin relation inkluderande de två heltalen för att beräkna ett tidsbegränsat nyckelpar: Den härledda privata nyckeln är $d_i$ där $d_i = u_i * d + v_i$ (modulo ordningen för P-224-kurvan) och motsvarande publika del är $P_i$ och verifierar $P_i = u_i*P + v_i*G$.

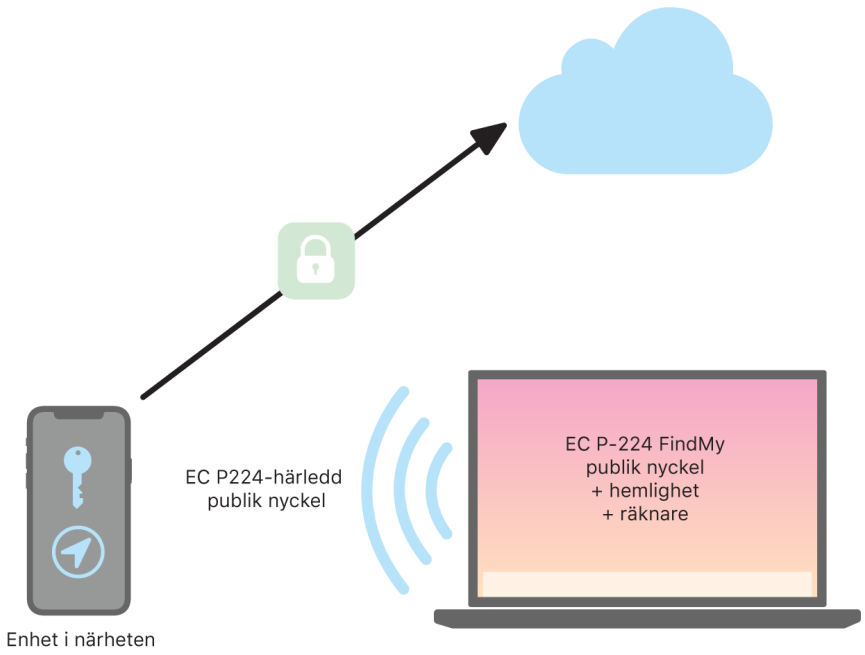
När en enhet försvinner och inte kan ansluta till ett Wi-Fi- eller ett mobilnätverk – t.ex. en kvarglömd MacBook Pro på en parkbänk – börjar den periodiskt skicka ut den härledda publika nyckeln $P_i$ under en begränsad tidsperiod i en Bluetooth-nyttolast. Genom att använda P-224 kan den publika nyckelrepresentationen rymmas i en enda Bluetooth-nyttolast. Enheter i närheten kan då hjälpa till att hitta den nedkopplade enheten genom att kryptera deras platser till den publika nyckeln. Ungefär var 15 minut ersätts den publika nyckeln med en ny med ett stegvis ökat värde på räknaren och den process som beskrivs ovan så att användaren inte kan spåras av en beständig identifierare. Härledningsmekanismen är utformad för att förhindra att de olika publika nycklarna $P_i$ länkas till samma enhet.

### Hålla användare och enheter anonyma

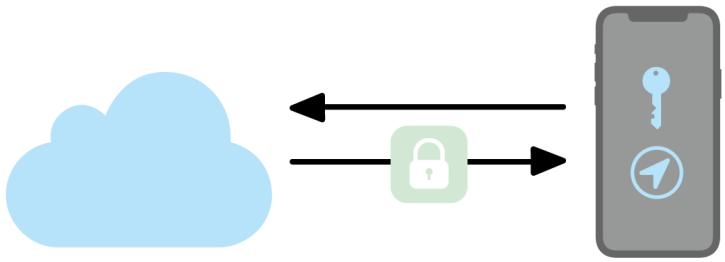
Utöver att säkerställa att platsinformation och andra data är helt krypterade är deltagarnas identiteter anonyma för varandra och för Apple. Den trafik som skickas till Apple av upptäckarenheter innehåller inte någon autentiseringsinformation vare sig i innehållet eller i rubriker. Därför vet inte Apple vem upphittaren är och vems enhet som har hittats. Dessutom loggar Apple inte någon information som kan avslöja upphittarens identitet och bevarar ingen information som gör det möjligt för någon att koppla samman upphittaren och ägaren. Enhetens ägare får bara den krypterade platsinformation som avkrypteras och visas i Hitta utan någon information om vem som hittat enheten.

## Använda Hitta till att hitta saknade Apple-enheter

Alla Apple-enheter inom räckvidden för Bluetooth, och som har Hitta i nedkopplat läge aktiverat, kan upptäcka en signal från en annan Apple-enhet som konfigurerats för att tillåta Hitta och läsa nyckeln $P_i$ som just sänds. Med en ECIES-konstruktion och den publika nyckeln $P_i$ från sändningen kan upptäckarenheterna kryptera sin platsinformation och vidarebefordra den till Apple. Den krypterade platsen är kopplad till ett serverindex som beräknas som SHA256-hashen för den publika P-224-nyckeln $P_i$ som hämtats från Bluetooth-nyttolasten. Apple har aldrig tillgång till avkrypteringsnyckeln, så Apple kan inte läsa den krypterade platsen. Personen som äger den borttappade enheten kan rekonstruera indexet och avkryptera den krypterade platsen.



Vid försöket att hitta en borttappad enhet uppskattas ett förväntat intervall av räknarvärden för platssökningsperioden. Med informationen om den ursprungliga privata P-224-nyckeln *d* och de hemliga värdena $SK_i$ i intervallet av räknarvärden under sökperioden kan ägaren sedan rekonstruera värdeuppsättningen $\{d_i, SHA256(P_i)\}$ för hela sökperioden. Den av ägarens enheter som används till att hitta den borttappade enheten kan sedan skicka förfrågningar till servern med uppsättningen av indexvärden $SHA256(P_i)$ och hämta de krypterade platserna från servern. Appen Hitta avkrypterar sedan de krypterade platserna lokalt med de matchande privata nycklarna $d_i$ och visar en ungefärlig plats för den saknade enheten i appen. Platsrapporter från flera upptäckarenheter kombineras i ägarens app för att generera en mer exakt plats.




## Hitta nedkopplade enheter

Om en användare har aktiverat Hitta min iPhone på sin enhet är Hitta i nedkopplat läge som förval aktiverat vid uppgradering av enheten till iOS 13 eller senare, iPadOS 13.1 eller senare eller macOS 10.15 eller senare. Detta ska säkerställa att varje användare får bästa möjliga chans att hitta sin enhet om den blir borttappad. En användare som inte vill delta kan dock när som helst avaktivera Hitta i nedkopplat läge i inställningarna för Hitta på sin enhet. När Hitta i nedkopplat läge är avaktiverat fungerar inte enheten längre som upptäckarenhet, och den går heller inte att upptäcka av andra upptäckarenheter. Användaren kan dock fortfarande hitta enheten så länge den kan ansluta till ett Wi-Fi- eller mobilnätverk.

När en saknad ej uppkopplad enhet hittas får användaren en notis och ett mejl om att enheten har hittats. Användaren kan visa platsen för den borttappade enheten genom att öppna Hitta och välja fliken Enheter. Istället för att visa enheten på en tom karta, som det skulle ha varit innan enheten hittades, visar Hitta en kartplats med en ungefärlig adress och information om för hur länge sedan enheten upptäcktes. Om fler platsrapporter kommer in blir den aktuella platsen och tidsstämpeln automatiskt uppdaterade. Även om användarna inte kan spela upp något ljud på en ej uppkopplad enhet, eller fjärradera den, kan de använda platsinformationen till att hitta tillbaka till enheten eller vidta andra åtgärder för att få tillbaka den.

# Kontinuitet

## Säkerhet för Kontinuitet i översikt

Kontinuitetsfunktionen drar nytta av teknik som iCloud, Bluetooth och Wi-Fi och gör det möjligt för användare att påbörja en aktivitet på en enhet och fortsätta den på en annan, ringa och ta emot samtal, skicka och ta emot SMS och dela en mobil internetanslutning.

## Handoff-säkerhet

Apple hanterar överlämningar på ett säkert sätt, både mellan enheter och mellan en inbyggd app och en webbplats, även när de involverar stora mängder data.

### Hur Handoff fungerar på ett säkert sätt

När en användares iOS-, iPadOS- och macOS-enheter är i närheten av varandra ser Handoff till att användaren automatiskt kan övergå från att arbeta på en enhet till en annan. Med Handoff kan användaren växla mellan enheter och fortsätta jobba direkt.

När en användare loggar in på iCloud på en andra enhet som är Handoff-kompatibel upprättar de två enheterna en direkt parkoppling via Bluetooth Low Energy (BLE) 4.2 med hjälp av APNs. De enskilda meddelandena krypteras på ett liknande sätt som meddelanden i iMessage. När enheterna har parkopplats genererar var och en av dem en symmetrisk 256-bitars AES-nyckel som lagras i enhetens nyckelring. Den här nyckeln kan kryptera och autentisera BLE-annonseringar som förmedlar enhetens aktuella aktivitet till andra iCloud-parkopplade enheter via AES256 i GCM-läge, med skyddsåtgärder mot replay-angrepp.

Första gången en enhet tar emot en annonsering från en ny nyckel upprättar den en BLE-anslutning till den första enheten och utför en utväxling av krypteringsnycklar. Anslutningen skyddas med standardkryptering för BLE 4.2 samt kryptering av de enskilda meddelandena – en metod som påminner om krypteringen i iMessage. I vissa fall skickas dessa meddelanden via APNs istället för BLE. Aktivitetens innehåll skyddas och överförs på samma sätt som ett iMessage-meddelande.

### Handoff mellan inbyggda appar och webbplatser

Med Handoff kan inbyggda appar i iOS, iPadOS och macOS återuppta användaraktivitet på en webbsida i domäner som styrs och kontrolleras av apputvecklaren. Användarens aktivitet i en inbyggd app kan också återupptas i en webbläsare.

För att förhindra att inbyggda appar återupptar webbplatser som inte kontrolleras av utvecklaren måste appen bevisa att den har legitim kontroll över de webbdomäner den försöker återuppta. Kontroll över en webbplats avgörs via mekanismen för delade webbinloggningsuppgifter. Mer information finns i Apptillgång till sparade lösenord. Systemet måste kunna bekräfta att appen har kontroll över domännamnet innan appen får godkänna Handoff för användaraktiviteter.

Källan till en Handoff för en webbsida kan vara vilken webbläsare som helst som använder API:erna för Handoff. När användaren öppnar en webbsida tillkännager systemet webbsidans domännamn i form av en krypterad Handoff-annonsering. Det är bara användarens andra enheter som kan avkryptera annonseringen.

På den mottagande enheten upptäcker systemet att en installerad Apple-app godkänner Handoff från det annonserade domännamnet och visar Apple-appens symbol som Handoff-alternativ. När Apple-appen startas tar den emot den fullständiga webbadressen och titeln på webbsidan. Ingen annan information överförs från webbläsaren till appen.

I andra riktningen fungerar det så att en Apple-app kan ange en reserv-URL som används om enheten som tar emot Handoff inte har samma app installerad. I detta fall visar systemet användarens förvalda webbläsare som alternativ till Handoff-appen (förutsatt att den webbläsaren använder Handoff-API:er). När en begäran om Handoff görs startas webbläsaren och öppnas med den reserv-URL som källappen har tillhandahållit. Reserv-URL:en behöver inte vara begränsad till de domännamn som kontrolleras av apputvecklaren.

## Handoff av större datamängder

Utöver att använda den grundläggande funktionen i Handoff kan vissa appar välja att använda API:er som har stöd för att skicka större mängder data via Apples P2P-teknik för Wi-Fi-anslutning direkt mellan enheter (ungefär som AirDrop). Ett exempel är Mail-appen som använder dessa API:er som stöder överlämning av mejlutkast som kan innehålla stora bilagor.

När en app använder de här API:erna startar utbytet mellan de två enheterna precis som i Handoff. Däremot upprättar den mottagande enheten en ny anslutning via Wi-Fi när den har tagit emot ett första informationsinnehåll via Bluetooth Low Energy (BLE). Den här anslutningen krypteras (med TLS) och härleder tillförlitlighet genom en identitet som delas via iCloud-nyckelring. Identiteten i certifikaten kontrolleras mot användarens identitet. Därefter skickas efterföljande information via denna krypterade anslutning tills överföringen är klar.

## Universella urklipp

Universella urklipp drar nytta av Handoff för att säkert överföra innehållet i en användares urklipp mellan enheter så att användaren kan kopiera på en enhet och klistra in på en annan. Innehållet skyddas på samma sätt som andra Handoff-data. Om apputvecklaren inte väljer att förhindra delning delas innehållet med universella urklipp som förval.

Appar har tillgång till urklippsdata oavsett om användaren har klistrat in urklipp i appen eller ej. Med universella urklipp sträcker sig denna datatillgång till appar på användarens andra enheter (som är inloggade på iCloud).

## Säkerhet och vidarekoppling av mobilsamtal från iPhone

När en användares Mac, iPad eller HomePod finns i samma Wi-Fi-nätverk som dennas iPhone kan den ringa och besvara telefonsamtal via iPhones mobilabonnemang. För att det ska fungera måste alla enheterna vara inloggade på både iCloud och FaceTime med samma Apple-ID-konto.

När ett samtal kommer in meddelas alla konfigurerade enheter via APNs (Apples pushnotistjänst). Alla enskilda notiser använder samma heltäckande kryptering som iMessage använder. Enheter som finns i samma nätverk visar en notis om inkommande samtal i användargränssnittet. När användaren besvarar samtalet överförs ljudet direkt från användarens iPhone via en säker P2P-anslutning mellan enheterna.

När ett samtal besvaras på en enhet slutar det att ringa på enheter som är iCloud-parkopplade i närheten genom att en kort annonsering visas via Bluetooth Low Energy (BLE). Annonseringen krypteras på samma sätt som Handoff-annonseringar.

Utgående samtal vidarekopplas också till iPhone via APNs och ljudet skickas på liknande sätt via den säkra P2P-länken mellan enheterna. Användarna kan avaktivera vidarekoppling av samtal på en enhet genom att stänga av iPhone-mobilsamtal i FaceTime-inställningarna.

## Säkerhet och SMS-koppling från iPhone

Vid SMS-koppling skickas SMS som tas emot på användarens iPhone automatiskt till användarens registrerade iPad eller Mac. Alla enheter måste vara inloggade till iMessage-tjänsten med samma Apple-ID. När SMS-koppling är påslagen registreras enheter inom en användares tillförlitlighetscirkel automatiskt om tvåfaktorsautentisering är aktiverad. I annat fall verifieras registreringen på enskilda enheter genom att skriva in en slumpmässig sexsiffrig kod som genereras av iPhone.

När enheterna är länkade till varandra krypterar och vidarebefordrar iPhone inkommande SMS till varje enhet med hjälp av de metoder som beskrivs under iMessage-säkerhet i översikt. Svaren skickas tillbaka till iPhone med samma metod, och iPhone skickar sedan svaret i form av ett SMS via operatörens överföringsmetod för SMS. SMS-koppling kan aktiveras och avaktiveras i inställningarna för Meddelanden.

## Säkerhet för Instant Hotspot

Instant Hotspot ansluter andra Apple-enheter till en personlig iPhone- eller iPad-anslutning. iPhone- och iPad-enheter som har stöd för Instant Hotspot använder BLE (Bluetooth Low Energy) för att upptäcka och kommunicera med enheter som är inloggade på samma enskilda iCloud-konto eller konton som används med familjedelning (i iOS 13 och iPadOS). Kompatibla Mac-datorer med OS X 10.10 eller senare använder samma teknik för att upptäcka och kommunicera med iPhone- och iPad-enheter som använder Instant Hotspot.

När en användare först anger Wi-Fi-inställningar på en enhet sänder den ut en BLE-annonsering med en identifierare som alla enheter som är inloggade på samma iCloud-konto har kommit överens om. Identifieraren genereras från ett DSID (Destination Signaling Identifier) som är kopplat till iCloud-kontot och byts ut med jämna mellanrum. När andra enheter som är inloggade på samma iCloud-konto finns i närheten och har stöd för Internetdelning känner de av signalen och svarar med att signalera tillgänglighet för användning av Instant Hotspot.

När en användare som inte ingår i familjedelningen väljer en iPhone eller iPad som är tillgänglig för Internetdelning skickas en förfrågan till enheten om att aktivera Internetdelning. Förfrågan skickas via en länk som är krypterad med BLE-kryptering och förfrågan krypteras på ett sätt som liknar det för iMessage-meddelanden. Enheten svarar sedan genom att skicka anslutningsinformation för Internetdelning via samma BLE-länk med samma typ av kryptering för enskilda meddelanden.

För användare som ingår i en familjedelning delas anslutningsinformationen för Internetdelning med en mekanism som liknar den som används för synkronisering av information av HomeKit-enheter. Anslutningen som delar anslutningsinformation mellan användare skyddas med en tillfällig ECDH-nyckel (Curve25519) som autentiseras med användarnas respektive enhetsspecifika publika Ed25519-nycklar. De publika nycklar som används är de som tidigare har synkroniserats mellan medlemmarna i en familjedelning med IDS när familjedelningen upprättades.

# Nätverkssäkerhet

## Nätverkssäkerhet i översikt

Förutom de inbyggda säkerhetsfunktionerna som Apple använder för att skydda data på Apple-enheter finns det många åtgärder som företag och organisationer kan vidta för att skydda information som överförs till och från en enhet. Alla de här skyddsfunktionerna och åtgärderna faller under nätverkssäkerhet.

Eftersom användare behöver kunna ansluta till företagets nätverk oavsett var de befinner sig är det viktigt att se till att de är auktoriserade och att deras data skyddas under överföringen. För att uppnå den här höga säkerhetsnivån använder iOS, iPadOS och macOS beprövade tekniker och de senaste standarderna för både Wi-Fi och mobildatanät. Det är därför våra operativsystem använder – och ger utvecklare tillgång till – vanliga nätverksprotokoll för autentiserad, behörighetsskyddad och krypterad kommunikation.

## TLS-säkerhet

iOS, iPadOS och macOS har stöd för TLS-säkerhet (TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3) och DTLS (Datagram Transport Layer Security). TLS-protokollet stöder både AES128 och AES256 samt föredrar kodpaket med FS (Forward Secrecy). Internetappar som Safari, Kalender och Mail använder det här protokollet automatiskt för att skapa en krypterad kommunikationskanal mellan enheter och nätverkstjänster. API:er på hög nivå (som CFNetwork) gör det enkelt för utvecklare att använda TLS i sina appar, medan API:er på låg nivå (som Network.framework) ger en mer finmaskig kontroll. SSL 3 tillåts inte av CFNetwork och appar som använder WebKit (t.ex. Safari) får inte öppna SSL 3-anslutningar.

I iOS 11 eller senare och macOS 10.13 eller senare är SHA-1-certifikat inte längre tillåtna för TLS-anslutningar, med undantag för om de är betrodda av användaren. Certifikat med RSA-nycklar som är kortare än 2048 bitar är inte heller tillåtna. Den symmetriska kodningsgruppen RC4 är utfasad i iOS 10 och macOS 10.12. Som förval är RC4 inte aktiverat för TLS-klienter eller -servrar som implementeras med SecureTransport API, och de kan inte ansluta när RC4 är enda tillgängliga kodningsgruppen. För att öka säkerheten bör tjänster och appar som kräver RC4 uppgraderas för användning med säkra kodningsgrupper. I iOS 12.1 måste certifikat som är utfärdade efter den 15 oktober 2018 från ett systembetrott rotcertifikat loggas i en betrodd Certificate Transparency-logg för att tillåtas för TLS-anslutningar. I iOS 12.2 är TLS 1.3 aktiverat som förval för Network.framework och NSURLSession-API:er. TLS-klienter som använder SecureTransport-API:erna kan inte använda TLS 1.3.

## App Transport Security

App Transport Security tillhandahåller förvalda anslutningskrav så att appar följer bästa praxis för säkra anslutningar när API:erna NSURLConnection, CFURL eller NSURLSession används. Som förval begränsar App Transport Security kodvalet till att endast inkludera grupper som tillhandahåller FS (forward secrecy), mer specifikt:

- ECDHE_ECDSA_AES och ECDHE_RSA_AES i GCM-läge (Galois/Counter Mode)
- CBC-läge (Cipher Block Chaining)

Appar kan avaktivera FS-kravet per domän. I sådana fall läggs RSA_AES till i uppsättningen tillgängliga koder.

Servrarna måste ha stöd för TLS 1.2 och FS, och certifikaten måste vara giltiga och signerade med SHA256 eller starkare och ha minst en 2048-bitars RSA-nyckel eller en elliptisk 256-bitars kurvnyckel.

Nätverksanslutningar som inte uppfyller dessa krav kommer att misslyckas, förutsatt att appen inte förbigår App Transport Security. Ogiltiga certifikat leder obevekligen till fel och att ingen anslutning upprättas. App Transport Security används automatiskt i appar som kompileras för iOS 9 eller senare och för macOS 10.11 eller senare.

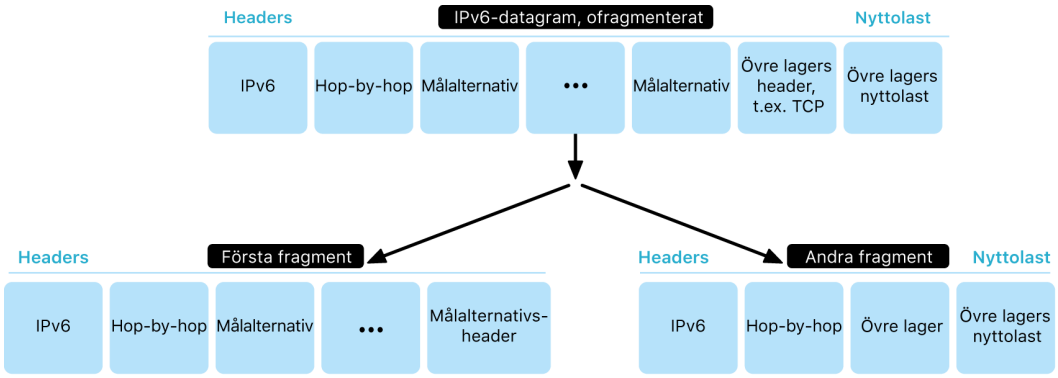
## Kontroll av certifikatvalidering

Utvärdering av den betrodda statusen för ett TLS-certifikat sker i enlighet med etablerad branschstandard, beskrivet i RFC 5280, och innehåller blivande standarder som RFC 6962 (certifikattransparens). I iOS 11 eller senare, samt macOS 10.13 eller senare, uppdateras Apple-enheter regelbundet med en aktuell lista över återkallade och begränsade certifikat. Listan sammanställs utifrån certifikatåterkallningslistor (CRL) som publiceras av de olika inbyggda rotcertifikatutfärdare som är betrodda av Apple, liksom av deras underordnade certifikatutfärdare. Listan kan också innehålla andra begränsningar efter beslut från Apple. Den här informationen efterfrågas varje gång en nätverks-API-funktion används till att skapa en säker anslutning. Om det finns för många återkallade certifikat från en certifikatutfärdare för att lista dem ett och ett kan en tillförlitlighetsutvärderingen istället kräva ett OCSP (online certificate status response), och tillförlitlighetsutvärderingen misslyckas om svaret inte är tillgängligt.

# IPv6-säkerhet

Alla Apple-operativsystem har stöd för IPv6 som implementerar flera mekanismer för att skydda användarnas integritet och nätverksstackens stabilitet. När SLAAC (Stateless Address Autoconfiguration) används genereras IPv6-adressen för alla gränssnitt på ett sätt som förhindrar att enheter spåras över nätverk och samtidigt tillåter en bra användarupplevelse genom att säkerställa adresstabilitet då inga nätverksändringar sker. Algoritmen för adressgenerering baseras på kryptografiskt genererade adresser, med start från RFC 3972, förstärkt av en gränssnittsspecifik modifierare för att garantera att även olika gränssnitt i samma nätverk i slutändan får olika adresser. Dessutom skapas tillfälliga adresser med en förvald giltighet på 24 timmar, och dessa används som förval för alla nya anslutningar. I enlighet med funktionen för privat Wi-Fi-adress som introducerades i iOS 14, iPadOS 14 och watchOS 7 genereras en unik Link-Local Address för varje Wi-Fi-nätverk som en enhet ansluter till. Nätverkets SSID används sedan som ett ytterligare element i adressgenereringen, på liknande sätt som parametern Network_ID som beskrivs i RFC 7217. Den här metoden används i iOS 14, iPadOS 14 och watchOS 7.

För att skydda mot attacker som baseras på IPv6-tilläggshuvud och fragmentering använder Apples enheter skyddsåtgärder som anges i RFC 6980, RFC 7112 och RFC 8021. De förhindrar bland annat attacker där huvudet i det övre lagret endast finns i det andra fragmentet (visas nedan), vilket i sin tur kan leda till tvetydigheter vid säkerhetskontroller som paketfilter utan status.




Apple-enheter har dessutom olika gränser för IPv6-relaterade datastrukturer, som antalet prefix per gränssnitt, för att garantera att IPv6-stacken i Apples operativsystem är stabil.

# VPN-säkerhet (Virtual Private Network)

Säkra nätverkstjänster som VPN (virtuella privata nätverk) kräver oftast minimala insatser vad gäller installation och konfiguration för att fungera med iPhone-, iPad- och Mac-enheter.

## Protokoll som stöds

Dessa enheter fungerar med VPN-servrar som stöder följande protokoll och autentiseringsmetoder:

- IKEv2/IPsec med autentisering med en delad hemlighet, RSA-certifikat, ECDSA-certifikat, EAP-MSCHAPv2 eller EAP-TLS.
- SSL-VPN med lämplig klientapp från App Store.
- L2TP/IPsec med användarautentisering via MS-CHAPV2-lösenord och maskinautentisering med en delad hemlighet (iOS, iPadOS och macOS) och RSA SecurID eller CRYPTOCard (endast macOS).
- Cisco IPsec med användarautentisering via lösenord, RSA SecurID eller CRYPTOCard och maskinautentisering med en delad hemlighet och certifikat (endast macOS).

## VPN-driftsättningar som stöds

iOS, iPadOS och macOS har stöd för följande:

- *VPN On Demand:* För nätverk som använder certifikatbaserad autentisering. IT-policyer anger vilka domäner som kräver en VPN-anslutning genom att använda en konfigurationsprofil.
- *VPN per app:* Möjliggör mycket mer detaljkontrollerade VPN-anslutningar. MDM-lösningar kan ange en anslutning för varje hanterad app och specifika domäner i Safari. Detta bidrar till att säkra data alltid skickas till och från företagets nätverk – och att användarnas personliga data inte skickas.

iOS och iPadOS har stöd för följande:

- *Alltid på-VPN:* För enheter som hanteras via en MDM-lösning och övervakas med Apple Configurator för Mac, Apple School Manager, Apple Business Manager eller Apple Business Essentials. Alltid på-VPN gör att användarna slipper slå på VPN för att aktivera skyddet när de ansluter till mobilnät och Wi-Fi-nätverk. Det ger också organisationen full kontroll över trafiken till och från enheter genom att all IP-trafik dirigeras genom en tunnel tillbaka till organisationen. Det förvalda utbytet av parametrar och nycklar för den efterföljande krypteringen, IKEv2, skyddar trafiköverföringar med datakryptering. Organisationen kan övervaka och filtrera trafiken till och från enheter, skydda data inom nätverket och begränsa enheters tillgång till internet.

# Wi-Fi-säkerhet

## Säker åtkomst till trådlösa nätverk

Alla Apples plattformar har stöd för branschstandardprotokoll för Wi-Fi-autentisering och -kryptering för att tillhandahålla autentiserad åtkomst och konfidentialitet vid anslutning till följande säkra trådlösa nätverk:

- WPA2 Personal
- WPA2 Enterprise
- WPA2/WPA3 Transitional
- WPA3 Personal
- WPA3 Enterprise
- WPA3 Enterprise med 192-bitars säkerhet

WPA2 och WPA3 autentiserar varje anslutning och tillhandahåller 128-bitars AES-kryptering för att säkerställa konfidentialitet för data som skickas trådlöst. Det här innebär att användarna kan känna sig trygga i vetskapen om att deras data förblir skyddade när de skickar och tar emot kommunikation via en Wi-Fi-anslutning.

### WPA3-stöd

WPA3 stöds på följande Apple-enheter:

- iPhone 7 eller senare
- iPad femte generationen eller senare
- Apple TV 4K eller senare
- Apple Watch Series 3 eller senare
- Mac-datorer (sent 2013 eller senare med 802.11ac eller senare)

Nyare enheter stöder autentisering med WPA3 Enterprise med 192-bitars säkerhet, vilket omfattar stöd för 256-bitars AES-kryptering vid anslutning till kompatibla trådlösa anslutningspunkter. Den här krypteringen ger ännu starkare konfidentialitetsskydd för trådlös trafik. WPA3 Enterprise med 192-bitars säkerhet stöds i alla modeller av iPhone 11 eller senare, alla iPad-modeller från och med iPad sjunde generationen och alla Mac-datorer med Apple Silicon.

## PMF-stöd

Apples plattformar skyddar inte bara data som skickas trådlöst, utan utökar WPA2- och WPA3-nivåskydden till unicast- och multicast-hanteringsramar via tjänsten PMF (Protected Management Frame) som definieras i 802.11w. PMF-stöd är tillgängligt på följande Apple-enheter:

- iPhone 6 eller senare
- iPad Air 2 eller senare
- Apple TV HD eller senare
- Apple Watch Series 3 eller senare
- Mac-datorer (sent 2013 eller senare med 802.11ac eller senare)

Tack vare stödet för 802.1X kan Apple-enheter integreras i en mängd olika miljöer med RADIUS-autentisering. Bland de trådlösa autentiseringsmetoder med 802.1X som stöds finns EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAP 0 och PEAP 1.

## Plattformsskydd

Apple-operativsystemen skyddar enheten från sårbarheter i nätverksprocessorns fasta programvara. Det här innebär att nätverksstyrenheter med Wi-Fi har begränsad åtkomst till approcessorminne.

- När USB eller SDIO (Secure Digital Input Output) används som gränssnitt till nätverksprocessorn kan nätverksprocessorn inte inleda DMA-transaktioner (Direct Memory Access) till approcessorn.
- När PCIe används är varje nätverksprocessor en egen isolerad PCIe-buss. En IOMMU (Input/Output Memory Management Unit ) på varje PCIe-buss begränsar nätverksprocessorns DMA-behörighet ytterligare till endast minne och resurser som innehåller dess nätverkspaket och kontrollstrukturer.

## Utfasade protokoll

Apple-produkter har stöd för följande utfasade protokoll för Wi-Fi-autentisering och -kryptering:

- WEP Open med både 40-bitars och 104-bitars nycklar
- WEP Shared med både 40-bitars och 104-bitars nycklar
- Dynamisk WEP
- TKIP (Temporal Key Integrity Protocol)
- WPA
- WPA/WPA2 Transitional

Dessa protokoll anses inte längre vara säkra, och deras användning avrådes starkt av kompatibilitets-, tillförlitlighets-, prestanda- och säkerhetsskäl. De stöds endast för att tillhandahålla bakåtkompatibilitet och kan tas bort i framtida programvaruversioner.

Byte till WPA3 Personal eller WPA3 Enterprise rekommenderas för alla Wi-Fi-implementationer, detta för att tillhandahålla så stabila, säkra och kompatibla Wi-Fi-anslutningar som möjligt.

# Wi-Fi-integritet

## Slumpgenererade MAC-adresser

Apples plattformar använder en slumpgenererad MAC-adress (Media Access Control) när Wi-Fi-sökning utförs när de inte är kopplade till något Wi-Fi-nätverk. Dessa sökningar kan utföras för att hitta och ansluta till ett känt Wi-Fi-nätverk eller för att hjälpa Platstjänster för appar som använder geostängsel, som platsbaserade påminnelser eller fastställning av en plats i Apples Kartor. Observera att Wi-Fi-sökningar som utförs vid försök att ansluta till ett förvalt Wi-Fi-nätverk inte är slumpmässiga. Slumpgenererade Wi-Fi-MAC-adresser stöds på iPhone 5 och senare.

Apples plattformar använder också en slumpgenererad MAC-adress vid ePNO-sökningar (enhanced Preferred Network Offload) när en enhet inte är kopplad till ett Wi-Fi-nätverk eller dess processor är i vila. ePNO-sökningar körs när en app på enheten använder platstjänster som drar nytta av så kallade geostängsel, till exempel platsbaserade påminnelser som känner av när enheten är i närheten av en specifik plats.

Eftersom en enhets MAC-adress ändras när den kopplas ned från ett Wi-Fi-nätverk kan en passiv observatör av Wi-Fi-trafik inte använda den till att spåra enheten kontinuerligt, även om enheten är ansluten till ett mobilnät. Apple har informerat tillverkare av Wi-Fi-utrustning om att Wi-Fi-sökningar i iOS och iPadOS använder en slumpgenererad MAC-adress och att varken Apple eller tillverkarna kan förutsäga dessa slumpgenererade MAC-adresser.

När en iPhone, iPad eller Apple Watch (med iOS 14 eller senare, iPadOS 14 eller senare och watchOS 7 eller senare) ansluter till ett Wi-Fi-nätverk identifierar enheten sig själv med en unik (slumpmässig) MAC-adress per nätverk. Den här funktionen kan avaktiveras av antingen användaren eller genom användning av ett nytt alternativ i nyttolasten Wi-Fi. Under vissa omständigheter återgår enheten till användning av den faktiska MAC-adressen.

Mer information finns i Apple Support-artikeln Använda privata wifi-adresser på iPhone, iPad, iPod touch och Apple Watch.

## Slumpgenerering av sekvensnummer för Wi-Fi-ramar

Wi-Fi-ramar innehåller ett sekvensnummer som används i lågnivåprotokollet 802.11 för effektiv och tillförlitlig Wi-Fi-kommunikation. Eftersom dessa sekvensnummer stiger för varje överförd ram kan de vid Wi-Fi-sökning användas till att koppla samman överförd information med andra ramar som skickas av samma enhet.

Som skydd mot detta skapar Apple-enheter slumpgenererade sekvensnummer varje gång en MAC-adress ändras till en ny slumpgenererad adress. Detta inkluderar slumpgenererade sekvensnummer för varje ny sökningsförfrågan som initieras när enheten inte är kopplad. Den här slumpfunktionen stöds på följande enheter:

- iPhone 7 eller senare
- iPad femte generationen eller senare
- Apple TV 4K eller senare
- Apple Watch Series 3 eller senare
- iMac Pro (Retina 5K, 27-tums, 2017) eller senare
- MacBook Pro (13-tums, 2018) eller senare
- MacBook Pro (15-tums, 2018) eller senare
- MacBook Air (Retina, 13-tums, 2018) eller senare
- Mac mini (2018) eller senare
- iMac (Retina 4K, 21,5-tums, 2019) eller senare
- iMac (Retina 5K, 27-tums, 2019) eller senare
- Mac Pro (2019) eller senare

## Wi-Fi-anslutningar

Apple genererar slumpmässiga MAC-adresser för de P2P-Wi-Fi-anslutningar som används för AirDrop och AirPlay. Slumpgenererade adresser används också för Internetdelning i iOS och iPadOS (med ett SIM-kort) och Internetdelning i macOS.

Nya, slumpmässiga adresser genereras varje gång dessa nätverksgränssnitt startas och unika adresser genereras oberoende för varje gränssnitt efter behov.

## Dolda nätverk

Wi-Fi-nätverk identifieras med ett nätverksnamn som kallas för *SSID (Service Set Identifier)*. Vissa Wi-Fi-nätverk är konfigurerade så att deras SSID döljs, vilket leder till att anslutningspunkten inte sänder ut nätverkets namn. Dessa kallas för *dolda nätverk*. iPhone 6s och senare modeller upptäcker automatiskt när ett nätverk är dolt. Om ett nätverk är dolt skickar iOS- eller iPadOS-enheten en testare med det SSID som ingår i förfrågan – annars inte. Detta förhindrar att enheten sänder ut namnet på tidigare dolda nätverk som en användare har varit ansluten till, vilket ytterligare säkerställer integriteten.

# Bluetooth-säkerhet

Det finns två typer av Bluetooth i Apple-enheter: Bluetooth Classic och Bluetooth Low Energy (BLE). Bluetooth-säkerhetsmodellen för båda versionerna innehåller följande distinkta säkerhetsfunktioner:

- *Parkoppling:* Processen för att skapa en eller flera delade hemliga nycklar.
- *Sammanlänkning:* När de nycklar som skapas under parkoppling lagras för att kunna användas till senare anslutningar och på så vis skapa ett betrott enhetspar.
- *Autentisering:* Verifiering av att de två enheterna har samma nycklar.
- *Kryptering:* Meddelandekonfidentialitet.
- *Meddelandeintegritet:* Skydd mot meddelandeförfalskning.
- *SSP (Secure Simple Pairing):* Skydd mot passiv avlyssning och skydd mot man-in-the-middle-angrepp

I Bluetooth 4.1 tillkom funktionen Secure Connections för den fysiska BR/EDR-transporten (Bluetooth Classic).

Säkerhetsfunktionerna för respektive typ av Bluetooth visas i listan nedan.

| Stöd | Bluetooth Classic | Bluetooth Low Energy |
|---|---|---|
| Parkoppling | Elliptisk P-256-kurva | FIPS-godkända algoritmer (AES-CMAC och elliptisk P-256-kurva) |
| Sammanlänkning | Parkopplingsinformation lagras på en säker plats i iOS-, iPadOS-, macOS-, tvOS- och watchOS-enheter | Parkopplingsinformation lagras på en säker plats i iOS-, iPadOS-, macOS-, tvOS- och watchOS-enheter |
| Autentisering | FIPS-godkända algoritmer (HMAC-SHA256 och AES-CTR) | FIPS-godkända algoritmer |
| Kryptering | AES-CCM-kryptering, utförd i styrenheten | AES-CCM-kryptering, utförd i styrenheten |
| Meddelandeintegritet | AES-CCM, används för meddelandeintegritet | AES-CCM, används för meddelandeintegritet |
| SSP (Secure Simple Pairing): Skydd mot passiv avlyssning | ECDHE (Elliptic Curve Diffie-Hellman Exchange Ephemeral) | ECDHE (Elliptic Curve Diffie-Hellman Exchange) |
| SSP (Secure Simple Pairing): Skydd mot man-in-the-middle-angrepp (MITM) | Två användarassisterade numeriska metoder: numerisk jämförelse eller nyckelinmatning | Två användarassisterade numeriska metoder: numerisk jämförelse eller nyckelinmatning<br>Parkoppling kräver svar från användaren, inklusive alla parkopplingslägen för icke-MITM. |
| Bluetooth 4.1 eller senare | iMac sent 2015 eller senare<br>MacBook Pro tidigt 2015 eller senare | iOS 9 eller senare<br>iPadOS 13.1 eller senare<br>macOS 10.12 eller senare<br>tvOS 9 eller senare<br>watchOS 2.0 eller senare |

| Stöd | Bluetooth Classic | Bluetooth Low Energy |
|---|---|---|
| Bluetooth 4.2 eller senare | iPhone 6 eller senare | iOS 9 eller senare<br>iPadOS 13.1 eller senare<br>macOS 10.12 eller senare<br>tvOS 9 eller senare<br>watchOS 2.0 eller senare |

## Integritet för Bluetooth Low Energy (BLE)

BLE innehåller två funktioner som bidrar till användarintegriteten: slumpgenererade adresser och nyckelhärledning genom transportflödet.

*Slumpgenererade adresser* är en funktion som minskar möjligheten att spåra en BLE-enhet över en längre tidsperiod genom att upprepade gånger ändra Bluetooth-enhetens adress. För att en enhet som använder integritetsfunktionen ska kunna återansluta till kända enheter måste den andra enheten kunna lösa enhetsadressen, även kallad den *privata adressen*. Den privata adressen genereras med enhetens IRK-nyckel som utbyttes under parkopplingsprocessen.

iOS 13 eller senare och iPadOS 13.1 eller senare har möjlighet att härleda länknycklar mellan transporter, en funktion som kallas *nyckelhärledning genom transportflödet*. En länk som genereras med BLE kan t.ex. användas till att härleda en Bluetooth Classic-länknyckel. Dessutom har Apple lagt till Bluetooth Classic-till-BLE-stöd för enheter som stöder funktionen Secured Connections som introducerades i Bluetooth Core Specification 4.1 (se Bluetooth Core Specification 5.1).

# Ultra Wideband-säkerhet i iOS

Den nya Apple-utformade U1-kretsen använder tekniken Ultra Wideband för rumsuppfattning, vilket gör det möjligt för iPhone 11, iPhone 11 Pro och iPhone 11 Pro Max eller senare iPhone-modeller att exakt lokalisera andra U1-utrustade Apple-enheter. Ultra Wideband-tekniken använder samma teknik för slumpgenererade data som finns i andra Apple-enheter som stöds:

- Slumpgenererade MAC-adresser
- Slumpgenerering av sekvensnummer för Wi-Fi-ramar

# Säkerhet för enkel inloggning

## Enkel inloggning

iOS och iPadOS stöder autentisering till företagsnätverk genom enkel inloggning (Single sign-on, SSO). SSO fungerar med Kerberos-baserade nätverk och autentiserar användare till tjänster som de har behörighet till. SSO kan användas vid en rad olika nätverksaktiviteter, från säkra Safari-sessioner till tredjepartsappar. Stöd finns även för certifikatbaserad autentisering som PKINIT.

macOS har stöd för autentisering till företagsnätverk med Kerberos. Appar kan använda Kerberos till att autentisera användare till tjänster som de har behörighet till. Kerberos kan också användas vid en rad olika nätverksaktiviteter, från säkra Safari-sessioner och autentisering av nätverksfilsystem till tredjepartsappar. Stöd finns för certifikatbaserad autentisering, även om appar då måste hantera ett utvecklar-API.

Vid enkel inloggning i iOS, iPadOS och macOS används SPNEGO-tokens och HTTP Negotiate-protokollet i kombination med Kerberos-baserade gateways för autentisering och IWA-system (Integrated Windows Authentication) med stöd för Kerberos-biljetter. SSO-stödet bygger på det öppna källkodsprojektet Heimdal.

Följande krypteringstyper hanteras i iOS, iPadOS och macOS:

- AES-128-CTS-HMAC-SHA1-96
- AES-256-CTS-HMAC-SHA1-96
- DES3-CBC-SHA1
- ARCFOUR-HMAC-MD5

Safari stöder SSO och tredjepartsappar som använder standard-API:er för nätverk i iOS och iPadOS kan också konfigureras för det. För konfiguration av SSO har iOS och iPadOS en nyttolast för konfigurationsprofilen som gör det möjligt för MDM-lösningar att bestämma vissa inställningar. Exempel på sådana inställningar är användarnamnet (det vill säga Active Directory-användarkontot) och sfärinställningarna för Kerberos samt konfiguration av vilka appar och Safari-webbadresser som får använda SSO.

## Utökningsbar enkel inloggning

Apputvecklare kan tillhandahålla egna implementeringar av enkel inloggning genom att använda SSO-tillägg. SSO-tillägg anropas när en inbyggd app eller en webbapp behöver en identitetsleverantör för användarautentisering. Utvecklarna kan tillhandahålla två typer av tillägg: de som omdirigerar till HTTPS och de som använder en mekanism för utmaning/svar som exempelvis Kerberos. Detta gör det möjligt för utökningsbar enkel inloggning att ge stöd åt OpenID-, OAuth-, SAML2- och Kerberos-autentiseringsscheman. SSO-tillägg kan även ge stöd för macOS-autentisering genom att anamma ett systemspecifikt SSO-protokoll som gör det möjligt att hämta SSO-token under macOS-inloggning.

För att använda ett tillägg till enkel inloggning kan en app antingen använda AuthenticationServices API eller utnyttja den URL-avlyssningsmekanism som erbjuds av operativsystemet. WebKit och CFNetwork tillhandahåller ett avlyssningsskikt som ger smidigt stöd för enkel inloggning åt alla inbyggda appar och WebKit-appar. För att ett tillägg för enkel inloggning ska kunna anropas måste en konfiguration tillhandahållen av en administratör installeras via en MDM-lösning. Dessutom måste omdirigeringstillägg använda nyttolasten Associated Domains till att bevisa att den identitetsserver som de stöder är medveten om deras existens.

Det enda tillägg som följer med operativsystemet är Kerberos SSO-tillägget.

# AirDrop-säkerhet

Apple-enheter som stöder AirDrop använder Bluetooth Low Energy (BLE) och Apples egen teknik för P2P-Wi-Fi vid överföring av filer och information mellan enheter som befinner sig i närheten av varandra, vilket omfattar AirDrop-kompatibla iOS- och iPad-enheter med iOS 7 eller senare samt Mac-datorer med OS X 10.11 eller senare. Enheterna kommunicerar direkt med varandra genom att sända och ta emot Wi-Fi-signaler, utan att gå via internet eller någon trådlös anslutningspunkt. Den här anslutningen krypteras med TLS.

AirDrop är inställt på att endast dela med kontakter som förval. Användaren kan också välja att låta AirDrop dela med alla, eller helt stänga av funktionen. Organisationer kan begränsa användningen av AirDrop för enheter eller appar som hanteras med en MDM-lösning.

## AirDrop-drift

AirDrop använder iCloud-tjänster till att hjälpa användare att autentisera. När en användare loggar in på iCloud lagras en 2048-bitars RSA-identitet på enheten, och när användaren slår på AirDrop skapas en kort AirDrop-identitetshashkod baserad på de e-postadresser och telefonnummer som är kopplade till användarens Apple-ID.

När en användare väljer AirDrop som metod för delning av ett objekt skickar den sändande enheten en AirDrop-signal som innehåller användarens korta identitetshashkod för AirDrop via BLE. Andra Apple-enheter som är aktiverade, befinner sig i närheten och har aktiverat AirDrop upptäcker signalen och svarar via P2P-Wi-Fi så att den sändande enheten kan upptäcka den svarande enhetens identitet.

I läget Endast kontakter jämförs den mottagna korta identitetshashkoden för AirDrop med hashkoderna för personer som finns i appen Kontakter på den mottagande enheten. Om de matchar varandra svarar den mottagande enheten med sin identitetsinformation via P2P-Wi-Fi. Om de inte matchar varandra svarar enheten inte.

I läget Alla är processen ungefär densamma. Skillnaden är att den mottagande enheten svarar även om det inte finns någon matchning i Kontakter på enheten.

Den sändande enheten initierar sedan en AirDrop-anslutning via P2P-Wi-Fi och använder den enheten till att skicka en lång identitetshash till den mottagande enheten. Om den långa identitetshashen matchar hashkoden för en person som finns i mottagarens kontakter svarar den mottagande enheten med dess långa identitetshash.

Om hashkoderna verifieras visas mottagarens namn och bild (om informationen finns i Kontakter) på avsändarens AirDrop-delningsblad. I iOS och iPadOS visas de under Personer eller Enheter. Enheter som inte har verifierats eller autentiserats visas i sändarens AirDrop-delningsblad med en silhuettsymbol och enhetens namn, enligt vad som står under Inställningar > Allmänt > Om > Namn. I iOS och iPadOS visas de under Andra personer på AirDrop-delningsbladet.

Sändaren kan sedan välja vem han/hon vill dela med. När användaren har gjort sitt val upprättar den sändande enheten en krypterad (TLS) anslutning med den mottagande enheten och enheterna utbyter iCloud-identitetscertifikat. Identiteten i certifikaten kontrolleras mot varje användares kontakter (i appen Kontakter).

Om certifikaten verifieras får den mottagande användaren en förfrågan om att ta emot en inkommande överföring från den identifierade användaren eller enheten. Om flera mottagare har valts upprepas processen för var och en av dem.

# Säkerhet och Wi-Fi-lösenordsdelning på iPhone och iPad

iPhone- och iPad-enheter som stöder Wi-Fi-lösenordsdelning använder en mekanism som liknar AirDrop till att skicka ett Wi-Fi-lösenord mellan två enheter.

När en användare väljer ett Wi-Fi-nätverk (sökare) och blir ombedd att ange Wi-Fi-lösenordet startar Apple-enheten en BLE-annonsering som talar om att den vill ha Wi-Fi-lösenordet. Andra Apple-enheter som är vakna, i närheten och har lösenordet för det valda Wi-Fi-nätverket ansluter via BLE till enheten som behöver lösenordet.

Enheten som har Wi-Fi-lösenordet (givare) kräver sökarens kontaktinformation och sökaren måste bevisa sin identitet med en mekanism som påminner om AirDrop. När identiteten är bekräftad skickar givaren en lösenkod som kan användas till att ansluta till nätverket till sökaren.

Organisationer kan begränsa användningen av Wi-Fi-lösenordsdelning för enheter eller appar som hanteras med en MDM-lösning.

# Brandväggssäkerhet och macOS

I macOS finns en inbyggd brandvägg som skyddar datorn från nätverksåtkomst och denial-of-service-angrepp. Den kan konfigureras genom att öppna Systeminställningar > Integritet och säkerhet (macOS 13 eller senare), inställningspanelen Säkerhet och integritet i Systeminställningar (macOS 12 eller tidigare) eller genom att använda en konfigurationsprofil med Firewall-nyttolasten manuellt installerad eller tillhandahållen av en MDM-lösning. Följande konfigurationer stöds:

- Blockera alla inkommande anslutningar, oavsett app.
- Tillåt inbyggd programvara att ta emot inkommande anslutningar automatiskt.
- Tillåt hämtad och signerad programvara att ta emot inkommande anslutningar automatiskt.
- Lägg till eller neka åtkomst baserat på användarangivna appar.
- Förhindra att datorn svarar på ICMP-sondering (Internet Control Message Protocol) och portskanningsförfrågningar.

# Säkerhet och kit för utvecklare

## Säkerhet och kit för utvecklare i översikt

Apple tillhandahåller ett antal ”kitramverk” för att göra det möjligt för tredjepartsutvecklare att utöka Apple-tjänster. Dessa ramverk är byggda med användarens integritet och säkerhet i centrum:

- HomeKit
- CloudKit
- SiriKit
- WidgetKit
- DriverKit
- ReplayKit
- ARKit

## HomeKit-säkerhet

### HomeKit-kommunikationssäkerhet

HomeKit ger tillgång till en infrastruktur för automatisering av hemmet som drar nytta av iCloud och funktioner för enhetssäkerhet för att skydda och synkronisera privata data utan att avslöja dem för Apple.

HomeKit-identiteter och -säkerhet bygger på publika/privata Ed25519-nyckelpar. Ett Ed25519-nyckelpar genereras på användarens enhet som blir dess HomeKit-identitet. Nyckelparet används som en del av HomeKit-tillbehörsprotokollet (HomeKit Accessory Protocol, HAP) till att autentisera direkt kommunikation mellan användarens Apple-enheter och deras HomeKit-tillbehör.

I hem där det finns en hemhubb kan medlemmar i det delade hemmet skicka kommandon till tillbehör via denna hemhubb. Dessa kommandon skickas, heltäckande krypterade och autentiserade, från användarens enheten till hemhubben genom IDS (Apple Identity Service) där de vidarebefordras till det relevanta tillbehöret via HomeKit-tillbehörsprotokollet (HAP) eller den smarta hemanslutningsstandarden Matter.

Nycklarna – som förvaras i nyckelringen och bara inkluderas i krypterade säkerhetskopieringar av nyckelringen – hålls uppdaterade mellan enheter med hjälp av iCloud-nyckelring.

## Kommunikation mellan HomeKit-tillbehör

HomeKit-tillbehör genererar sina egna Ed25519-nyckelpar för kommunikation med Apple-enheter. Om tillbehöret återställs till fabriksinställningarna genereras ett nytt nyckelpar.

För att upprätta ett förhållande mellan en Apple-enhet och ett HomeKit-tillbehör utväxlas nycklar med protokollet Secure Remote Password (3072 bitar) med en åttasiffrig kod som tillhandahålls av tillbehörets tillverkare, anges på användarens enhet och sedan krypteras med ChaCha20-Poly1305 AEAD med HKDF-SHA512-härledda nycklar. Tillbehörets MFi-certifiering kontrolleras också under installationen. Tillbehör som saknar en MFi-krets kan bygga in stöd för programvaruautentisering med iOS 11.3 eller senare.

När enheten och HomeKit-tillbehöret kommunicerar under användning autentiserar de varandra med hjälp av nycklarna som utväxlades i den ovanstående processen. Varje session upprättas med STS-protokollet (Station-to-Station) och krypteras med HKDF-SHA512-härledda nycklar baserade på sessionsgenererade Curve25519-nycklar. Det gäller både IP-baserade tillbehör och BLE-tillbehör (Bluetooth Low Energy).

BLE-tillbehör som har stöd för utsända notiser får en krypteringsnyckel skickad till sig av en parkopplad enhet via en säker session. Nyckeln används till att kryptera data gällande statusförändringar på enheten som meddelas via BLE-annonseringar. Den utsända krypteringsnyckeln är en HKDF-SHA512-härledd nyckel och data krypteras med algoritmen ChaCha20-Poly1305 AEAD. Den utsända krypteringsnyckeln ändras då och då och uppdateras på andra enheter som använder iCloud enligt beskrivningen i avsnittet HomeKit-datasäkerhet.

## Kommunikation med Matter-tillbehör

Identitet och säkerhet med Matter-tillbehör är certifikatbaserade. För Apple-hem genereras certifikatutfärdaren (CA) som är betrodd rot på den inledande användarens enhet (”ägaren”) och den privata nyckeln för CA:n lagras i dess iCloud-nyckelring. Varje Apple-enhet i hemmet genererar en certifikatsigneringsförfrågan (CSR) med NIST P256. Denna CSR tas emot av ägarens enhet, vilket skapar ett Matter-identitetscertifikat för enheten genom att använda dess privata CA-nyckel. Det här certifikatet används sedan till att autentisera kommunikation mellan användares enheter och deras tillbehör.

Matter-tillbehör genererar sitt eget NIST P256-nyckelpar och CSR och får ett certifikat från CA:n när tillbehöret parkopplas. Innan nyckelpar genereras utväxlar Matter-tillbehöret och hemägarens enheter nycklar – via protokollet SPAKE2+ med en PIN-kod som tillhandahålls av tillbehörets tillverkare – och en enhetsattesteringsprocess utförs. CSR och certifikat utväxlas sedan via denna kanal och krypteras med AES-CCM med HKDF-SHA256-härledda nycklar. Om tillbehöret återskapas till fabriksinställningarna genereras ett nytt nyckelpar och en CSR och ett nytt certifikat utfärdas för tillbehöret när det parkopplas.

När en Apple-enhet och Matter-tillbehöret kommunicerar under användning autentiserar vardera certifikat det andra med hjälp av sina egna certifikat. Varje session upprättas med ett trefasprotokoll (sigma) och krypteras med HKDF-SHA256-härledda nycklar baserade på sessionsgenererade P256-nycklar.

För mer information om hur Apple-enheter interagerar säkert med Matter-tillbehör, se Matter support in iOS 16 på Apple Developer-webbplatsen.

### HomeKit och Siri

Siri kan användas till att söka efter och styra tillbehör samt för att aktivera scenarier. Mycket begränsad information om hemmets konfiguration överförs anonymt till Siri. Det gäller namnen på rum, tillbehör och scenarier som är nödvändiga för att förstå kommandon. Ljud som skickas till Siri kan gälla specifika tillbehör eller kommandon, men sådana Siri-data associeras inte med andra Apple-funktioner som HomeKit.

### Siri-kompatibla HomeKit-tillbehör

Användare kan aktivera nya funktioner, som Siri och andra HomePod-funktioner som timers, alarm, Intercom och dörrklockor, på Siri-kompatibla tillbehör via appen Hem. När de här funktionerna aktiveras koordineras tillbehöret med en parkopplad HomePod som agerar värd för Apple-funktionerna i det lokala nätverket. Ljud utväxlas mellan enheterna via krypterade kanaler som använder både HomeKit- och AirPlay-protokoll.

När Lyssna efter ”Hej Siri” är på lyssnar tillbehöret efter frasen ”Hej Siri” genom att använda en lokal motor som känner igen utlösarfrasen. Om den här motorn känner igen frasen skickar den ljudbildrutor direkt till en parkopplad HomePod via HomeKit. HomePod utför en andra kontroll av ljudet och kan avbryta ljudsessionen ifall frasen inte verkar innehålla utlösarfrasen.

När Tryck för Siri är på kan användaren starta en konversation med Siri genom att trycka på en särskild knapp på tillbehöret. Ljudbildrutorna skickas direkt till den HomePod som är parkopplad.

När en Siri-förfrågan upptäcks skickar HomePod ljudet till Siri-servrar och uppfyller användarens avsikt med samma säkerhet, integritet och krypteringsskydd som HomePod använder för användarförfrågningar som görs till själva HomePod. Ifall Siri har ett ljudsvar skickas Siri-svaret via en AirPlay-kanal till tillbehöret. Vissa Siri-förfrågningar kräver mer information från användaren (exempelvis kan användaren få frågan ifall den vill höra fler alternativ). I sådana fall får tillbehöret en indikation om att användaren bör tillfrågas och det ytterligare ljudet strömmas till HomePod.

Tillbehöret måste ha en visuell indikator som visar för användaren när det lyssnar aktivt (exempelvis en LED-indikator). Tillbehöret saknar vetskap om avsikten med en Siri-förfrågan, med undantag för åtkomsten till ljudströmmar, och inga användardata lagras på tillbehöret.

## HomeKit-datasäkerhet

För hem som har uppgraderats till den nya HomeKit-arkitekturen (finns i iOS 16.2 och iPadOS 16.2) synkroniseras HomeKit-data säkert mellan en användares Apple-enheter med hjälp av iCloud och iCloud-nyckelring. Under denna process krypteras HomeKit-data med heltäckande iCloud-kryptering och är inte tillgängliga för Apple.

Användaren som först skapade hemmet i HomeKit (”ägaren”) eller en annan användare med redigeringsbehörighet kan lägga till nya användare. Ägarens enhet konfigurerar tillbehören med den nya användarens publika nyckel så att tillbehöret kan autentisera och ta emot kommandon från den nya användaren. När en användare med redigeringsbehörighet lägger till en ny användare delegeras processen till en hemhubb för att slutföra åtgärden.

### Hemdata och appar

Appars tillgång till hemdata regleras av användare i integritetsinställningarna. Användare uppmanas att tillåta åtkomst när appar begär tillgång till hemdata på liknande sätt som åtkomst till Kontakter, Bilder och andra datakällor i iOS, iPadOS och macOS fungerar. Om användaren ger sitt godkännande får apparna tillgång till namnen på rum, namnen på tillbehör och rummet som varje tillbehör finns i samt annan information som anges i dokumentationen för HomeKit-utvecklare på https://developer.apple.com/homekit/.

### Lokal datalagring

HomeKit lagrar data om hem, tillbehör, scenarier och användare på användarens Apple-enheter. Dessa data lagras med dataskyddsklassen Protected Until First User Authentication och inuti ett datavalv. HomeKit-data säkerhetskopieras inte i lokala säkerhetskopior.

## Säkra routrar med HomeKit

Användare kan förbättra säkerheten för sitt hemnätverk med routrar som har stöd för HomeKit. Med dessa routrar kan användare hantera den Wi-Fi-åtkomst som HomeKit-tillbehör har till sitt lokala nätverk och till internet. Routrarna har även stöd för PPSK-autentisering så att tillbehör kan läggas till i Wi-Fi-nätverket med en nyckel som är specifik för tillbehöret och som kan återkallas vid behov. PPSK-autentisering förbättrar säkerheten eftersom det primära Wi-Fi-lösenordet inte blir tillgängligt för tillbehör, och routern kan dessutom säkert identifiera ett tillbehör även om dess MAC-adress ändras.

Med appen Hem kan användare konfigurera åtkomstbegränsningar för tillbehörsgrupper på följande sätt:

- *Ingen begränsning:* Tillåter obegränsad åtkomst till internet och det lokala nätverket.
- *Automatiskt:* Det här är den förvalda inställningen. Tillåter åtkomst till internet och det lokala nätverket baserat på en lista med internetwebbplatser och lokala portar som tillbehörets tillverkare tillhandahåller till Apple. Den här listan innehåller alla webbplatser och portar som tillbehöret behöver för att fungera ordentligt. (Inställningen Ingen begränsning används tills en lista blir tillgänglig.)
- *Begränsa till Hem:* Ingen åtkomst till internet eller det lokala nätverket med undantag för anslutningar som krävs av HomeKit för att kunna upptäcka och styra tillbehöret från det lokala nätverket (inklusive från hemhubben så att fjärrkontroll stöds).

En PPSK är en stark WPA2 Personal-lösenordsfras som är specifik för ett tillbehör. Den genereras automatiskt av HomeKit och återkallas om och när tillbehöret senare tas bort från hemmet. En PPSK används när ett tillbehör läggs till i Wi-Fi-nätverket av HomeKit i ett hem som har konfigurerats med en HomeKit-router. Detta tillägg visas som Wi-Fi-information: *HomeKit-hanterad* på inställningsskärmen för tillbehöret i appen Hem. Tillbehör som har lagts till i Wi-Fi-nätverket innan routern lades till konfigureras om för användning av en PPSK om tillbehören har stöd för detta. Annars behåller de sina befintliga inloggningsuppgifter.

Som en ytterligare säkerhetsåtgärd måste användarna konfigurera HomeKit-routern via routertillverkarens app så att appen kan bekräfta att användarna har tillgång till routern och kan lägga till den i appen Hem.

# HomeKit-kamerasäkerhet

Kameror med en IP-adress (Internet Protocol) i HomeKit skickar video- och ljudströmmar direkt till den iOS-, iPadOS-, tvOS- och macOS-enhet som tar emot strömmarna i det lokala nätverket. Strömmarna krypteras med slumpmässigt genererade nycklar på både enheten och en IP-kamera, och de utväxlas via den säkra HomeKit-sessionen till kameran. När en enhet inte finns i det lokala nätverket vidarebefordras de krypterade strömmarna via hemhubben till enheten. Hemhubben krypterar inte strömmarna. Den fungerar bara som ett relä mellan enheten och IP-kameran. När en app visar HomeKit-IP-kamerans videovy för användaren renderar HomeKit videobildrutorna säkert från en separat systemprocess. Därför kan appen varken komma åt eller lagra videoströmmen. Utöver detta saknar appar behörighet att göra skärmavbilder från den här strömmen.

## Säker HomeKit-video

HomeKit tillhandahåller en heltäckande krypterad, säker och privat metod för att spela in, analysera och titta på klipp från HomeKit-IP-kameror utan att göra videoinnehållet tillgängligt för Apple eller tredje part. När IP-kameran registrerar rörelser skickas videoklipp direkt till en Apple-enhet som fungerar som hemhubb via en dedikerad lokal nätverksanslutning mellan hemhubben och IP-kameran. Den lokala nätverksanslutningen krypteras med ett sessionsgenererat HKDF-SHA512-härlett nyckelpar som förhandlas via HomeKit-sessionen mellan hemhubben och IP-kameran. HomeKit avkrypterar ljud- och videoströmmar på hemhubben och analyserar videobildrutorna lokalt för att hitta eventuella viktiga händelser. Om en viktig händelse upptäcks krypterar HomeKit videoklippet med AES-256-GCM som har en slumpmässigt genererad AES256-nyckel. HomeKit skapar även vinjettbilder för varje klipp och krypterar vinjettbilderna med samma AES256-nyckel. De krypterade vinjettbilderna samt ljud- och videodata överförs till iCloud-servrar. Relaterade metadata för varje klipp, inklusive krypteringsnyckeln, överförs till CloudKit med heltäckande iCloud-kryptering.

För ansiktsklassificering lagrar HomeKit alla data som används till att klassificera en viss persons ansikte i CloudKit med heltäckande iCloud-kryptering. De data som lagras inkluderar information om varje person, exempelvis namn, liksom bilder på den personens ansikte. Dessa ansiktsbilder kan hämtas från användarens Bilder om användaren väljer att godkänna detta, eller så kan de samlas in från tidigare analyserad IP-kameravideo. En analyssession för säker HomeKit-video använder dessa klassificeringsdata till att identifiera ansikten i den säkra videoström som sessionen tar emot direkt från IP-kameran och inkluderar den identifieringsinformationen i de klippmetadata som nämnts tidigare.

När appen Hem används till att visa klipp från en kamera hämtas data från iCloud. Nycklarna som används till att avkryptera strömmarna packas upp lokalt med heltäckande iCloud-kryptering. Det krypterade videoinnehållet strömmas från strömmarna och avkrypteras lokalt på iOS-enheten innan det visas i visningsfönstret. Varje videoklippssession kan delas upp i mindre delar där varje del krypterar innehållsströmmen med sin egen unika nyckel.

# HomeKit-säkerhet med Apple TV

HomeKit ansluter vissa fjärrkontrollstillbehör från tredje part till Apple TV på ett säkert sätt och gör det möjligt att lägga till användarprofiler till ägaren av hemmets Apple TV.

## Använda fjärrkontrollstillbehör från tredje part med Apple TV

Vissa fjärrkontrollstillbehör från tredje part tillhandahåller HID-händelser (Human Interface Design) och Siri-ljud till en associerad Apple TV som har lagts till via appen Hem. Fjärrkontrollen skickar HID-händelserna via den säkra sessionen till Apple TV-enheten. En Siri-kompatibel TV-fjärrkontroll skickar ljuddata till Apple TV när användaren uttryckligen aktiverar mikrofonen på fjärrkontrollen med en dedikerad Siri-knapp. Fjärrkontrollen skickar ljudbildrutorna direkt till Apple TV via en särskild lokal nätverksanslutning. Ett sessionsgenererat HKDF-SHA512-härlett nyckelpar som förhandlas via HomeKit-sessionen mellan Apple TV och TV-fjärrkontrollen används till att kryptera den lokala nätverksanslutningen. HomeKit avkrypterar ljudrutorna på Apple TV och vidarebefordrar dem till Siri-appen där de behandlas med samma integritetsskydd som all Siri-ljudinmatning.

## Apple TV-profiler för HomeKit-hem

När en användare med ett HomeKit-hem lägger till sin profil till hemmets ägares Apple TV får den användaren tillgång till dess TV-program, musik och podcaster. Inställningar som rör de enskilda användarnas profilanvändning på Apple TV delas till ägarens iCloud-konto via heltäckande iCloud-kryptering. Alla data ägs av den enskilda användaren och delas som skrivskyddade till ägaren. Varje användare i hemmet kan ändra dessa värden i appen Hem och ägarens Apple TV använder dessa inställningar.

När en inställning aktiveras blir användarens iTunes-konto tillgängligt på Apple TV. När en inställning stängs av blir all konto- och datainformation som rör den användaren raderade från Apple TV. Den första CloudKit-delningen inleds av användarens enhet och den token som upprättar den säkra CloudKit-delningen skickas via samma säkra kanal som används till att synkronisera data mellan användare i hemmet.

# SiriKit-säkerhet för iOS, iPadOS och watchOS

Siri använder systemet för apptillägg till att kommunicera med appar från tredje part. På en enhet kan Siri komma åt användarens kontaktinformation och enhetens nuvarande plats. Men innan Siri tillhandahåller skyddade data till en app kontrollerar Siri appens åtkomstbehörigheter som har angetts av användaren. Siri följer de behörigheterna och överlämnar endast den relevanta delen av det användaren ursprungligen sagt till apptillägget. Om en app exempelvis inte har tillgång till kontaktinformation kommer Siri inte att lösa relationen i en användarfråga av slaget ”Skicka 100 kr till mamma med Betalningsapp”. I det här fallet ser appen bara den faktiska termen ”mamma”.

Om användaren däremot har gett appen tillgång till kontaktinformation får appen tillgång till den fullständiga kontaktinformationen för användarens mamma. Om en referens till en relation görs i brödtexten för ett meddelande, t.ex. ”Meddela min mamma i <meddelandeapp> att min bror är fantastisk” löser Siri inte ”min bror” oavsett appens behörigheter.

SiriKit-aktiverade appar kan skicka appspecifik eller användarspecifik vokabulär till Siri, t.ex. namnen på användarens kontakter. Den här informationen gör det möjligt för Siris röstigenkänning och tolkning av naturligt tal att identifiera vokabulär för den appen och är kopplad till en slumpmässig identifierare. Den anpassade informationen fortsätter att vara tillgänglig så länge identifieraren används, tills användaren avaktiverar appens Siri-integration under Inställningar eller tills den SiriKit-aktiverade appen avinstalleras.

För uttalanden som ”Skaffa skjuts hem till mamma med <appnamn>” kräver förfrågan att platsinformation hämtas från användarens kontakter. Siri tillhandahåller då den information som krävs till appens tillägg, endast för denna förfrågan, oavsett vilka användarbehörigheter som har ställts in för plats- eller kontaktinformation för appen.

# WidgetKit-säkerhet

WidgetKit är det ramverk som utvecklare använder till att erbjuda widgetar och klockkomplikationer. Båda kan visa känslig information som är väldigt synlig, särskilt på enheter med en Alltid på-skärm.

I iOS kan användare ställa in om känsliga data ska visas på låsskärmen och när Alltid på-skärmen används. De kan avaktivera dataåtkomst för låsskärmswidgetar i avsnittet ”Tillåt åtkomst från låst skärm” i Inställningar > Face ID och lösenkod.

På Apple Watch kan användare ställa in om känsliga data ska visas på Alltid på-skärmen genom att välja Inställningar > Visning och ljusstyrka > Alltid på > Göm känsliga komplikationer. De kan också välja att visa censurerat innehåll för alla eller enskilda komplikationer.

Om användare väljer att gömma innehåll som de betraktar som privat visar WidgetKit en platshållare eller censureringar. För att konfigurera censureringar måste utvecklare:

1. Implementera callback-funktionen `redacted(reason:)`.

2. Läsa ut egenskapen `privacy`.

3. Tillhandahålla anpassade platshållarvyer.

Utvecklare kan också rendera en vy som ocensurerad med vymodifieraren `unredacted()`.

Istället för att markera enskilda vyer som integritetskänsliga, till exempel om en hel widgets innehåll är integritetskänsligt, kan utvecklaren lägga till dataskyddsfunktionen i ett widgettillägg. WidgetKit visar platshållare istället för widgetens innehåll tills användaren låser upp sin enhet så att den matchar den valda integritetsnivån. Utvecklare måste aktivera dataskyddsfunktionen för widgettillägget i Xcode och sedan ställa in behörigheten `Data Protection` på värdet som överensstämmer med den integritetsnivå de vill erbjuda:

- `NSFileProtectionComplete`
- `NSFileProtectionCompleteUnlessOpen`

WidgetKit gömmer innehållet i de här widgetarna när enheten är låst med lösenkod och visar en platshållare tills en användare autentiserar efter en omstart av enheten. De här iOS-widgetarna är dessutom inte tillgängliga som iPhone-widgetar på Mac.

## DriverKit-säkerhet för macOS

DriverKit är det ramverk som låter utvecklare skapa enhetsdrivrutiner som användaren installerar på sin dator. Drivrutiner som byggs med DriverKit körs i användarutrymmet, istället för som kärntillägg, och ökar därmed systemsäkerhet och stabilitet. Detta leder till enklare installation och ger ökad stabilitet och säkerhet i macOS.

Användaren hämtar bara appen (installerare krävs inte när systemtillägg eller DriverKit används) och tillägget aktiveras bara när det krävs. Dessa ersätter i många fall kärntillägg som kräver administratörsbehörigheter för att installeras i /System/Bibliotek eller /Bibliotek.

IT-administratörer som använder enhetsdrivrutiner, molnlagringslösningar, nätverkssystem och säkerhetsappar som kräver kärntillägg uppmuntras att byta till nyare versioner som bygger på systemtillägg. Dessa nyare versioner medför en kraftigt minskad risk för kernel panics på datorn och minskar även angreppsytan. Dessa nya tillägg körs i användarutrymmet, kräver inte specialbehörigheter för installation och tas automatiskt bort när paketappen flyttas till papperskorgen.

DriverKit-ramverket tillhandahåller C++-klasser för I/O-tjänster, enhetsmatchning, minnesbeskrivningar och sändköer. Det definierar också I/O-lämpliga typer för tal, samlingar, strängar och andra vanliga typer. Användaren använder dessa med familjespecifika drivrutinsramverk som USBDriverKit och HIDDriverKit. Använd systemtilläggsramverket till att installera och uppgradera en drivrutin.

# ReplayKit-säkerhet i iOS och iPadOS

ReplayKit är ett ramverk som gör det möjligt för utvecklare att lägga till funktioner för inspelning och direktsändningar i sina appar. Dessutom får användare möjlighet att kommentera sina inspelningar och sändningar via enhetens framåtvända kamera och mikrofon.

## Filminspelning

Flera säkerhetslager är inbyggda vid inspelningen av en film:

- *Tillståndsdialogruta:* Innan inspelningen börjar visar ReplayKit en varningsdialogruta där användaren blir ombedd att bekräfta sin avsikt att spela in skärmen och spela in med mikrofonen och kameran på framsidan. Varningen visas en gång per approcess och visas igen om appen lämnas i bakgrunden längre tid än 8 minuter.
- *Skärm- och ljudinsamling:* Skärm- och ljudinsamling sker via appens process i ReplayKit-bakgrundsappens replayd. Det är utformat för att säkerställa att det inspelade innehållet aldrig är tillgängligt för approcessen.
- *Skärm- och ljudinsamling i appar:* Detta gör det möjligt för en app att hämta video- och samplingsbuffertar, vilka skyddas av behörighetsdialogrutan.
- *Skapa och lagra film:* Filmfilen skrivs till en katalog som bara är tillgänglig för ReplayKit-undersystemen. Den är aldrig tillgänglig för några appar. Detta hjälper till att förhindra att inspelningar används av tredje part utan användarens godkännande.
- *Förhandsvisning och delning:* Användaren kan förhandsvisa och dela filmen med ett användargränssnitt som tillhandahålls av ReplayKit. Detta användargränssnitt presenteras vid sidan av processen via iOS-tilläggsinfrastrukturen och har tillgång till den skapade filmfilen.

## ReplayKit-sändning

Flera säkerhetslager är inbyggda vid sändningen av en film:

- *Skärm- och ljudinsamling:* Mekanismen för skärm- och ljudinsamling under sändning är identisk med filminspelning och sker i `replayd`.
- *Sändningstillägg:* För att tjänster från tredje part ska kunna delta i ReplayKit-sändningar måste de skapa två nya tillägg som konfigureras med slutpunkten com.apple.broadcast-services:
  - Ett användargränssnittstillägg där användaren kan ställa in sin sändning.
  - Ett överföringstillägg som hanterar överföring av video- och ljuddata till tjänstens back-end-servrar.

Arkitekturen hjälper till att säkerställa att värdappar inte har någon behörighet till det video- och ljudinnehåll som sänds. Endast ReplayKit och sändningstillägg från tredje part har tillgång till det.

- *Sändningsväljare:* Med sändningsväljaren kan användare starta systemsändningar direkt från en app med samma systemdefinierade användargränssnitt som är tillgängligt via Kontrollcenter. Användargränssnittet implementeras med en privat API och är ett tillägg som finns i ReplayKit-ramverket. Det ligger inte i samma process som värdappen.
- *Överföringstillägg:* Tillägget som sändningstjänster från tredje part implementerar för att hantera video- och ljudinnehåll under sändning använder okodade raw-samplingsbuffertar. I det här hanteringsläget serialiseras video- och ljuddata och skickas till överföringstillägget från tredje part i realtid via en direkt-XPC-anslutning. Videodata kodas genom att extrahera IOSurface-objektet från videosamplingsbufferten, säkert kodat som ett XPC-objekt. Data skickas sedan via XPC till tillägget från tredje part och avkodas säkert tillbaka till ett IOSurface-objekt.

# ARKit-säkerhet i iOS och iPadOS

ARKit är ett ramverk som gör det möjligt för utvecklare att skapa upplevelser med förstärkt verklighet i sina appar eller spel. Utvecklarna kan lägga till 2D- och 3D-element genom att använda kameran på framsidan eller på baksidan av en iOS- eller iPadOS-enhet.

Apple har utvecklat kameror med integritet i centrum och appar från tredje part måste få användarens tillstånd innan de får tillgång till kameran. I iOS och iPadOS kan appar som användaren låter få tillgång till kameran komma åt realtidsbilder från kamerorna på framsidan och på baksidan. Appar får inte använda kameran utan att tydligt indikera att kameran används.

Bilder och videor som har tagits med kameran kan innehålla annan information, till exempel var och när de togs och skärpedjupet. Om användarna inte vill att bilder och videor som har tagits med appen Kamera ska inkludera platsen kan de när som helst ange detta under Inställningar > Integritet > Platstjänster > Kamera. Om användarna inte vill att bilder och videor ska inkludera platsen när de delas kan de när som helst stänga av platsen i menyn Alternativ på delningsbladet.

För att bättre placera in användarens AR-upplevelse kan appar som använder ARKit använda världs- och ansiktsspårningsinformation från den andra kameran. Världsspårning använder algoritmer på användarens enhet till att bearbeta information från dessa sensorer för att avgöra användarens position i förhållande till en fysisk plats. Med världsspårning blir funktioner som Optisk kurs i Kartor möjliga.

# Säker enhetshantering

## Säker enhetshantering i översikt

iOS, iPadOS, macOS, tvOS och watchOS har stöd för flexibla säkerhetspolicyer och konfigurationer som är enkla att genomdriva och hantera. Med dem kan företag och organisationer skydda information och försäkra sig om att medarbetarna uppfyller företagets krav, även när de använder egna enheter – till exempel inom ramen för ett BYOD-program.

Organisationer kan använda MDM (Mobile Device Management)-ramverket som implementerats av en MDM-lösning till att genomdriva lösenkodskrav, konfigurera inställningar, begränsa funktioner och till och med fjärradera företagsdata på hanterade enheter. Det hjälper till att skydda företagsdata även om medarbetare använder sina privata enheter till att komma åt dessa data.

## Säkerhet för parkopplingsmodell för iPhone och iPad

iOS och iPadOS använder en parkopplingsmodell för att styra tillgången till en enhet från en värddator. Vid parkoppling upprättas en betrodd relation mellan en enhet och dess värd som bekräftas genom utväxling av publika nycklar. iOS och iPadOS använder också detta bevis på tillit för att möjliggöra extra funktioner, som datasynkronisering, mellan enheten och den anslutna värden. I iOS 9 eller senare gäller att tjänster:

- som kräver parkoppling inte går att starta förrän enheten har låsts upp av användaren
- startas endast om enheten nyligen har låsts upp
- kan kräva att enheten är upplåst för att starta (t.ex. bildsynkronisering).

Parkopplingsprocessen kräver att användaren låser upp enheten och godkänner förfrågan om parkoppling från värden. I iOS 9 eller senare måste användaren även ange sin lösenkod, varpå värden och enheten utväxlar 2 048-bitars publika RSA-nycklar och sparar dem. Värden får sedan en 256-bitars nyckel som kan låsa upp en deponerad nyckelsamling lagrad på enheten. De utväxlade nycklarna används till att starta en krypterad SSL-session som enheten kräver innan den skickar skyddade data till värden eller startar en tjänst (iTunes- eller Finder-synkronisering, filöverföringar, Xcode-utveckling och så vidare). För att använda den här krypterade sessionen till all kommunikation kräver enheten anslutning från en värd via Wi-Fi, så den måste ha parkopplats via USB tidigare. Parkoppling gör det också möjligt att använda flera olika diagnosfunktioner. Parkopplingsposter som inte har använts på mer än sex månader upphör att gälla i iOS 9. I iOS 11 och senare har den här tidsramen kortats till 30 dagar.

Vissa diagnostjänster, som com.apple.mobile.pcapd, fungerar endast via USB. Tjänsten com.apple.file_relay kräver dessutom att en Apple-signerad konfigurationsprofil är installerad. I iOS 11 eller senare kan Apple TV använda SRP-protokollet (Secure Remote Password) till att trådlöst upprätta en parkopplingsrelation.

Användaren kan rensa listan över betrodda värdar med alternativet Nollställ nätverk eller Nollställ plats och integritet.

# MDM (Mobile Device Management)

## MDM-säkerhet i översikt

Apples operativsystem har stöd för MDM (Mobile Device Management) som gör det möjligt för organisationer att säkert konfigurera och hanterade skalade driftsättningar av Apple-enheter.

### Hur MDM fungerar på ett säkert sätt

MDM-funktionerna bygger på operativsystemteknik som konfigurationer, trådlös registrering och Apples tjänst för pushnotiser (APNs). Exempelvis används APNs till att väcka enheten och trigga den så att den kommunicerar direkt med MDM-lösningen via en säker anslutning. Ingen konfidentiell eller företagsägd information överförs via APNs.

Med hjälp av MDM kan IT-avdelningar registrera Apple-enheter i en företags- eller utbildningsmiljö, konfigurera och uppdatera inställningar trådlöst, övervaka att policyer efterlevs, hantera programuppdateringar och till och med fjärrlåsa eller fjärradera hanterade enheter.

I iOS 13, iPadOS 13.1 och macOS 10.15 eller senare får Apple-enheter stöd för ett nytt alternativ för användarregistrering som är specifikt utformat för BYOD (bring your own device)-program. Användarregistrering ger användarna större möjlighet att styra över sina egna enheter, samtidigt som säkerheten för företagsdata ökar genom att hanterade data separeras kryptografiskt. Detta ger en bättre balans mellan säkerhet, integritet och användarupplevelse i BYOD-program. En liknande mekanism för dataseparering har lagts till för kontodriven enhetsregistrering i iOS 17, iPadOS 17 och macOS 14 eller senare.

## Registreringstyper

- *Användarregistrering:* Användarregistreringen är utformad för enheter som ägs av användaren och är integrerad med hanterade Apple-ID:n för att skapa en användaridentitet på enheten. Hanterade Apple-ID:n krävs för att inleda registreringen och användare måste autentisera innan registreringen slutförs. Hanterade Apple-ID:n kan användas tillsammans med ett personligt Apple-ID som användaren redan har loggat in med. Hanterade appar och konton använder ett hanterat Apple-ID och personliga appar och konton använder ett personligt Apple-ID.
- *Enhetsregistrering:* Med enhetsregistrering kan organisationer låta användarna registrera enheter manuellt och sedan hantera många olika aspekter av enhetsanvändningen, inklusive möjligheten att radera enheten. Enhetsregistrering har också en större uppsättning konfigurationer och begränsningar som kan användas på enheten. När en användare tar bort en registreringsprofil blir även alla konfigurationer, inställningar och hanterade appar som är kopplade till den registreringsprofilen borttagna. I likhet med användarregistrering kan även enhetsregistrering integreras med ett hanterat Apple-ID. Den kontodrivna enhetsregistreringen gör det också möjligt att använda ett hanterat Apple-ID tillsammans med ett personligt Apple-ID och separerar företagsdata kryptografiskt.
- *Automatisk enhetsregistrering:* Med automatisk enhetsregistrering kan organisationer konfigurera och hantera enheter från det ögonblick de plockas upp ur förpackningen. De här enheterna kallas för *övervakade* och organisationen kan välja att förhindra att enhetsanvändaren kan ta bort MDM-profilen. Automatisk enhetsregistrering är skapad för enheter som ägs av organisationen.

## Enhetsbegränsningar

Begränsningar kan aktiveras – eller i vissa fall avaktiveras – för att förhindra att användare kommer åt en specifik app, tjänst eller funktion på en iPhone, iPad, Mac, Apple TV eller Apple Watch som har registrerats i en MDM-lösning. Begränsningar skickas till enheter i en begränsningsnyttolast som är en del av en konfiguration. En del begränsningar på en iPhone kan speglas på en parkopplad Apple Watch.

## Hantering av inställningar för lösenkoder och lösenord

Den förvalda inställningen är att användaren får ange en numerisk PIN-kod som lösenkod i iOS, iPadOS och watchOS. På iPhone- och iPad-enheter med Face ID eller Touch ID är den förvalda lösenkodslängden sex siffror och den kortast möjliga är fyra siffror. Längre och mer komplexa lösenkoder är svårare att gissa eller knäcka och rekommenderas därför.

Administratörer kan genomdriva komplexa lösenkodskrav och andra policyer med MDM eller via Microsoft Exchange i iOS och iPadOS. Ett administratörslösenord krävs vid manuell installation av nyttolasten för macOS-lösenkodspolicyn. Lösenkodspolicyer kan kräva en särskild lösenkodslängd, sammansättning eller andra attribut.

Apple Watch använder numeriska lösenkoder som förval. Om en lösenkodspolicy som används för en hanterad Apple Watch kräver att icke-numeriska tecken används måste dess parkopplade iPhone användas till att låsa upp enheten.

## Konfigurationskrav

Det är främst genom konfigurationer som en MDM-lösning levererar och hanterar policyer och begränsningar på hanterade enheter. Om organisationen behöver konfigurera ett stort antal enheter – eller tillhandahålla många anpassade mejlinställningar, nätverksinställningar eller certifikat till ett stort antal enheter – är konfigurationer ett tryggt och säkert sätt att göra det.

### Konfigurationer

En *konfiguration* är en XML-profil eller json-formaterad fil som följer en viss struktur och består av nyttolaster som läser in inställningar och auktoriseringsinformation på Apple-enheter. Konfigurationer automatiserar konfigureringen av inställningar, konton, begränsningar och behörigheter. De här filerna kan skapas med en MDM-lösning eller Apple Configurator för Mac eller skapas manuellt. Innan organisationen skickar en konfiguration till en Apple-enhet måste enheten registreras i MDM-lösningen med en registreringsprofil.

*Obs!* Apple Configurator för Mac kan endast användas till att hantera konfigurationsprofiler på iPhone-, iPad- och Apple TV-enheter.

### Registreringsprofiler

En *registreringsprofil* är en konfiguration med en MDM-nyttolast som registrerar enheten i den MDM-lösning som har angetts för just den enheten. MDM-lösningen kan då skicka kommandon och konfigurationer till enheten och göra förfrågningar vad gäller vissa aspekter av enheten. När en användare tar bort en registreringsprofil blir alla konfigurationer, deras inställningar och, beroende på registreringstypen och använd konfiguration, även hanterade appar som är kopplade till den registreringsprofilen borttagna. Det kan bara finnas en registreringsprofil i taget på en och samma enhet.

### Exempelkonfigurationer

En konfiguration innehåller ett antal inställningar i specifika nyttolaster som kan anges, inklusive (men inte begränsat till):

- Policyer för lösenkoder och lösenord
- Begränsningar för enhetsfunktioner (t.ex. avaktivering av kameran)
- Nätverks- och VPN-inställningar
- Microsoft Exchange-inställningar
- Mail-inställningar
- Passpoint-inställningar
- Inställningar för LDAP-katalogtjänster
- Inställningar för CalDAV-kalendertjänster
- Behörigheter och identiteter
- Certifikat
- Programuppdateringar

### Profilsignering och kryptering

Konfigurationsprofiler kan signeras för att validera deras ursprung och krypteras för att garantera deras integritet och skydda deras innehåll. Konfigurationsprofiler för iOS och iPadOS krypteras med den CMS (Cryptographic Message Syntax) som är angiven i RFC 5652 och har stöd för 3DES och AES128.

### Profilinstallation

Konfigurationer kan installeras på enheter via en MDM-lösning eller manuellt av användare. Apple Configurator för Mac kan också användas till att driftsätta konfigurationer på iOS-, iPadOS- och tvOS-enheter. Vissa konfigurationer kräver att de installeras via en MDM-lösning. Information om hur du tar bort profiler finns i Introduktion till MDM i Apple och driftsättning.

*Obs!* På övervakade enheter kan konfigurationsprofiler också låsas till en enhet. Det här är utformat för att de inte ska kunna tas bort eller endast kunna tas bort med en lösenkod.

## Automatisk enhetsregistrering

Organisationer kan automatiskt registrera iOS-, iPadOS-, macOS- och tvOS-enheter i en MDM-lösning utan att behöva förbereda eller ens röra vid själva enheterna innan användarna får dem. När organisationen har gått med i en av tjänsterna Apple School Manager, Apple Business Manager eller Apple Business Essentials kan administratörer logga in på tjänstens webbplats och länka programmet till den egna MDM-lösningen. Enheterna de har köpt kan sedan tilldelas till användarna via MDM. Under enhetskonfigurationen skickar enheten en förfrågan till Apples servrar om en tilldelad MDM och om en sådan finns skickas en signal till MDM-lösningen så att den utför registreringen. Med automatisk enhetsregistrering och en kompatibel MDM-lösning kan organisationer implementera följande säkerhetsåtgärder:

- Låt användare autentisera som en del av det inledande inställningsflödet i Apple-enhetens inställningsassistent under aktivering.
- Tillhandahåll en preliminär konfiguration med begränsad tillgång och kräv ytterligare konfiguration av enheten innan den får tillgång till känsliga data.
- Kräv att enheter kör en lägsta version av operativsystemet före registrering.
- Genomdriv FileVault-aktivering på Mac-datorer.

När en enhet har registrerats med MDM blir alla eventuella konfigurationer, begränsningar eller inställningar automatiskt installerade.

Installationsprocessen för användarna kan förenklas ytterligare genom att ta bort vissa steg i inställningsassistenten för enheterna så att användarna kommer igång snabbt. Om steg hoppas över används den mer integritetsbevarande inställningen. Om till exempel panelen för att konfigurera platstjänster hoppas över blir tjänsten inte aktiverad av inställningsassistenten.

Administratörer kan också bestämma om användarna ska kunna ta bort MDM-profilen från enheten eller inte och se till att konfigurationer och begränsningar används under hela den tid som enheten är i bruk.

### Apple School Manager, Apple Business Manager och Apple Business Essentials

Apple School Manager, Apple Business Manager och Apple Business Essentials är tjänster som IT-administratörer använder till att driftsätta Apple-enheter som en organisation har köpt direkt från Apple eller från auktoriserade Apple-återförsäljare och operatörer som deltar.

När de används med en MDM-lösning kan administratörer göra inställningsprocessen enklare för användare, konfigurera enhetsinställningar och distribuera appar och böcker som har köpts i de här tre tjänsterna. Apple School Manager integrerar dessutom med elevinformationsystem, antingen direkt eller med SFTP, och alla tre tjänsterna stöder katalogsynkronisering och federerad autentisering, så att konton automatiskt kan aktiveras, uppdateras och avaktiveras baserat på en organisations identitetsleverantör.

Apple upprätthåller certifieringar i enlighet med standarderna ISO/IEC 27001 och 27018 för att göra det möjligt för Apples kunder att efterleva sina skyldigheter enligt föreskrifter och avtal. Dessa certifieringar ger våra kunder tillgång till ett oberoende intyg på Apples praxis vad gäller informationsintegritet och säkerhet för de system som omfattas av certifieringarna. Mer information finns i Apples säkerhetscertifieringar för internettjänster i Apple och driftsättning.

*Obs!* Om du vill veta om ett Apple-program är tillgängligt i ett visst land eller en viss region läser du Apple Support-artikeln Tillgänglighet för Apple-program och betalningsmetoder för utbildnings- och företagskunder.

### Enhetsövervakning

*Övervakning* innebär normalt att enheten ägs av organisationen, vilket ger organisationen större kontroll över enhetens konfiguration och begränsningar. Mer information finns i Om Apple-enhetsövervakning i Apple och driftsättning.

Övervakning aktiveras automatiskt på en enhet när automatisk enhetsregistrering används.

## Säkerhet för Aktiveringslås

Hur Apple driver igenom Aktiveringslås beror på om enheten är en iPhone eller en iPad, en Mac med Apple Silicon eller en Intel-baserad Mac med Apple T2-säkerhetskretsen.

### Beteende på iPhone och iPad

På iPhone- och iPad-enheter drivs Aktiveringslås igenom under aktiveringsprocessen efter skärmen för val av Wi-Fi i inställningsassistenten för iOS och iPadOS. När enheten indikerar att det aktiveras skickas en begäran om att få ett aktiveringscertifikat till en Apple-server. Enheter med Aktiveringslås aktiverat uppmanar användaren att ange iCloud-inloggningsuppgifterna för den användare som aktiverade Aktiveringslås. Inställningsassistenten för iOS och iPadOS går inte vidare om inte ett giltigt certifikat erhålls.

### Beteende på Mac-datorer med Apple Silicon

På Mac-datorer med Apple Silicon verifierar LLB att det finns en giltig LocalPolicy-policy för enheten och att dess anti-replay-värden stämmer överens med de värden som lagras i SSC. LLB (Low-Level Bootloader) startar i recoveryOS om:

- Det inte finns någon LocalPolicy för aktuellt macOS.
- LocalPolicy är giltig för aktuellt macOS.
- LocalPolicys anti-replay-värden inte stämmer överens med de hash-värden som lagras i SSC.

recoveryOS märker att datorn inte är aktiverad och kontaktar aktiveringsservern för att få ett aktiveringscertifikat. Om enheten är låst med Aktiveringslås uppmanar recoveryOS användaren att ange iCloud-inloggningsuppgifterna för användaren som aktiverade Aktiveringslås. När ett giltigt aktiveringscertifikat har erhållits används nyckeln för det aktiveringscertifikatet till att få ett RemotePolicy-certifikat. Mac-datorn använder LocalPolicy-nyckeln och RemotePolicy-certifikatet till att skapa en giltig LocalPolicy. LLB tillåter inte start i macOS om det inte finns någon giltig LocalPolicy.

### Beteende på Intel-baserade Mac-datorer

I Intel-baserade Mac-datorer med T2-krets verifierar T2-kretsens fasta programvara att det finns ett giltigt aktiveringscertifikat innan datorn tillåts att starta i macOS. Den fasta UEFI-programvaran som läses in av T2-kretsen ansvarar för att skicka en förfrågan om aktiveringsstatus för enheten från T2-kretsen och starta i recoveryOS i stället för macOS om det inte finns något giltigt aktiveringscertifikat. recoveryOS märker att datorn inte är aktiverad och kontaktar aktiveringsservern för att få ett aktiveringscertifikat.
Om enheten är låst med Aktiveringslås uppmanar recoveryOS användaren att ange iCloud-inloggningsuppgifterna för användaren som aktiverade Aktiveringslås.
Den fasta UEFI-programvaran tillåter inte start i macOS om det inte finns något giltigt aktiveringscertifikat.

## Hanterat förlorat läge och fjärradering

Hanterat förlorat läge används till att hitta övervakade enheter som har stulits. När de har återfunnits kan de fjärrlåsas eller fjärraderas.

### Hanterat förlorat läge

Om en övervakad iOS- eller iPadOS-enhet med iOS 9 eller senare blir stulen eller tappas bort kan en MDM-administratör fjärraktivera förlorat läge (så kallat Hanterat förlorat läge) på den enheten. När Hanterat förlorat läge är aktiverat loggas den aktuella användaren ut och enheten kan inte låsas upp. På skärmen visas ett meddelande som administratören kan anpassa. Det kan t.ex. visa ett telefonnummer att ringa till om enheten hittas. Administratören kan också begära att enheten skickar information om sin aktuella plats (även om Platstjänster är avstängt) och välja att spela upp ett ljud. När en administratör stänger av Hanterat förlorat läge, som är det enda sättet som läget kan avaktiveras på, får användaren veta detta genom att ett meddelande visas på låsskärmen eller en notis på hemskärmen.

### Fjärradering

iPhone, iPad, Mac, Apple TV och Apple Watch kan raderas på distans av en administratör eller användare så att alla data blir oläsbara.

När ett fjärraderingskommando skickas från MDM eller iCloud svarar enheten med en bekräftelse till MDM-lösningen och utför raderingen. Om fjärraderingen utförs via Microsoft Exchange ActiveSync stämmer enheten av med Microsoft Exchange-servern innan den utför raderingen.

Fjärradering är inte möjligt i följande fall:

- Med användarregistrering
- Via Microsoft Exchange ActiveSync när kontot har installerats med användarregistrering
- Via Microsoft Exchange ActiveSync om enheten övervakas

Användare kan också radera enheter som stöds som de har i sin ägo via Inställningar (iPhone och iPad) eller Systeminställningar (Mac). Som tidigare nämnts kan iPhone-, iPad- och Apple Watch-enheter ställas in på att raderas automatiskt efter en serie misslyckade lösenkodsförsök.

Direkt fjärradering är tillgänglig på Mac-datorer med Apple Silicon och Mac-datorer med Apples T2-säkerhetskrets eller om FileVault är påslaget. Direkt fjärradering uppnås genom säker borttagning av medienyckeln.

## Säkerhet för Delad iPad i iPadOS

Delad iPad är ett läge för flera användare som används vid driftsättning av iPad. Det gör det möjligt för användare att dela en iPad samtidigt som dokument och data för varje enskild användare hålls isär. Varje användare får en egen privat, reserverad lagringsplats som skapas på en APFS-volym som skyddas av användarens inloggningsuppgifter. Delad iPad kräver användning av ett hanterat Apple-ID som är utfärdat och ägt av organisationen.

Med Delad iPad kan en användare logga in på valfri organisationsägd enhet som är konfigurerad för användning av flera användare. Användarnas data delas upp i separata kataloger, var och en i sin egen dataskyddsdomän, som skyddas genom både UNIX-behörigheter och körning i sandlåda. I iPadOS 13.4 och senare kan användarna också logga in till en tillfällig session. När användaren loggar ut från en tillfällig session raderas användarens APFS-volym och det reserverade utrymmet återlämnas till systemet.

### Logga in på Delad iPad

Stöd finns för både inbyggda och federerade hanterade Apple-ID:n vid inloggning på Delad iPad. När ett federerat konto används första gången dirigeras användaren om till identitetsleverantörens inloggningsportal. Efter autentisering utfärdas en kortlivad åtkomsttoken för stödjande hanterade Apple-ID:n och inloggningsprocessen fortsätter på ett snarlikt sätt som för inbyggda hanterade Apple-ID:n. Efter inloggning ber inställningsassistenten för Delad iPad användaren att ange en lösenkod (behörighet) för att skydda alla lokala data på enheten och i fortsättningen autentisera på inloggningsskärmen. Precis som på en enhet som bara används av en person, där användaren loggar in en gång till sitt hanterade Apple-ID med sitt federerade konto och sedan låser upp enheten med sin lösenkod, loggar en användare med en Delad iPad in en gång med sitt federerade konto och använder därefter sin lösenkod.

När en användare loggar in utan federerad autentisering blir det hanterade Apple-ID:t autentiserat med hjälp av SRP-protokollet från Apples identitetstjänst. Om autentiseringen godkänns beviljas en kortlivad åtkomsttoken som är specifik för enheten. Om användaren har använt enheten tidigare har han eller hon redan ett lokalt användarkonto som låses upp på samma sätt.

Om användaren inte har använt enheten förut, eller om funktionen för tillfällig session används, lägger Delad iPad till ett nytt UNIX-användar-ID, en APFS-volym för lagring av användarens personliga data samt en lokal nyckelring. Eftersom lagring allokeras (reserveras) för användaren när APFS-volymen skapas kan utrymmet vara otillräckligt för att skapa en ny volym. Om så är fallet identifierar systemet en befintlig användare vars data har slutat synkroniseras till molnet och tar bort den användaren från enheten så att den nya användaren kan logga in. Om det osannolika inträffar att överföringen av data till molnet inte har slutförts för någon av de befintliga användarna kan ingen ny användare skapas. För att kunna logga in måste den nya användaren vänta tills en av användarnas data har synkroniserats färdigt eller låta en administratör tvångsradera ett befintligt användarkonto, vilket innebär en risk för förlust av data.

Om enheten inte är ansluten till internet (exempelvis om användaren inte har tillgång till någon Wi-Fi-anslutningspunkt) kan autentiseringen ske mot det lokala kontot under ett begränsat antal dagar. I sådana fall kan endast användare med redan befintliga lokala konton eller en tillfällig session logga in. När tidsgränsen har överskridits måste användarna autentisera via internet även om det redan finns ett lokalt konto.

När en användares lokala konto har låsts upp eller skapats (om det är fjärrautentiserat) kommer den kortlivade token som har utfärdats av Apples servrar att omvandlas till en iCloud-token som tillåter inloggning till iCloud. Efter detta återskapas användarens inställningar och dokument och data synkroniseras från iCloud.

Dokument och data lagras på iCloud allteftersom de skapas eller ändras när användarsessionen är aktiv och enheten är ansluten. Utöver detta finns det en funktion för synkronisering i bakgrunden som ser till att ändringar synkroniseras med iCloud, eller andra webbtjänster med NSURLSession-bakgrundssessioner, direkt när användaren loggar ut. När användarens bakgrundssynkronisering är klar blir användarens APFS-volym avlänkad och kan inte länkas in igen utan att användaren loggar in igen.

Tillfälliga sessioner synkroniserar inte data med iCloud och även om tillfälliga sessioner kan logga in till en synkroniseringstjänst från tredje part, t.ex. Box eller Google Drive, finns det ingen möjlighet att fortsätta synkronisera data när den tillfälliga sessionen avslutats.

## Logga ut från Delad iPad

När en användare loggar ut från Delad iPad blir den användarens användarnyckelsamling omedelbart låst och alla appar avslutas. För att snabba upp inloggningen för en ny användare pausas vissa vanliga utloggningsåtgärder tillfälligt av iPadOS och ett inloggningsfönster visas för den nya användaren. Om en användare loggar in under detta tidsintervall (ca 30 sekunder) slutför Delad iPad den pausade utloggningen som en del av inloggningen på det nya användarkontot. Om Delad iPad förblir overksam utlöser detta den pausade utloggningen. Under utloggningsfasen blir inloggningsfönstret omstartat som om en annan utloggning hade inträffat.

När en tillfällig session avslutas utför Delad iPad den fullständiga utloggningssekvensen och raderar omedelbart den tillfälliga sessionens APFS-volym.

# Apple Configurator-säkerhet

Apple Configurator för Mac har en flexibel, säker och enhetsfokuserad utformning så att administratörer snabbt och enkelt kan konfigurera enskilda eller dussintals iOS-, iPadOS- och tvOS-enheter som är anslutna till en Mac via USB (eller tvOS-enheter som är parkopplade via Bonjour) innan enheterna delas ut till användarna. Med Apple Configurator för Mac kan en administratör uppdatera programvara, installera appar och konfigurationsprofiler, byta namn på enheter och byta deras bakgrundsbilder, exportera enhetsinformation och dokument med mera.

Apple Configurator för Mac kan också återaktivera eller återskapa Mac-datorer med Apple Silicon och Mac-datorer med Apples T2-säkerhetskrets. När en Mac återaktiveras eller återskapas på det här sättet hämtas filen som innehåller de senaste mindre uppdateringarna för operativsystemen (macOS, recoveryOS för Apple Silicon eller sepOS för T2) säkert från Apples servrar och installeras direkt på datorn. Efter en återaktivering eller återskapning raderas filen från den Mac som kör Apple Configurator. Användaren kan aldrig granska eller använda den här filen utanför Apple Configurator.

Administratörer kan också välja att lägga till enheter i Apple School Manager, Apple Business Manager eller Apple Business Essentials med Apple Configurator för Mac eller Apple Configurator för iPhone, även om enheterna inte har köpts direkt från Apple, en auktoriserad Apple-återförsäljare eller en auktoriserad mobiloperatör. När en administratör ställer in en enhet som har registrerats manuellt beter den sig som alla andra enheter i en av de tjänsterna med obligatorisk övervakning och MDM-registrering. För enheter som inte har köpts direkt har användaren 30 dagar på sig att ta bort enheten från en av de tjänsterna, övervakning och MDM.

Organisationer kan också använda Apple Configurator för Mac till att aktivera iOS-, iPadOS- och tvOS-enheter som helt och hållet saknar internetanslutning genom att ansluta dem till en värd-Mac som har en internetanslutning medan enheterna ställs in. Administratörer kan återskapa, aktivera och förbereda enheter med de konfigurationer som behövs, inklusive appar, profiler och dokument, utan att någon gång behöva ansluta till varken Wi-Fi- eller mobilnätverk. Den här funktionen tillåter inte administratörer att förbigå eventuella befintliga Aktiveringslåskrav som normalt krävs vid obunden aktivering.

# Säkerhet för Skärmtid

Skärmtid är en inbyggd funktion för att se och hantera hur mycket tid vuxna och deras barn ägnar åt appar, webbplatser med mera. Det finns två typer av användare: vuxna och (hanterade) barn.

Även om Skärmtid inte är någon ny systemsäkerhetsfunktion är det viktigt att förstå hur den skyddar integriteten och säkerheten för de data som samlas in och delas mellan enheter. Skärmtid är tillgänglig i iOS 12 eller senare, iPadOS 13.1 eller senare, macOS 10.15 eller senare och en del funktioner i watchOS 6 eller senare.

I tabellen nedan beskrivs huvudfunktionerna i Skärmtid.

| Funktion | Operativsystem som stöds |
|---|---|
| Visa användningsinformation. | iOS<br>iPadOS<br>macOS |
| Genomdriva ytterligare begränsningar. | iOS<br>iPadOS<br>macOS<br>watchOS |
| Ställa in begränsningar för webbanvändning. | iOS<br>iPadOS<br>macOS |
| Ställa in appbegränsningar. | iOS<br>iPadOS<br>macOS<br>watchOS |
| Konfigurera Skärmfri tid. | iOS<br>iPadOS<br>macOS<br>watchOS |

Användare som hanterar sin egen enhetsanvändning kan synkronisera begränsningar och användningsdata för Skärmtid mellan enheter som är associerade till samma iCloud-konto med CloudKits heltäckande kryptering. Det förutsätter att användarens konto har tvåfaktorsautentisering aktiverad (synkronisering är på som förval). Skärmtid ersätter funktionen Begränsningar som finns i tidigare versioner av iOS och iPadOS, och funktionen Föräldrakontroll som finns i tidigare versioner av macOS.

I iOS 13 eller senare, iPadOS 13.1 eller senare och macOS 10.15 eller senare delar Skärmtid-användare och hanterade barn automatiskt sin användning mellan enheter om tvåfaktorsautentisering är aktiverat för deras iCloud-konto. När en användare rensar historiken i Safari eller raderar en app tas motsvarande användningsdata bort från enheten och alla synkroniserade enheter.

## Föräldrar och Skärmtid

Föräldrar kan även använda Skärmtid på iOS-, iPadOS- och macOS-enheter i syfte att förstå och ta kontroll över sina barns användning. En förälder som är en familjesamordnare (i iCloud-familjedelning) kan se användningsdata och hantera inställningar för Skärmtid för sina barn. Barn blir meddelade när föräldrar slår på Skärmtid och de kan också övervaka sin egen användning. När föräldrar slår på Skärmtid för sina barn kan föräldrarna ställa in en lösenkod så att barnen inte kan göra ändringar. När barn blir myndiga (åldern varierar beroende på land eller region) kan de själva stänga av den här övervakningen.

Användningsdata och konfigurationsinställningar överförs mellan förälderns och barnets enheter via ett heltäckande krypterat IDS-protokoll (Apple Identity Service). Krypterade data kan tillfälligt lagras på IDS-servrar tills de kan läsas av den mottagande enheten (exempelvis så fort en avslagen iPhone eller iPad slås på igen). Dessa data kan inte läsas av Apple.

## Skärmtidsanalys

Om användaren slår på Dela iPhone- och Watch-analys blir endast följande anonymiserade data insamlade så att Apple får en bättre förståelse för hur Skärmtid används:

- Om Skärmtid aktiverades i inställningsassistenten eller senare i Inställningar
- Ändring i kategorianvändning efter att ha ställt in en begränsning för kategorin (inom 90 dagar)
- Om Skärmtid är på
- Om Skärmfri tid är aktiverad
- Antal gånger som förfrågan Be om mer tid användes
- Antal appbegränsningar
- Antal gånger användare visar användningen i Skärmtid-inställningarna per användartyp och per visningstyp (lokalt, fjärr, widget)
- Antal gånger användare ignorerar en gräns per användartyp
- Antal gånger användare raderar en gräns per användartyp

Apple samlar inte in några specifika app- eller användningsdata. När en användare ser en lista med appar i Skärmtids användningsinformation hämtas appsymbolerna direkt från App Store som inte behåller några data gällande de här förfrågningarna.

# Ordlista

**AES (Advanced Encryption Standard)** En populär global krypteringsstandard som används till att kryptera data för att hålla dem privata.

**AES-krypteringsmotor** En dedikerad maskinvarukomponent som implementerar AES.

**AES-XTS** Ett läge för AES definierat i IEEE 1619-2007 och avsett att användas vid kryptering av lagringsmedier.

**APFS (Apple File System)** Det förvalda filsystemet för iOS, iPadOS, tvOS, watchOS och Mac-datorer med macOS 10.13 eller senare. APFS har stark kryptering, utrymmesdelning, ögonblicksbilder, snabb beräkning av katalogstorlek och förbättrade filsystemsgrunder.

**APNs (Apples tjänst för pushnotiser)** En världsomspännande tjänst från Apple som levererar pushnotiser till Apple-enheter.

**Apple Business Manager** En enkel, webbaserad portal för IT-administratörer som gör att organisationer snabbt och smidigt kan driftsätta Apple-enheter som organisationen har köpt direkt av Apple eller via auktoriserade Apple-återförsäljare och operatörer som deltar. De kan automatiskt registrera enheter i sin MDM-lösning utan att behöva förbereda eller ens röra vid själva enheterna innan användarna får dem.

**Apple School Manager** En enkel, webbaserad portal för IT-administratörer som gör att organisationer snabbt och smidigt kan driftsätta Apple-enheter som organisationen har köpt direkt av Apple eller via auktoriserade Apple-återförsäljare och operatörer som deltar. De kan automatiskt registrera enheter i sin MDM-lösning utan att behöva förbereda eller ens röra vid själva enheterna innan användarna får dem.

**Apples säkerhetsbelöning** En belöning som ges av Apple till personer som rapporterar en sårbarhet som påverkar de senast levererade operativsystemen och (om det är relevant) den senaste maskinvaran.

**ASLR (Address Space Layout Randomization)** En teknik som operativsystem använder för att göra det svårare att utnyttja eventuella buggar i programvaran. Genom att se till att minnesadresser och -förskjutningar är oförutsägbara förhindras skadlig kod från att hårdkoda dessa värden.

**auktorisering av systemprogramvara** En process som kombinerar kryptografiska nycklar inbyggda i maskinvaran med en webbtjänst för att kontrollera att endast legitim programvara från Apple, lämplig för de enheter som stöds, tillhandahålls och installeras vid uppgradering.

**Boot Camp** Ett Mac-verktyg som stöder installation av Microsoft Windows på Mac-datorer som stöds.

**Boot ROM (startminne)** Den första kod som körs av enhetens processor när den startar. Eftersom den är integrerad i processorn kan den inte ändras, varken av Apple eller någon annan.

**BPR (Boot Progress Register)** En samling SoC-maskinvaruflaggor som programvara kan använda till att spåra de startlägen som enheten befinner sig i, som DFU-läge (Device Firmware Update) och återställningsläge. När en BPR-flagga (Boot Progress Register) anges kan den inte rensas efteråt. Det här gör det möjligt för senare programvaror att få en tillförlitlig indikator för systemets status.

**CKRecord** En ordbok med nyckelvärdepar som innehåller data som har sparats till eller hämtats från CloudKit.

**Dataskydd** En mekanism för skydd av filer och nyckelringar på Apple-enheter som stöds. Det kan också syfta på de API:er som appar använder till att skydda filer och nyckelringsobjekt.

**Datavalv** En mekanism – driven av kärnan – som ger skydd mot obehörig tillgång till data vare sig appen som skickar begäran själv körs i en sandlåda eller inte.

**DFU-läge (Device Firmware Upgrade)** Ett läge där enhetens Boot ROM-kod väntar på att återskapas via USB. Skärmen är svart i DFU-läge, men vid anslutning till en dator med iTunes eller Finder visas följande meddelande: ”Finder (eller iTunes) har upptäckt en (iPhone eller iPad) i återhämtningsläge. Användaren måste återställa denna (iPhone eller iPad) innan den kan användas med Finder (eller iTunes).”

**DMA (Direct Memory Access)** En funktion som tillåter att maskinvarudelsystem kommer åt huvudminnet direkt och förbigår processorn.

**ECDHE (Elliptic Curve Diffie-Hellman Exchange Ephemeral)** En mekanism för nyckelutbyte som baseras på elliptiska kurvor. Med ECDHE kan två parter komma överens om en hemlig nyckel på ett sätt som förhindrar att nyckeln upptäcks av någon som avlyssnar meddelandena mellan de båda parterna.

**ECDSA (Elliptic Curve Digital Signature Algorithm)** En digital signeringsalgoritm som baseras på elliptisk kurvkryptografi.

**ECID (Exclusive Chip Identification)** En 64-bitars identifierare som är unik för processorn i varje iPhone eller iPad.

**eSPI (Enhanced Serial Peripheral Interface)** En allt-i-ett-buss som är utformad för asynkron seriell kommunikation.

**Fast UEFI (Unified Extensible Firmware Interface)-programvara** En ersättningsteknik för BIOS för anslutning av fast programvara till en dators operativsystem.

**filnyckel** Den nyckel som används av dataskydd till att kryptera en fil i filsystemet. Filnyckeln paketeras av en klassnyckel och sparas i filens metadata.

**filsystemsnyckel** Den nyckel som krypterar varje fils metadata, inklusive dess klassnyckel. Den förvaras i det raderingsbara lagringsutrymmet för möjlighet till snabb radering snarare än konfidentialitet.

**Gatekeeper** En teknik i macOS som är utformad för att säkerställa att endast betrodd programvara körs på en användares Mac.

**grupp-ID (GID)** Som UID, men gemensamt för alla processorer i en klass.

**HMAC** En hashbaserad meddelandeautentiseringskod som bygger på en kryptografisk hashfunktion.

**HSM-modul (Hardware Security Module)** En specialiserad, manipuleringssäker komponent som skyddar och hanterar digitala nycklar.

**iBoot** Steg 2-startinläsare för alla Apple-enheter. Kod som läser in XNU som en del av den säkra startsekvensen. Beroende på SoC (System on Chip)-generation kan iBoot läsas in av LLB (Low-Level Bootloader) eller direkt av startminnet (Boot ROM).

**IDS (Apple Identity Service)** Apples katalog över publika iMessage-nycklar, APNs-adresser samt telefonnummer och e-postadresser som används till att söka efter nycklar och enhetsadresser.

**integrerad krets (IC)** Kallas också *mikrochip*.

**Integritetsskydd för systemcoprocessor (SCIP)** En mekanism Apple använder som är utformad för att förhindra ändring av coprocessorns fasta programvara.

**IOMMU (Input/Output Memory Management Unit)** En enhet för hantering av inmatning/utmatning och minne. Ett delsystem på en integrerad krets som styr åtkomsten till adressutrymme från andra I/O-enheter och tillbehör.

**JTAG (Joint Test Action Group)** Ett vanligt felsökningsverktyg för maskinvara som används av programmerare och kretsutvecklare.

**LLB (Low Level Bootloader)** I Mac-datorer med en startarkitektur i två steg innehåller LLB den kod som anropas av Boot ROM, och som i sin tur läser in iBoot, som en del av den säkra startsekvensen.

**Lösenkodshärledd nyckel (PDK)** Krypteringsnyckeln som härleds genom att användarens lösenord knyts till den långvariga SKP-nyckeln och UID:t för Secure Enclave

**MDM (Mobile Device Management)** En tjänst som administratören kan använda till att fjärrhantera registrerade enheter. När en enhet är registrerad kan administratören använda MDM-tjänsten via nätverket till att konfigurera inställningar och utföra andra åtgärder på enheten utan att enhetsanvändaren behöver göra något.

**medienyckel** En del av krypteringsnyckelhierarkin som bidrar till att utföra en säker och direkt radering. I iOS, iPadOS, tvOS och watchOS paketeras metadata på datavolymen med medienyckeln (utan den blir all åtkomst till enskilda filnycklar omöjlig, vilket innebär att filer som skyddas med dataskydd blir oåtkomliga). I macOS paketerar medienyckeln nyckelskapningsmaterialet, alla metadata och data på den FileVault-skyddade volymen. I båda fallen blir alla krypterade data oåtkomliga om medienyckeln raderas.

**minnesstyrenhet** Undersystemet i SoC som styr gränssnittet mellan SoC och dess huvudminne.

**NAND** Icke-flyktigt flash-minne.

**nyckelgenerering** Den process då en användares lösenkod görs om till en kryptografisk nyckel och förstärks med enhetens UID. Denna process gör att automatiserade intrångsförsök måste utföras på en given enhet, vilket begränsar hur ofta angreppen kan utföras och förhindrar att de utförs parallellt. Algoritmen för nyckelgenerering är PBKDF2. Den använder AES krypterad med enhetens UID som PRF-funktion (pseudorandom function) för varje iteration.

**nyckelpaketering** Kryptering av en nyckel med en annan. iOS och iPadOS använder NIST AES-nyckelpaketering i enlighet med RFC 3394.

**nyckelring** Den infrastruktur och den uppsättning API:er som används av operativsystem från Apple och tredjepartsappar till att lagra och hämta lösenord, nycklar och andra känsliga inloggningsuppgifter.

**nyckelsamling** En datastruktur som används till att lagra en samling klassnycklar. Varje typ (användare, enhet, system, säkerhetskopiering, deponering samt iCloud-säkerhetskopiering) har samma format.

En rubrik som innehåller: Version (angett till fyra i iOS 12 eller senare), Typ (system, säkerhetskopia, deponering eller iCloud-säkerhetskopiering), nyckelsamlingens UUID, ett HMAC om nyckelsamlingen är signerad och den metod som används för paketering av klassnycklarna – tillsammans med UID eller PBKDF2, eller tillsammans med salt och iterationsantal.

En lista över klassnycklar: Nyckelns UUID, Klass (dataskyddsklass för filer eller nyckelring), typ av paketering (endast UID-härledd nyckel, UID-härledd nyckel och lösenkodshärledd nyckel), paketerad klassnyckel och en publik nyckel för asymmetriska klasser.

**programvarustartbitar** Dedikerade bitar i Secure Enclaves AES-motor som bifogas i UID:t när nycklar genereras från UID:t. Varje programvarustartbit har en motsvarande låsbit. Secure Enclaves Boot ROM och operativsystem kan oberoende av varandra ändra värdet för varje programvarustartbit så länge som dess motsvarande låsbit inte har ställts in. När låsbiten har ställts in kan varken programvarustartbiten eller låsbiten ändras. Programvarustartbitar och deras lås nollställs när Secure Enclave startar om.

**Raderingsbart lagringsutrymme** Ett dedikerat NAND-lagringsutrymme för lagring av kryptografiska nycklar, som kan anropas direkt och raderas säkert. Det ger inget skydd mot angripare som har fysisk tillgång till enheten. Däremot kan nycklarna i det raderingsbara lagringsutrymmet användas i nyckelhierarkier för snabb radering och förebyggande säkerhet.

**ridge flow angle mapping** En matematisk representation av riktningen och bredden på de linjer som extraheras från en del av ett fingeravtryck.

**sepOS** Den fasta maskinvaran för Secure Enclave, baserad på en Apple-anpassad version av L4-mikrokärnan.

**SKP (Sealed Key Protection)** En teknik i dataskyddet som skyddar, eller *förseglar*, krypteringsnycklar med åtgärder i programvaran i systemet och nycklar som bara finns i maskinvaran (som UID:t för Secure Enclave).

**SoC (System on Chip)** En integrerad krets (IC) där flera komponenter är samlade på en krets. Approcessorn, Secure Enclave och andra coprocessorer är komponenter i SoC.

**SSC (Secure Storage Component)** En krets utformad med statisk RO-kod, en maskinvarubaserad slumptalsgenerator, kryptografimotorer och detektering av fysisk manipulering. På enheter som stöds parkopplas Secure Enclave med en integrerad krets för säker lagring av anti-replay-värde (SSC). Secure Enclave och lagringskretsen läser och uppdaterar alla anti-replay-värden genom ett säkert protokoll som endast ger tillgång till anti-replay-värden. Det finns flera generationer med den här tekniken med olika säkerhetsgarantier.

**SSD-styrenhet** Ett maskinvarudelsystem som hanterar lagringsmediet (solid-state drive).

**tillhandahållandeprofil** En egenskapslista (.plist-fil) som är signerad av Apple och innehåller en uppsättning entiteter och behörigheter som tillåter att appar installeras och testas på en iOS- eller iPadOS-enhet. En tillhandahållandeprofil för utveckling listar de enheter som utvecklaren har valt för specialdistribution, och en tillhandahållandeprofil för distribution innehåller app-ID:t för en företagsutvecklad app.

**UID (unikt ID)** En 256-bitars AES-nyckel som bränns in i processorn vid tillverkningen. Den kan inte läsas av fast programvara eller programvara och den används endast av processorns maskinvarubaserade AES-motor. För att komma åt den faktiska nyckeln måste en angripare utföra ett mycket avancerat och kostsamt fysiskt angrepp mot processorkretsen. UID:t har inget samband med någon annan identitetsmärkning på enheten, inklusive men inte begränsat till UDID:t.

**URI (Uniform Resource Identifier)** En teckensträng som identifierar en webbaserad resurs.

**xART** En förkortning av eXtended Anti-Replay Technology. En uppsättning tjänster som tillhandahåller krypterad, autentiserad bestående lagring för Secure Enclave med anti-replay-funktioner baserade på den fysiska lagringsarkitekturen. Se SSC (Secure Storage Component).

**XNU** Kärnan i operativsystemen från Apple. Den förutsätts vara betrodd och den kräver säkerhetsåtgärder som kodsignering, sandlåda, behörighetskontroller och ASLR (Address Space Layout Randomization).

**XProtect** En antivirusteknik i macOS för signaturbaserad upptäckt och borttagning av sabotageprogram.

**Återställningsläge** Ett läge som används till att återställa många Apple-enheter om användarens enhet inte kan identifieras så att användaren kan ominstallera operativsystemet.

# Dokumentets versionshistorik

## Dokumentets versionshistorik

### Maj 2024

Tillagda ämnen:

- Hash för Cryptex1 Image4-manifestet (spih)
- Cryptex1 Generation (stng)
- BlastDoor för Meddelanden och IDS
- Säkerhet med Låst läge
- Om App Store-säkerhet
- WidgetKit-säkerhet

Uppdaterade ämnen:

- Introduktion till Apple och säkerhetsteknik
- Säkerhet och Apples SoC:er
- Secure Enclave
- Face ID, Touch ID, lösenkoder och lösenord
- Säkerhet för ansiktsmatchning
- Användningsområden för Face ID och Touch ID
- Expresskort med strömsparläge
- Operativsystemets integritet
- Aktivera dataanslutningar säkert
- Verifiera tillbehör för iPhone och iPad
- Systemsäkerhet för watchOS
- Lösenkoder och lösenord
- Dataskydd i översikt
- Nyckelsamlingar för dataskydd
- Skydda nyckar i alternativa startlägen
- Skydda användardata i händelse av angrepp
- Hantera FileVault i macOS

- Introduktion till appsäkerhet för iOS och iPadOS
- Gatekeeper och säkerhet vid användning i macOS
- Säkerhet och hanterade Apple-ID:n
- iCloud-kryptering
- Säkerhet för kontakt för kontoåterställning
- Säkerhet för kontakt för digitalt arv
- Säkerhet för iCloud-nyckelring i översikt
- Säker synkronisering av nyckelringen
- Säkerhet vid deponering för iCloud-nyckelring
- Säkerhet för tillägg av kort i översikt
- Lägga till kredit- eller bankkort i Apple Pay
- Betala med kort via Apple Pay
- Säkerhet och Apple Card
- Säkerhet för Tap to Pay on iPhone
- Åtkomst med Plånbok
- Åtkomstnyckeltyper
- ID:n i Plånbok
- Säkerhet för ID:n i Plånbok
- Säkerhet och kit för utvecklare i översikt
- HomeKit-kommunikationssäkerhet
- MDM-säkerhet i översikt
- Konfigurationskrav

## December 2022

Tillagda ämnen:

- Avancerat dataskydd för iCloud

Uppdaterade ämnen:

- iCloud-säkerhet i översikt
- iCloud-kryptering
- Säkerhet och iCloud-säkerhetskopiering
- Säkerhet för kontakt för kontoåterställning
- Säkerhet för kontakt för digitalt arv

## Maj 2022

Uppdaterat för:

- iOS 15.4
- iPadOS 15.4
- macOS 12.3
- tvOS 15.4
- watchOS 8.5

Tillagda ämnen:

- Begränsningar för parkopplat recoveryOS
- Local Operating System Version (love)
- Hälsodelning
- Säkerhet för kontakt för kontoåterställning
- Säkerhet för kontakt för digitalt arv
- Säkerhet för Tap to Pay on iPhone
- Åtkomst med Plånbok
- Åtkomstnyckeltyper
- ID:n i Plånbok
- Siri-kompatibla HomeKit-tillbehör

Uppdaterade ämnen:

- Magic Keyboard med Touch ID
- Face ID, Touch ID, lösenkoder och lösenord
- Säkerhet för ansiktsmatchning
- Expresskort med strömsparläge
- Startlägen för Mac-datorer med Apple Silicon
- Innehåll i en LocalPolicy-fil för Mac-datorer med Apple Silicon
- Säkerhet för signerade systemvolymer
- Systemsäkerhet för watchOS
- Apple Security Research Device
- Apple File System
- Skydda apptillgång till användardata
- Introduktion till appsäkerhet för macOS
- Skydd mot sabotageprogram i macOS
- iCloud-säkerhet i översikt
- Säker synkronisering av nyckelringen
- Säker återställning av iCloud-nyckelring
- Betala med kort via Apple Pay

- Kontaktlösa kuponger i Apple Pay
- Göra kort oanvändbara med Apple Pay
- Apple Card-ansökan
- Säkerhet och Apple Cash
- Lägga till rese- och eMoney-kort i Plånbok
- Säkra Apple Messages for Business
- FaceTime-säkerhet
- Säkerhet för bilnycklar i iOS
- Apple Configurator-säkerhet

Borttagna ämnen:

- HomeKit-tillbehör och iCloud

## Maj 2021

Uppdaterat för:

- iOS 14.5
- iPadOS 14.5
- macOS 11.3
- tvOS 14.5
- watchOS 7.4

Tillagda ämnen:

- Magic Keyboard med Touch ID.
- Säker avsikt och säkra anslutningar till Secure Enclave
- Autoupplåsning och Apple Watch
- Hash för CustomOS Image4-manifestet (coih)

Uppdaterade ämnen:

- Två nya transaktioner i Expressläge lades till i Expresskort med strömsparläge.
- Funktionssammanfattning för Secure Enclave ändrades.
- Innehåll om programuppdatering lades till i Secure Multi-Boot (smb3).
- Ytterligare innehåll lades till om SKP (Sealed Key Protection).

## Februari 2021

Uppdaterat för:

- iOS 14.3
- iPadOS 14.3
- macOS 11.1
- tvOS 14.3
- watchOS 7.2

Tillagda ämnen:

- Minnessäker iBoot-implementering
- Startprocessen för Mac-datorer med Apple Silicon
- Startlägen för Mac-datorer med Apple Silicon
- Kontroll av säkerhetspolicyn för Startskiva för Mac-datorer med Apple Silicon
- Skapa och hantera signeringsnyckeln för LocalPolicy
- Innehåll i en LocalPolicy-fil för Mac-datorer med Apple Silicon
- Säkerhet för signerade systemvolymer
- Apple Security Research Device
- Lösenordsövervakning
- IPv6-säkerhet
- Säkerhet för bilnycklar i iOS

Uppdaterade ämnen:

- Secure Enclave
- Maskinvarubortkoppling av mikrofonen
- Miljöer för recoveryOS och diagnos för Intel-baserade Mac-datorer
- DMA-skydd i Mac-datorer
- Säker utökning av kärnan i macOS
- Systemintegritetsskydd
- Systemsäkerhet för watchOS
- Hantera FileVault i macOS
- Apptillgång till sparade lösenord
- Säkerhetsrekommendationer för lösenord
- Säkerhet och Apple Cash
- Säkra Apple Messages for Business
- Wi-Fi-integritet
- Säkerhet för Aktiveringslås
- Apple Configurator-säkerhet

## April 2020

Uppdaterat för:

- iOS 13.4
- iPadOS 13.4
- macOS 10.15.4
- tvOS 13.4
- watchOS 6.2

Uppdateringar:

- Bortkoppling av mikrofonen på iPad tillagt i Bortkoppling av mikrofonen.
- Datavalv har lagts till i Skydda apptillgång till användardata.
- Uppdateringar i Hantera FileVault i macOS och Kommandoradsverktyg.
- Tillägg under Borttagningsverktyg för sabotageprogram i Skydd mot sabotageprogram i macOS.
- Uppdateringar i Säkerhet för Delad iPad i iPadOS.

## December 2019

Sammanslagning av Säkerhetsguiden för iOS, macOS-säkerhet i översikt och Apple T2-säkerhetskretsen i översikt

Uppdaterat för:

- iOS 13.3
- iPadOS 13.3
- macOS 10.15.2
- tvOS 13.3
- watchOS 6.1.1

Integritetsinställningar, Siri och Siri-förslag samt Intelligent spårningsförebyggande i Safari har tagits bort. Se https://www.apple.com/se/privacy/ för den senaste informationen om de funktionerna.

## Maj 2019

Uppdaterat för iOS 12.3

- Stöd för TLS 1.3
- Reviderad beskrivning av AirDrop-säkerhet
- DFU-läge och återställningsläge
- Lösenkodskrav för tillbehörsanslutningar

## November 2018

Uppdaterat för iOS 12.1

- FaceTime-grupper

## September 2018

Uppdaterat för iOS 12 Secure Enclave

- OS-integritetsskydd
- Expresskort med strömsparläge
- DFU-läge och återställningsläge
- HomeKit TV-fjärrkontrollstillbehör
- Kontaktlösa kuponger
- Studentkort
- Siri-förslag
- Genvägar i Siri
- Appen Genvägar
- Hantering av användarlösenord
- Skärmtid
- Säkerhetscertifieringar och program

## Juli 2018

Uppdaterat för iOS 11.4

- Biometriska policyer
- HomeKit
- Apple Pay
- Business Chat
- Meddelanden på iCloud
- Apple Business Manager

## December 2017

Uppdaterat för iOS 11.2

- Apple Pay Cash

## Oktober 2017

Uppdaterat för iOS 11.1

- Säkerhetscertifieringar och program
- Touch ID/Face ID
- Delade anteckningar
- CloudKit med heltäckande kryptering
- TLS-uppdatering
- Apple Pay, betalning med Apple Pay på webben
- Siri-förslag
- Delad iPad

## Juli 2017

Uppdaterat för iOS 10.3

- Secure Enclave
- Dataskydd på filnivå
- Nyckelsamlingar
- Säkerhetscertifieringar och program
- SiriKit
- HealthKit
- Nätverkssäkerhet
- Bluetooth
- Delad iPad
- Förlorat läge
- Aktiveringslås
- Integritetsinställningar

## Mars 2017

Uppdaterat för iOS 10 Systemsäkerhet

- Dataskyddsklasser
- Säkerhetscertifieringar och program
- HomeKit, ReplayKit, SiriKit
- Apple Watch
- Wi-Fi, VPN
- Enkel inloggning
- Apple Pay, betalning med Apple Pay på webben
- Tillägg av kreditkort, bankkort och förbetalda kort
- Safari-förslag

## Maj 2016

Uppdaterat för iOS 9.3

- Hanterat Apple-ID
- Tvåfaktorsautentisering för Apple-ID
- Nyckelsamlingar
- Säkerhetscertifieringar
- Förlorat läge, aktiveringslås
- Säkra anteckningar
- Apple School Manager
- Delad iPad

## September 2015

Uppdaterat för iOS 9 Apple Watch och aktiveringslås

- Lösenkodspolicyer
- API-stöd för Touch ID
- Dataskyddet på A8 använder AES-XTS
- Nyckelsamlingar för obevakad programuppdatering
- Certifieringsuppdateringar
- Förtroendemodell för företagsappar
- Dataskydd för Safari-bokmärken
- App Transport Security
- VPN-specifikationer
- iCloud-fjärråtkomst för HomeKit
- Apple Pay-bonuskort, Apple Pay-kortutfärdares app
- Spotlight-indexering på enheten
- iOS-parkopplingsmodell
- Apple Configurator 2
- Begränsningar

# Copyright



Användning av Apple-logotypen på tangentbordet (alternativ-A) i kommersiella syften, utan Apples föregående skriftliga tillstånd, kan utgöra ett intrång i Apples varumärke och bryta mot upphovsrättslig lagstiftning i din jurisdiktion.

Apple, Apples logotyp, AirDrop, AirPlay, Apple Books, Apple Card, Apple Music, Apple Pay, Apple TV, Apple Wallet, Apple Watch, AppleScript, ARKit, Bonjour, Boot Camp, CarPlay, Face ID, FaceTime, FileVault, Finder, FireWire, Find My, Handoff, HealthKit, HomeKit, HomePod, HomePod mini, iMac, iMac Pro, iMessage, iPad, iPadOS, iPad Air, iPad Pro, iPhone, iTunes, Keychain, Lightning, Mac, Mac Catalyst, Mac mini, Mac Pro, MacBook, MacBook Air, MacBook Pro, macOS, Magic Keyboard, Objective-C, OS X, QuickType, Retina, Rosetta, Safari, Siri, Siri Remote, SiriKit, Swift, Spotlight, Touch ID, TrueDepth, tvOS, watchOS och Xcode är varumärken som tillhör Apple Inc. och är registrerade i USA och andra länder och regioner.

App Clips och Touch Bar är varumärken som tillhör Apple Inc.

App Store, AppleCare, CloudKit, iCloud, iCloud Drive, iCloud Keychain och iTunes Store är servicemärken som tillhör Apple Inc. och är registrerade i USA och andra länder och regioner.

Apple Messages for Business är ett servicemärke som tillhör Apple Inc.

Apple
One Apple Park Way
Cupertino, CA 95014
apple.com

IOS är ett varumärke eller registrerat varumärke som tillhör Cisco i USA och andra länder och används under licens.

Ordmärket Bluetooth och Bluetooth-logotyperna är varumärken som är registrerade och ägs av Bluetooth SIG, Inc. och Apple använder dessa under licens.

Java är ett registrerat varumärke som tillhör Oracle och/eller dess samarbetspartners.

UNIX är ett registrerat varumärke som tillhör The Open Group.

Namn på andra produkter och företag som nämns kan vara varumärken som tillhör respektive företag.

Informationen i handboken har kontrollerats för att vara korrekt. Apple ansvarar inte för tryck- eller korrekturfel. Information om produkter som inte tillverkas av Apple, eller fristående webbplatser som inte administreras eller testas av Apple, tillhandahålls utan att utgöra en rekommendation. Apple lämnar inga som helst garantier gällande urval, prestanda eller användning av webbplatser eller produkter från tredje part. Apple lämnar inga som helst utsagor gällande riktighet eller tillförlitlighet hos webbplatser från tredje part. Kontakta leverantören för ytterligare information.

En del appar är inte tillgängliga i alla områden. Apptillgängligheten kan ändras.

S028-00780