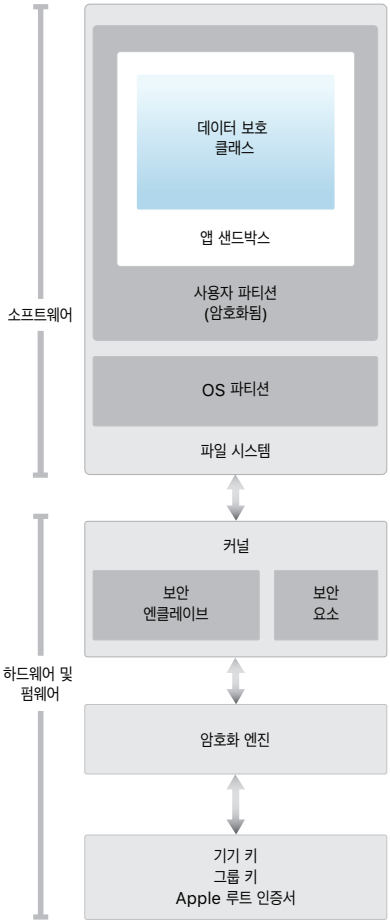
iOS 보안

iOS 11

2018년 1월

목차

소개



iOS의 보안 아키텍처 도표는 이 문서에서 논의되는 여러 가지 기술의 시각적인 개요를 제공합니다.

Apple은 보안을 핵심으로 iOS 플랫폼을 디자인했습니다. Apple은 최고의 모바일 플랫폼을 개발하기 위해 수십 년의 경험을 바탕으로 완전히 새로운 아키텍처를 구축할 수 있었습니다. 데스크탑 환경의 보안 취약점에 주의하여 iOS를 개발하였고 보안에 대한 새로운 접근 방법을 디자인했습니다. 또한 모바일 보안을 엄격하게 관리하고 기본적으로 전체 시스템을 보호할 수 있는 혁신적인 기능을 개발하고 통합하였습니다. 그로 인해 iOS는 모바일 기기 보안 분야에서 큰 발전을 이룰 수 있었습니다.

모든 iOS 기기는 소프트웨어, 하드웨어 및 서비스를 결합하여 보안성을 최대로 높이고 투명한 사용자 경험을 제공합니다. 또한 기기와 데이터만을 보호하는 것이 아니라 사용자의 작업, 네트워크, 그리고 주요 인터넷 서비스 모두를 포함하는 전체 생태계를 보호합니다.

iOS 및 iOS 기기는 고급 보안 기능을 제공하면서 사용하기에도 쉽습니다. 기본적으로 대부분의 보안 기능이 활성화되어 있기 때문에 IT 부서에서는 추가로 구성을 설정할 필요가 없습니다. 그리고 기기 암호화 같은 주요 보안 기능은 설정이 불가능하기 때문에 사용자가 실수로 비활성화할 우려도 없습니다. 또한 Face ID와 같은 기능으로 간단하고 직관적으로 기기를 보호할 수 있어 사용자 경험의 수준이 더욱 높아집니다.

이 도큐멘트에서는 보안 기술 및 기능이 iOS 플랫폼에 구현된 방법에 대한 자세한 정보를 제공합니다. 또한 이 도큐멘트는 조직에서 iOS 플랫폼 보안 기술 및 기능을 조직의 정책 및 절차와 통합하여 해당 조직의 특정 요구사항에 맞추는 데 도움을 줄 수 있습니다.

이 도큐멘트는 다음과 같은 주제로 구성되어 있습니다.

- **시스템 보안:** iPhone, iPad, iPod touch를 위한 플랫폼인 안전하고 통합된 소프트웨어 및 하드웨어
- **암호화 및 데이터 보호:** 기기 분실 또는 도난의 경우 혹은 인증받지 않은 사람이 기기를 사용 또는 수정하는 경우에 사용자 데이터를 보호하는 아키텍처와 디자인
- **앱 보안:** 앱이 플랫폼 무결성을 해치지 않으면서 안전하게 작동하도록 하는 시스템
- **네트워크 보안:** 전송 시 데이터 암호화와 보안 인증을 제공하는 업계 표준 네트워크 프로토콜
- **Apple Pay:** Apple이 구현한 안전한 결제 방식
- **인터넷 서비스:** 메시지, 동기화, 백업에 사용되는 Apple의 네트워크 기반 인프라
- **기기 제어:** iOS 기기 관리를 허용하고, 인증받지 않은 사람이 기기를 사용할 수 없도록 하며, 기기 분실 또는 도난의 경우 기기를 원격으로 삭제할 수 있는 방법
- **개인 정보 보호 제어:** 위치 서비스 및 사용자 데이터에 대한 접근 권한을 제어하는 iOS의 기술

시스템 보안

DFU(기기 펌웨어 업그레이드) 모드 들어가기
DFU 모드에 들어간 기기를 복원하면 수정되지 않은 상태의 Apple 서명 코드만 있는 정상 상태로 기기가 돌아갑니다. DFU 모드는 수동으로 들어갈 수 있습니다.

먼저 USB 케이블을 사용하여 기기를 컴퓨터에 연결합니다.

그런 다음, 기기 모델별로 다음을 수행합니다.

iPhone X, iPhone 8 또는 iPhone 8 Plus—음량 높이기 버튼을 빠르게 눌렀다 떼십시오. 음량 낮추기 버튼을 빠르게 눌렀다 떼십시오. 그런 다음, 복구 모드 화면이 표시될 때까지 측면 버튼을 길게 누르십시오.

iPhone 7 또는 iPhone 7 Plus—측면 버튼과 음량 낮추기 버튼을 동시에 길게 누르십시오. 복구 모드 화면이 표시될 때까지 누릅니다.

iPhone 6s 및 이전 모델, iPad 또는 iPod touch—홈 버튼과 상단(또는 측면) 버튼을 동시에 길게 누르십시오. 복구 모드 화면이 표시될 때까지 누릅니다.

참고: 기기가 DFU 모드인 동안 화면에는 아무 것도 표시되지 않습니다. 측면 버튼 또는 잠자기/깨우기 버튼을 너무 오래 누르면 Apple 로고가 나타나게 됩니다.

시스템 보안은 소프트웨어 및 하드웨어를 모든 iOS 기기의 주요 구성요소에서 안전하게 보호할 수 있도록 디자인되었습니다. 시스템 보안에는 부팅 프로세스, 소프트웨어 업데이트 및 보안 엔클레이브를 포함합니다. 이 시스템 보안 아키텍처는 iOS 보안의 중심이라고 할 수 있으며 기기의 사용성을 절대 방해하지 않습니다.

iOS 기기의 하드웨어, 소프트웨어 및 서비스 간의 높은 통합성으로 시스템의 각 구성요소를 항상 신뢰할 수 있으며 시스템을 전체로서 검증할 수 있습니다. 초기 부팅에서 iOS 소프트웨어 업데이트 그리고 타사 앱까지의 모든 단계를 분석하고 하드웨어 및 소프트웨어가 모두 최적으로 실행되고 있고 적절하게 리소스를 사용하고 있는지 점검할 수 있습니다.

보안 부팅 체인

시동 프로세스의 모든 단계에서는 무결성 확인을 위해 Apple이 암호화하여 서명한 구성 요소를 포함합니다. 또한 신뢰 체인을 확인한 이후에만 다음 단계를 진행할 수 있습니다. 신뢰 체인에는 부트로더, 커널, 커널 확장 프로그램 및 베이스밴드 펌웨어가 포함됩니다. 이러한 보안 부팅 체인을 통해 소프트웨어의 가장 낮은 단계에서 조작되는 것을 방지할 수 있습니다.

iOS 기기가 켜질 때 기기의 응용 프로그램 프로세서가 부트 ROM이라는 읽기 전용 메모리(ROM)에 저장된 코드를 즉시 실행합니다. 이 변경 불가능 코드(하드웨어 신뢰 루트라고 함)는 칩 제조 단계에서 저장되어 무조건 신뢰를 받습니다. 이 부트 ROM 코드에는 Apple 루트 CA 공개 키가 포함되어 있으며, 이 키를 사용하여 iBoot 부트로더를 로드하기 전에 Apple이 서명했는지를 확인합니다. 위의 단계가 신뢰 체인의 첫 번째 단계로서 이와 같이 각 단계에서 다음 단계의 Apple 서명을 확인합니다. iBoot가 자신의 작업을 완료하면 iOS 커널을 확인하고 실행합니다. S1, A9 또는 초기 버전의 A 시리즈 프로세서를 사용하는 기기에서는 부트 ROM에서 저레벨 부트로더(LLB) 단계를 추가로 로드하고 확인한 다음 iBoot를 로드하고 확인합니다.

부트 ROM에서 LLB(이전 기종) 또는 iBoot(새 기종)를 로드하지 못하는 경우, 기기는 DFU 모드로 진입합니다. LLB 또는 iBoot에서 다음 단계를 로드하거나 확인할 수 없는 경우, 시동이 멈추고 기기에 'iTunes에 연결' 화면이 표시됩니다. 이러한 상태를 복구 모드라고 부릅니다. 두 가지 상황 모두 USB를 통해 기기를 iTunes에 연결하고 초기 설정값으로 복원해야 합니다.

또한 셀룰러 접속이 가능한 기기에서는 서명된 소프트웨어 및 베이스밴드 프로세서가 확인한 키를 사용하는 방식으로 베이스밴드 하위 시스템이 보안 부팅과 비슷한 프로세스를 사용합니다.

그리고 보안 엔클레이브를 사용하는 기기에서는 보안 엔클레이브 보조 프로세서가 보안 부팅 프로세스를 사용해 Apple이 개별 소프트웨어를 확인하고 서명했는지 여부를 확인합니다. 이 도큐멘트의 '보안 엔클레이브' 섹션을 참조하십시오.

수동으로 복구 모드 들어가기에 대한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
support.apple.com/ko-kr/HT201263

시스템 소프트웨어 승인

Apple은 정기적으로 소프트웨어 업데이트를 출시하여 최근 보안 문제를 해결하고 새로운 기능도 제공합니다. 이러한 업데이트는 지원되는 모든 기기에 동시에 제공됩니다. 사용자가 기기 및 iTunes를 통해 iOS 업데이트 알림을 받고 최신 보안 수정 사항을 빠르게 적용할 수 있도록 업데이트가 무선으로 전송됩니다.

위에서 설명한 시동 프로세스를 통해 Apple이 서명한 코드만이 기기에 설치됩니다. iOS에서는 최신 보안 업데이트를 포함하지 않는 이전 버전으로 기기를 다운그레이드하는 것을 막기 위해 **시스템 소프트웨어 승인**이라는 프로세스를 사용합니다. 다운그레이드가 가능하다면 기기의 소유권을 획득한 해커가 이전 버전의 iOS를 설치해 새로운 버전에서 수정한 취약점을 악용할 수 있습니다.

또한 보안 엔클레이브를 사용하는 기기에서는 보안 엔클레이브 보조 프로세서가 시스템 소프트웨어 승인을 활용해 소프트웨어 무결성을 유지하고 다운그레이드 설치를 방지합니다. 이 도큐멘트의 '보안 엔클레이브' 섹션을 참조하십시오.

iTunes 또는 무선 전송(OTA)을 통해 iOS 소프트웨어 업데이트를 설치할 수 있습니다. iTunes에서는 전체 iOS 복사본을 다운로드하여 설치합니다. OTA 소프트웨어 업데이트는 OS 전체를 다운로드하지 않고 네트워크 효율성을 위해 업데이트 완료에 필요한 구성요소만을 다운로드합니다. 추가적으로 소프트웨어 업데이트는 macOS High Sierra가 설치되어 있고 콘텐츠 캐싱이 켜져 있는 Mac에 캐시되기 때문에 iOS 기기는 필요한 업데이트를 인터넷에서 다시 다운로드할 필요가 없습니다. 하지만 업데이트를 완료하기 위해서는 Apple 서버에 접속해야 합니다.

iOS를 업그레이드하는 동안 iTunes(OTA 소프트웨어 업데이트의 경우 기기 자체)는 Apple 설치 승인 서버와 연결되며 설치해야 할 설치 번들의 각 부분(예: iBoot, 커널, OS 이미지)의 암호화 측정값 목록, 재전송 방지 무작위 값(nonce) 및 기기 고유 ID(ECID)를 전송합니다.

승인 서버가 제공된 암호화 측정값 목록을 확인하여 허용된 설치 버전을 찾아 일치하는 버전을 찾으면 해당 암호화 측정값에 ECID를 추가하고 결과를 서명합니다. 그리고 업그레이드 프로세스의 일부로서 서버가 서명된 데이터 전체를 기기에 전송합니다. ECID를 추가하게 되면 업데이트를 요청하는 기기에 대한 승인이 '개인화'됩니다. 확인된 암호화 측정값만 서명하고 인증하여 Apple이 제공하는 방식대로 서버에서 업데이트를 진행합니다.

부팅 시의 신뢰 체인 평가를 통해 Apple의 서명을 확인하고 디스크에서 불러온 항목의 암호화 측정값 및 기기의 ECID가 서명에서 명시한 것과 일치하는지 확인합니다.

이러한 단계를 통해 특정 기기에서만 소프트웨어 업데이트가 승인되어 이전 버전의 iOS를 다른 기기로 복사할 수 없습니다. 또한 nonce는 해커가 서버 응답을 저장하지 못하도록 하여 기기 변조 또는 시스템 소프트웨어 수정을 방지합니다.

보안 엔클레이브

보안 엔클레이브는 Apple T1, Apple S2, Apple S3, Apple A7 또는 이상 버전의 A 시리즈 프로세서에 내장된 보조 프로세서입니다. 보안 엔클레이브는 암호화된 메모리를 사용하고 하드웨어 무작위 번호 생성기를 포함합니다. 또한 보안 엔클레이브는 데이터 보호 키 관리를 위한 모든 암호화 작업을 제공하며 커널이 손상된 경우에도 데이터 보호의 무결성을 유지합니다. 보안 엔클레이브와 응용 프로그램 프로세서 간의 통신은 Interrupt-Driven 방식 메일상자와 공유 메모리 데이터 버퍼로 구분되어 있습니다.

보안 엔클레이브는 Apple 환경에 맞추어 변경된 L4 마이크로커널을 운용합니다. Apple이 서명한 이 마이크로커널은 iOS 보안 부팅 체인의 일부로써 확인받고 개인화된 소프트웨어 업데이트 프로세스를 통해 업데이트됩니다.

기기가 시동될 때 임시 키가 생성되어 기기의 UID와 연결되고 이 키는 기기 메모리 공간의 보안 엔클레이브 부분을 암호화하는 데 사용됩니다. Apple A7을 제외하고 보안 엔클레이브의 메모리도 임시 키로 인증됩니다. Apple A11에서 무결성 트리는 보안에 있어서 매우 중요한 보안 엔클레이브 메모리의 재전송을 방지하기 위해 사용됩니다. 보안 엔클레이브 메모리는 임시 키 및 온칩 SRAM에 저장된 nonce를 통해 인증되었습니다.

추가적으로 보안 엔클레이브가 파일 시스템에 저장한 데이터는 UID와 함께 연결된 키와 재전송 방지 카운터로 암호화됩니다. 보안 엔클레이브의 재전송 방지 서비스는 재전송 방지 경계를 표시하는 이벤트를 통한 데이터 해지에 사용됩니다. 이러한 이벤트는 다음 항목을 포함하며 이에 국한되지는 않습니다.

- 암호 변경
- Touch ID 또는 Face ID 활성화/비활성화

- 지문 추가/삭제
- Face ID 재설정
- Apple Pay 카드 추가/삭제
- 모든 콘텐츠 및 설정 지우기

또한 보안 엔클레이브는 Touch ID 및 Face ID 센서의 지문 및 얼굴 데이터를 처리하여 일치하는 데이터가 있는지 판단하고 사용자를 대신해 접근 또는 구매를 활성화합니다.

Touch ID

Touch ID는 안전한 방법으로 iPhone 및 iPad에 빠르고 쉽게 접근할 수 있는 지문 인식 시스템입니다. Touch ID 기술은 모든 각도의 지문 데이터를 읽고 센서를 사용할 때마다 추가로 겹쳐지는 노드를 확인하여 지문 지도를 계속 확장시키는 방법으로 계속해서 사용자의 지문을 학습합니다.

Face ID

Face ID를 사용하면 iPhone X을 보기만 해도 안전하게 잠금 해제할 수 있습니다. Face ID는 고급 기술을 사용하는 TrueDepth 카메라 시스템을 통해 활성화되는 직관적이고 안전한 인증 방식을 제공하여, 사용자 얼굴의 기하학적 구조를 정확히 매핑합니다. Face ID는 사용자의 시선 방향을 감지하여 카메라를 주시하고 있는지 확인한 다음, 신경망을 통해 사용자가 일치하는지 스푸핑은 아닌지 가려내므로 사용자가 휴대폰을 보기만 해도 잠금 해제할 수 있습니다. Face ID는 사용자의 외모 변화에 자동으로 적응하며 사용자의 생체 데이터 보안 및 개인 정보를 안전하게 보호합니다.

Touch ID, Face ID 및 암호

Touch ID 또는 Face ID를 사용하려면 사용자는 먼저 기기에서 잠금 해제에 필요한 암호를 설정해야 합니다. Touch ID 또는 Face ID가 일치하는 데이터를 감지하면 기기는 암호를 묻지 않고 잠금을 해제합니다. 이 기능을 통해 사용자는 길고 복잡한 암호를 자주 입력할 필요가 없어지므로 안전한 암호를 현실적으로 사용할 수 있게 됩니다. Touch ID 및 Face ID는 암호를 대신할 수 없지만 안전한 공간이나 시간적 제약 내에서는 기기에 쉽게 접근할 수 있습니다. 강력한 암호는 iOS 기기의 암호화 보호의 토대가 되기 때문에 이 기능이 중요합니다.

사용자는 언제든지 Touch ID 또는 Face ID를 대신하여 암호를 사용할 수 있으며, 다음과 같은 상황에서는 암호가 요구됩니다.

- 기기가 방금 켜졌거나 재시동된 경우
- 기기가 48시간 이상 잠금 해제되지 않은 경우
- 156시간(6.5일) 동안 기기를 잠금 해제하는 데 암호를 사용하지 않고 4시간 동안 Face ID로 기기를 잠금 해제하지 않은 경우
- 기기가 원격으로 잠겨진 경우
- 데이터 일치 시도에 다섯 번 실패한 경우
- 전원 끄기/긴급 구조 요청을 실행한 경우

Touch ID 또는 Face ID가 활성화되어 있으면 기기의 측면 버튼을 누르는 즉시 잠기며, 기기가 잠자기 상태가 될 때마다 잠깁니다. 기기를 깨울 때마다 Touch ID 및 Face ID는 일치하는 데이터(아니면 암호 입력 선택 가능)를 요구합니다.

누군가가 사용자의 iPhone X을 보고 잠금 해제할 수 있는 확률은 약 1/1,000,000입니다 (Touch ID의 경우 1/50,000). 보안을 강화하기 위해 Touch ID 및 Face ID 데이터 일치 시도에 다섯 번 실패하면 기기 접근 시 암호가 요구됩니다. 사용자와 닮은 형제나 자매, 쌍둥이의 경우 Face ID를 사용한 오인식 확률이 다를 수 있습니다. 마찬가지로 13세 미만의 어린이의 경우에도 뚜렷한 얼굴 특징이 완전히 발달되지 않았을 수 있기 때문에 이 확률이 다를 수 있습니다. 이 문제가 우려되는 사용자의 경우, Apple에서는 암호를 통한 인증을 권장합니다.

Touch ID 보안

지문 센서는 홈 버튼을 둘러싸고 있는 정전식 강철 링이 손가락의 터치를 인식할 경우에만 작동합니다. 터치가 인식되면 고급 이미지 어레이가 작동되어 손가락을 스캔하고 스캔한 이미지를 보안 엔클레이브에 전송합니다. 프로세서와 Touch ID 센서 간의 통신은 SPI(직렬 주변기기 인터페이스) 버스 상에서 이루어집니다. 프로세서는 데이터를 보안 엔클레이브로 전달만 할 수 있고 읽을 수는 없습니다. 제조 과정 중에 각 Touch ID 센서 및 그에 해당하는 보안 엔클레이브용으로 권한이 설정된 공유 키를 사용하여 양도 받은 세션 키로 통신을 암호화하고 인증하였습니다. 공유 키는 강력하고 무작위로 설정되며 모든 Touch ID 센서마다 다릅니다. 세션 키 교환은 통신하는 양측에서 제공하는 무작위 키를 사용한 AES 키 래핑으로 이루어집니다. 무작위 키는 세션 키를 설정하고 AES-CCM 전송 암호화를 사용합니다.

스캔된 래스터 이미지는 보안 엔클레이브의 암호화된 메모리에 임시로 저장됩니다. 이미지는 벡터화되어 분석되고 분석이 끝난 뒤에는 삭제됩니다. 그 분석 단계에서는 피하 융선 흐름 각도 매핑을 사용합니다. 이 기술은 사용자의 실제 지문을 복원하는 데 필요한 데이터를 삭제하는 손실 프로세스입니다. 결과로 나온 노드 지도는 신원 정보를 전혀 포함하지 않으며 암호화된 포맷으로 저장됩니다. 이 포맷은 보안 엔클레이브에서만 읽을 수 있으며 절대로 Apple에 전송되지 않고 iCloud 또는 iTunes에도 백업되지 않습니다.

Face ID 보안

Face ID는 사용자가 카메라를 주시하고 있는지 확인하도록 설계되었고 낮은 오인식률의 탄탄한 인증 방식을 제공하며 디지털 스푸핑 및 물리적 스푸핑을 줄입니다.

TrueDepth 카메라는 사용자가 iPhone X을 들어 올리거나 iPhone X의 화면을 탭하여 깨울 때 자동으로 사용자의 얼굴을 탐색하며 iPhone X이 수신 알림을 표시하기 위해 인증을 시도하거나 Face ID 지원 앱에서 Face ID 인증을 요청하는 경우에도 자동으로 사용자의 얼굴을 탐색합니다. 얼굴이 감지되면 Face ID는 사용자가 눈을 뜨고 있는지와 기기의 정면을 보는지를 감지하여 사용자가 카메라를 보고 있으며 잠금 해제를 시도하는 것인지 확인합니다. 이 기능은 VoiceOver가 켜져 있는 경우 비활성화되며 필요한 경우 각 기능을 별개로 비활성화할 수 있습니다.

얼굴이 카메라를 향해 있음이 확인되면 TrueDepth 카메라는 30,000개가 넘는 적외선 점을 투사하고 인식하여 2D 자외선 이미지에 따라 얼굴의 심도 맵을 만듭니다. 이 데이터는 디지털 서명되고 보안 엔클레이브에 보내지는 연속된 2D 이미지 및 심도 맵을 생성하는 데 사용됩니다. 디지털 및 물리적 스푸핑에 대응하기 위해 TrueDepth 카메라는 2D 이미지의 순서를 무작위로 지정하며, 심도 맵은 기기 전용 무작위 패턴을 캡처하고 투사합니다. A11 Bionic 칩의 뉴럴 엔진(보안 엔클레이브 내에서 보호됨) 일부는 이 데이터를 수학적 표현으로 변형시켜서 등록된 얼굴 데이터와 비교합니다. 등록된 얼굴 데이터는 다양한 각도에서 캡처된 사용자 얼굴에 대한 수학적 표현 그 자체입니다.

얼굴 인식은 이를 목적으로 특수하게 훈련된 신경망을 사용한 보안 엔클레이브 내에서 수행됩니다. Apple은 참가자들의 사전 동의 하에 수행된 연구에서 수집한 IR 및 심도 이미지를 포함해 십억 장이 넘는 이미지를 사용하여 얼굴 인식 신경망을 개발했습니다. Apple은 전 세계에서 온 참가자들과 작업하여 성별, 나이, 인종 및 기타 요소를 대표하는 그룹을 구성했습니다. 이 연구는 다양한 사용자들에게 높은 수준의 정확성을 제공하기 위해 확장되었습니다. Face ID는 모자, 스카프, 안경, 콘텍트 렌즈 및 여러 종류의 선글라스를 구분하도록 설계되었습니다. 나아가 실내와 실외, 심지어는 완전한 어둠 속에서도 인식하도록 설계되었습니다. 스푸핑을 발견하고 대응하도록 훈련된 추가 신경망은 사진이나 마스크로 사용자의 iPhone X을 잠금 해제하려는 시도로부터 방어합니다.

사용자 얼굴의 수학적 표현을 포함하여 Face ID 데이터는 암호화되어 있으며 보안 엔클레이브만 이를 사용할 수 있습니다. 이 데이터는 절대 기기 밖으로 유출될 수 없습니다. Apple에 전송되지도 않으며 기기 백업에도 포함되지도 않습니다. 정상 작동 중에 다음의 Face ID 데이터가 저장되며, 보안 엔클레이브용으로만 암호화됩니다.

- 등록하는 동안 산출된 사용자 얼굴의 수학적 표현.
- Face ID에서 추후 얼굴 인식을 보완하는 데 유용할 것으로 판단한 경우, 잠금 해제를 시도하는 동안 산출된 사용자 얼굴의 수학적 표현.

정상 작동 중 캡처된 얼굴 이미지는 저장되지 않으며, 얼굴 데이터 등록 또는 등록된 Face ID 데이터와의 비교를 위한 수학적 표현이 산출되는 즉시 폐기됩니다.

Touch ID 또는 Face ID가 iOS 기기를 잠금 해제하는 방법

Touch ID 또는 Face ID가 비활성화된 상태에서 기기가 잠기게 되면 보안 엔클레이브에서 보관하던 최상위 데이터 보호 키가 폐기됩니다. 해당 클래스에 있는 파일 및 키체인 항목은 암호를 입력하여 기기를 잠금 해제해야만 접근할 수 있습니다.

하지만 Touch ID 또는 Face ID가 켜져 있는 경우에는 기기가 잠금 상태가 되더라도 해당 키가 삭제되지 않습니다. 대신에 보안 엔클레이브 내의 Touch ID 또는 Face ID 보조 시스템에 할당된 키가 해당 키를 래핑합니다. 사용자가 기기 잠금 해제를 시도할 때 기기가 일치하는 데이터를 감지하면 데이터 보호 키를 래핑 해제하기 위한 키가 제공되며 기기가 잠금 해제됩니다. 이 프로세스는 데이터 보호와 Touch ID 또는 Face ID 보조 시스템의 조합을 통해 기기를 잠금 해제하기 때문에 더욱 안전합니다.

기기를 재시동하면 Touch ID 또는 Face ID로 기기를 잠금 해제하는 데 사용했던 키가 사라집니다. 또한 암호 입력에 필요한 조건(예를 들어, 48시간 동안 잠금 해제되지 않은 경우 또는 인식 시도가 다섯 번 실패한 경우)이 충족되면 해당 키는 보안 엔클레이브에 의해 폐기됩니다.

잠금 해제 성능을 개선하고 자연스럽게 변화해 가는 사용자의 얼굴과 외모를 적용하기 위해 Face ID는 저장된 수학적 표현을 계속해서 보완합니다. 잠금 해제에 성공하면 Face ID는 데이터가 폐기되기 전에 제한된 추가 잠금 해제 횟수에 대해 새로 산출된 수학적 표현(품질이 충분히 좋은 경우)을 사용할 수 있습니다. 반대로 Face ID가 사용자 인식을 실패했지만 특정 기준값보다 인식 품질이 높고 인식에 실패한 뒤 즉시 암호를 입력한 경우 Face ID는 사용자 얼굴은 다시 캡처한 다음, 등록된 Face ID 데이터에 새로 산출된 수학적 표현을 보완합니다. 새 Face ID 데이터는 사용자가 얼굴 인식을 중단하고 제한된 잠금 해제 횟수를 채우면 폐기됩니다. 이러한 보완 작업은 오인식률은 최소화하면서도 Face ID가 사용자의 헤어 스타일이나 메이크업이 크게 변화하더라도 사용자를 인식할 수 있도록 합니다.

Touch ID, Face ID 및 Apple Pay

또한 사용자는 Touch ID, Face ID 및 Apple Pay를 사용하여 스토어, 앱, 웹에서 간편하고 안전하게 구매할 수 있습니다. 자세한 내용은 이 도큐멘트의 Touch ID 및 Apple Pay 섹션을 참조하십시오.

Face ID로 스토어 내 결제를 승인하려면 먼저 측면 버튼을 이중 클릭하여 결제 의사가 있다는 것을 표시해야 합니다. 그런 다음 iPhone X을 비접촉식 결제 리더기 가까이에 가져가기 전에 Face ID를 사용하여 인증합니다. Face ID 인증 후 다른 Apple Pay 결제 수단을 선택하려면 다시 인증해야 하지만 측면 버튼을 또 이중 클릭할 필요는 없습니다.

앱 및 웹에서 결제하려면 측면 버튼을 이중 클릭하여 결제 의사를 표시한 다음, Face ID로 인증하여 결제를 승인합니다. Apple Pay 거래가 측면 버튼을 클릭한 후 30초 이내에 완료되지 않으면 측면 버튼을 이중 클릭하여 구입 의사를 다시 표시해야 합니다.

Face ID 진단

Face ID 데이터는 기기 밖으로 유출될 수 없으며 절대 iCloud 또는 다른 곳에 백업되지 않습니다. 지원을 받기 위해 AppleCare에 Face ID 진단 데이터를 제공하려는 경우에만 이 정보가 기기에서 전송됩니다. Face ID 진단을 활성화하려면 소프트웨어 업데이트 개인화 프로세스에서 사용한 것과 비슷한 Apple이 디지털 서명한 인증이 필요합니다. 인증이 완료되면 사용자는 Face ID 진단을 활성화하고 iPhone X의 설정 앱 내에서 설정 프로세스를 시작할 수 있습니다.

Face ID 진단 설정의 한 과정으로써 기존에 등록된 Face ID는 삭제되며 Face ID 재등록 요청이 표시됩니다. 이후 10일간 사용자가 인증을 시도할 때 iPhone X은 캡처한 Face ID 이미지를 기록합니다. 그 기간이 지나면 iPhone X은 자동으로 이미지 저장을 중단합니다. Face ID 진단은 Apple에 자동으로 데이터를 전송하지 않습니다. Apple에 Face ID 진단 데이터가 전송되기 전에 사용자가 먼저 검토하고 승인할 수 있습니다. 진단 데이터에는 진단 모드에서 수집한 등록 및 잠금 해제 이미지(잠금 해제에 성공한 이미지와 실패한 이미지 모두)가 포함됩니다. Face ID 진단은 사용자가 승인한 Face ID 진단 이미지만 업로드하고 업로드되기 전에 데이터를 암호화하며 업로드가 완료되는 즉시 iPhone X에서 데이터를 삭제합니다. 사용자가 거부한 이미지는 즉시 삭제됩니다.

이미지를 검토하고 승인된 이미지를 업로드하는 Face ID 진단 세션을 완료하지 않으면 Face ID 진단은 40일 후 자동으로 종료되며 모든 진단 이미지는 iPhone X에서 삭제됩니다. 또한 사용자는 언제든지 Face ID 진단을 비활성화할 수 있습니다. Face ID 진단을 비활성화하면 모든 로컬 이미지가 즉시 삭제되고 이 경우에는 Apple에 Face ID 데이터가 공유되지 않습니다.

Touch ID 및 Face ID의 기타 사용

타사 앱은 시스템이 제공하는 API를 사용하여 사용자에게 Touch ID나 Face ID, 또는 암호로 인증하도록 요청할 수 있습니다. Touch ID를 지원하는 앱은 별다른 변경 사항 없이 Face ID를 자동으로 지원합니다. Touch ID 또는 Face ID를 사용할 때 해당 앱은 인증 성공 여부만을 전달받기 때문에 등록된 사용자의 Touch ID, Face ID 또는 데이터에 접근할 수 없습니다. 또한 Touch ID 또는 Face ID는 키체인 항목을 보호하여 데이터가 일치하거나 기기 암호가 입력된 경우에만 보안 엔클레이브가 항목을 해제할 수 있도록 합니다. 앱 개발자는 사용자에게 키체인 항목을 잠금 해제하기 위해 Touch ID나 Face ID, 또는 암호를 요구하기 전에 사용자가 암호를 설정했는지 확인하는 API를 보유하고 있습니다. 앱 개발자는 다음을 수행할 수 있습니다.

- 인증 API 작업이 앱 암호 또는 기기 암호로 대치되지 않도록 설정해야 합니다. 앱 개발자는 보안에 민감한 앱에서 2차 인증 요소로 Touch ID 또는 Face ID를 사용하도록 허용하는 사용자 등록 여부를 질문할 수 있습니다.
- Touch ID 또는 Face ID로 보호될 수 있는 보안 엔클레이브 내부 ECC 키를 생성하고 사용할 수 있습니다. 해당 키를 사용한 작업은 사용을 승인 받은 후에 보안 엔클레이브 내부에서 항상 수행됩니다.

Touch ID 또는 Face ID를 구성하여 iTunes Store, App Store 및 iBooks Store 구입을 승인할 수 있도록 설정하면 사용자가 Apple ID 암호를 입력할 필요가 없습니다. iOS 11 이상에서는 Touch ID 및 Face ID로 보호된 보안 엔클레이브 ECC 키를 사용하여 스토어에서 보낸 요청을 서명해 결제를 승인합니다.

암호화 및 데이터 보호

모든 콘텐츠 및 설정 지우기
설정에서 '모든 콘텐츠 및 설정 지우기' 옵션은 삭제할 수 있는 저장 장치(Effaceable Storage)의 모든 키를 삭제하여 기기의 모든 사용자 데이터가 암호화되어 접근할 수 없도록 합니다. 그러므로 기기를 다른 사람에게 제공하거나 서비스를 위해 반환하기 전에 기기에서 모든 개인 정보를 삭제하는 것이 가장 좋은 방법입니다.

중요사항: 삭제된 데이터를 복구할 방법이 없기 때문에 기기를 백업하기 전에는 '모든 콘텐츠 및 설정 지우기' 옵션을 사용하지 마십시오.

보안 부팅 체인, 코드 서명, 런타임 프로세스 보안은 모두 신뢰하는 코드 및 앱만 기기에서 작동할 수 있도록 합니다. iOS는 이러한 추가적인 암호화 및 데이터 보호 기능을 통해 일부 다른 보안 인프라 부분이 제대로 동작하지 않는 상황(예: 기기에 승인되지 않은 변경 사항이 생긴 경우)에서도 사용자 데이터를 보호합니다. 이를 통해 사용자 및 IT 관리자 모두에게 개인과 기업의 정보를 항상 보호하고, 기기를 분실하거나 도난당한 경우에도 기기를 원격으로 지울 수 있는 방법 등 중요한 혜택을 제공합니다.

하드웨어 보안 기능

모바일 기기에서는 속도와 전력 효율이 가장 중요합니다. 암호화 작업은 복잡하여 속도와 전력 효율을 생각하지 않고 개발하면 성능 및 배터리 사용 시간이 문제가 될 수 있습니다.

모든 iOS 기기에는 전용 AES-256 암호화 엔진이 플래시 저장 장치와 메인 시스템 메모리 사이의 DMA 경로에 내장되어 있어 매우 효율적인 파일 암호화가 가능합니다. A9 또는 이후 버전의 A 시리즈 프로세서의 플래시 저장 장치 보조 시스템은 분리된 버스에 위치해 있으며 DMA 암호화 엔진을 통해서만 사용자 데이터가 포함되어 있는 메모리에 접근할 수 있습니다.

기기의 고유 ID(UID) 및 기기 그룹 ID(GID)는 AES 256비트 키가 제조 과정에서 응용 프로그램 프로세서 및 보안 엔클레이브에 융합(UID) 또는 컴파일(GID)된 것입니다. 소프트웨어나 펌웨어는 이를 직접적으로 읽을 수 없으며 실리콘 칩에 구현된 전용 AES 엔진이 UID 또는 GID를 키로 사용해 실행한 암호화 및 암호 해제 작업의 결과를 보기만 할 수 있습니다. 추가적으로 보안 엔클레이브의 UID 및 GID는 보안 엔클레이브 전용 AES 엔진에서만 사용할 수 있습니다. 또한 UID와 GID는 JTAG 또는 다른 디버그 인터페이스를 통해서는 사용이 불가능합니다.

T1, S2, S3 및 A9 또는 이후 버전의 A 시리즈 프로세서에서는 각 보안 엔클레이브가 UID(고유 ID)를 생성합니다. UID는 각 기기에 대해 고유하며 기기 밖 생산 시스템에서 생성되지 않고 완전히 보안 엔클레이브 내에서 생성되므로, UID는 Apple 또는 다른 공급업체에서 접근하거나 보관할 수 없습니다. 보안 엔클레이브에서 실행되는 소프트웨어는 UID를 사용하여 기기 전용 기밀을 보호합니다.

UID로 데이터를 특정 기기와 연결하여 암호화할 수 있습니다. 예를 들어 파일 시스템을 보호하는 키 계층에 UID가 포함되어 있어 메모리 칩을 물리적으로 다른 기기로 옮기는 경우 해당 파일에 접근할 수 없습니다. 하지만 UID는 기기의 다른 식별자와 관련되어 있지 않습니다.

GID는 같은 등급의 기기에 있는 모든 프로세서에 공통으로 적용됩니다(예: Apple A8 프로세서를 사용하는 모든 기기).

UID 및 GID를 제외한 다른 암호화 키는 CTR_DRBG 기반의 알고리즘을 사용한 시스템의 무작위 번호 생성기(RNG)를 통해 생성됩니다. 시스템 엔트로피는 부팅 중의 타이밍 변화가 있는 경우 생성되며 추가적으로 기기가 부팅된 이후에는 인터럽트 타이밍에도 생성됩니다. 보안 엔클레이브 내에서 생성된 키는 CTR_DRBG로 후처리된 다중 링 발진기(Multiple Ring Oscillator) 기반의 실제 하드웨어 무작위 번호 생성기를 사용합니다.

저장된 키를 안전하게 삭제하는 것은 키를 생성하는 것만큼이나 중요합니다. 특히 플래시 저장 장치에서 키를 삭제하는 것은 데이터의 다중 복사본을 지워야 할 수도 있는 웨어 레벨링 등으로 인해 더욱 복잡합니다. 이 문제를 해결하기 위해 iOS 기기는 삭제할 수 있는 저장 장치(Effaceable Storage)라고 불리는 안전한 데이터 삭제 전용 기능을 내장하고 있습니다. 이 기능은 기초 저장 장치 기술(예: NAND)에 접근하여 아주 낮은 레벨에 있는 작은 개수의 블록을 직접 찾아내고 삭제합니다.

파일 데이터 보호

Apple은 iOS 기기에 내장되어 있는 하드웨어 암호화 기능 이외에도 데이터 보호라는 기술을 사용하여 기기의 플래시 메모리에 저장된 데이터를 더욱 안전하게 보호합니다. 데이터 보호는 통화 수신과 같은 일반적인 이벤트에 기기가 응답하는 것을 허용하면서 사용자 데이터에 대해서는 높은 수준의 암호화를 제공합니다. 기본적으로 데이터 보호는 메시지, Mail, 캘린더, 연락처, 사진 및 건강 등의 중요 시스템 앱의 데이터 값에 사용됩니다. 또한 iOS 7 이상 버전에 설치된 타사 앱도 자동으로 이 데이터 보호를 사용합니다.

데이터 보호는 키 계층을 구성하고 관리하는 기능으로 각각의 iOS 기기에 내장된 하드웨어 암호화 기술을 기반으로 합니다. 데이터 보호는 각각의 파일을 클래스에 할당하는 파일별 기준으로 제어되며 접근성은 클래스 키가 암호 해제되었는지 여부에 따라 결정됩니다. APFS(Apple 파일 시스템)의 도입으로 파일 시스템은 키를 익스텐트 단위로 더욱 세분화할 수 있게 되었습니다(파일의 일부는 다른 키를 가질 수 있음).

아키텍처 개요

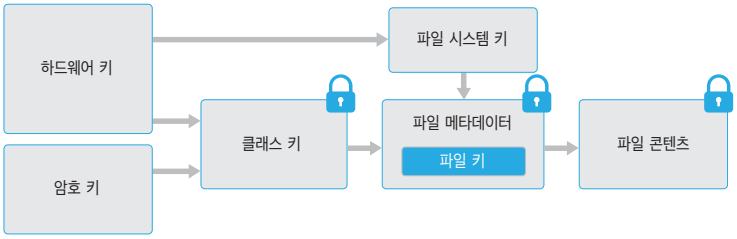
데이터 파티션에 파일이 생성될 때마다 데이터 보호는 256비트 키('파일별' 키)를 새로 생성하며 하드웨어 AES 엔진에 전달합니다. AES 엔진은 AES CBC 모드에서 플래시 메모리에 기록된 파일을 전달된 키로 암호화합니다(A8 이상의 프로세서를 사용하는 AES-XTS 모드를 사용). 초기화 벡터(IV)는 파일 내에서 블록 오프셋으로 계산되고 파일별 키의 SHA-1 해시로 암호화됩니다.

파일별(또는 익스텐트별) 키는 여러 클래스 키 중 하나로 래핑되며 클래스 키는 접근 가능한 파일에 따라 달라집니다. 일반적인 래핑과 마찬가지로 RFC 3394에 따라 NIST AES 키 래핑이 이루어집니다. 래핑된 파일별 키는 파일의 메타데이터에 저장됩니다.

Apple 파일 시스템 포맷으로 실행하는 기기는 파일 복제(Copy-On-Write 기술을 사용하는 제로 코스트 사본)를 지원할 수 있습니다. 파일이 복제되면 복제된 각 파일의 반은 수신되는 입력을 받아들이기 위해 새로운 키를 얻어서 새 키가 있는 미디어에 새 데이터를 씁니다. 시간이 지날수록 파일은 각기 다른 키에 매핑되는 다양한 익스텐트(또는 조각)로 구성됩니다. 하지만 파일을 구성하는 모든 익스텐트는 동일한 클래스 키에 의해 보호됩니다.

파일을 열면 파일 시스템 키로 메타데이터의 암호화가 해제되고 래핑된 파일별 키와 키를 보호하는 클래스 이름이 공개됩니다. 파일별(또는 익스텐트별) 키는 클래스 키를 통해 래핑이 해제되고 하드웨어 AES 엔진에 제공됩니다. AES 엔진은 파일을 플래시 메모리에서 읽은 상태 그대로 암호화를 해제합니다. 래핑된 파일 키에 대한 처리는 보안 엔클레이브에서 모두 이루어집니다. 또한 파일 키는 절대로 응용 프로그램 프로세서에 직접적으로 노출되지 않습니다. 부팅 시 보안 엔클레이브는 AES 엔진으로부터 임시 키를 양도받습니다. 보안 엔클레이브가 파일의 키를 래핑 해제하면 임시 키가 파일의 키를 다시 래핑하여 응용 프로그램 프로세서로 다시 보냅니다.

파일 시스템에 있는 모든 파일의 메타데이터는 무작위 키로 암호화되어 있습니다. 무작위 키는 iOS가 처음으로 설치되거나 사용자가 기기의 데이터를 지운 경우에 생성되는 키입니다. Apple 파일 시스템을 지원하는 기기에서 파일 시스템 메타데이터 키는 장기 보관용 보안 엔클레이브 UID에 의해 래핑됩니다. 파일별 또는 익스텐트별 키와 같이 메타데이터 키는 응용 프로그램 프로세서에 직접적으로 노출되지 않습니다. 보안 엔클레이브는 임시의 부트별 버전을 대신 제공합니다. 암호화된 파일 시스템 키가 보관되면 삭제할 수 있는 저장 장치(Effaceable Storage)에 보관된 '삭제할 수 있는 키'에 의해 추가적으로 래핑됩니다. 이 키는 데이터 기밀을 추가적으로 제공하지 않습니다. 대신에 키는 요청 시에 빠르게 삭제됩니다. 사용자가 '모든 콘텐츠 및 설정 지우기' 옵션을 사용하거나 사용자 또는 관리자가 MDM 솔루션, Exchange ActiveSync 또는 iCloud에서 원격 지우기 명령을 사용하여 삭제할 수 있습니다. 이러한 방식으로 키를 삭제하는 경우 모든 파일은 암호화되어 접근할 수 없습니다.



파일의 콘텐츠는 하나 이상의 파일별(또는 익스텐트별) 키로 암호화됩니다. 파일별 키는 클래스 키로 래핑되어 파일의 메타데이터에 저장됩니다. 또한 메타데이터는 파일 시스템 키로 암호화됩니다. 클래스 키는 주로 하드웨어 UID로 보호되지만 일부 클래스의 경우 사용자 암호로 보호됩니다. 이러한 보안 계층을 통해 유연성과 성능이 개선될 수 있습니다. 예를 들어 파일의 클래스를 변경하면 파일별 키만 다시 래핑되고 암호를 변경하면 클래스 키만 다시 래핑되기 때문입니다.

암호

암호 고려 사항

숫자만 포함하는 긴 암호를 입력하면 숫자 키패드가 전체 키보드 대신 잠금 화면에 표시됩니다. 비슷한 수준의 보안을 제공하는 짧은 알파벳 숫자 암호보다는 긴 숫자 암호가 더 입력하기 편할 수 있습니다.

기기 암호를 설정하면 사용자는 자동으로 데이터 보호 기능을 활성화하게 됩니다. iOS에서는 6자리 숫자, 4자리 숫자 및 사용자 지정 알파벳 숫자 암호를 지원합니다. 또한 암호는 기기를 잠금 해제하는 것 외에도 특정 암호화 키에 대한 엔트로피를 제공합니다. 이를 통해 해커가 기기의 소유권을 획득하여도 암호 없이는 특정 보호 클래스에 있는 데이터에 접근할 수가 없습니다.

암호는 기기의 UID와 연결되어 있어 무차별 대입 공격을 통해서만 해킹이 가능합니다. 하지만 반복 횟수가 늘어나면 암호 입력 속도가 점점 느려집니다. 반복 횟수는 시도 한 번에 약 80밀리초가 걸리도록 보정되었기 때문입니다. 결국에는 소문자와 숫자로 이루어진 6자리 알파벳 숫자 암호에 대한 모든 조합을 시도하는 데에만 5년 반 이상이 걸린다는 이야기입니다.

암호 입력 시도 지연 시간

시도	지연 시간
1-4	없음
5	1분
6	5분
7-8	15분
9	1시간

사용자 암호가 강력할수록 암호화 키도 더욱 강력해집니다. Touch ID 및 Face ID를 사용하면 일반적으로 사용되는 암호보다 더욱 강력한 암호를 설정할 수 있어 해킹에 더욱 잘 대처할 수 있습니다. 또한 iOS 기기를 손쉽게 잠금 해제하는 사용자 경험을 해치지 않으면서도 데이터 보호에 사용되는 암호화 키를 보호하는 엔트로피의 유효량을 증가시킬 수 있습니다.

무차별 암호 대입 공격의 추가 방지책으로는 잠금 화면에서 암호 입력을 실패한 경우 암호 입력을 지연시키는 시간 지연 기능이 있습니다. 설정 > Touch ID 및 암호 > 데이터 지우기가 켜져 있는 경우 10번 연속으로 암호 입력에 실패하면 기기가 자동으로 데이터를 지웁니다. 이 설정은 MDM 또는 Exchange ActiveSync의 관리 정책에서도 사용 가능하며 한계 시도 횟수를 낮추어 설정할 수 있습니다.

보안 엔클레이브를 사용하는 기기에서는 보안 엔클레이브 보조 프로세서가 시간 지연을 시행합니다. 덕분에 시간 지연 도중 기기가 재시동되어도 시간 지연은 계속되며 타이머가 현재 단계에 맞춰 다시 시작됩니다.

데이터 보호 클래스

iOS 기기에서 새로운 파일이 생성되면 그 파일을 생성한 앱이 파일을 클래스에 할당합니다. 각각의 클래스는 데이터 접근 시기를 결정하는 각기 다른 정책을 사용합니다. 기본 클래스와 정책은 다음 섹션에서 설명합니다.

Complete Protection

(`NSFileProtectionComplete`): 이 클래스 키는 사용자 암호 및 기기 UID에서 파생된 키로 보호됩니다. 사용자가 기기를 잠그고 잠시 후(암호 요구가 '즉시'로 설정되어 있는 경우 10초), 암호화가 해제된 클래스 키는 폐기되고, 사용자가 암호를 다시 입력하거나 Touch ID 또는 Face ID를 사용하여 기기를 잠금 해제하기 전까지 이 클래스에 있는 모든 데이터에 접근할 수 없도록 렌더링합니다.

Protected Unless Open

(`NSFileProtectionCompleteUnlessOpen`): 일부 파일은 기기가 잠금 상태일 때에도 쓸 수 있어야 합니다. 좋은 예로는 이메일 첨부 파일을 백그라운드에서 다운로드하는 상황입니다. 이 작업은 비대칭 타원곡선 암호화(Curve25519를 통한 ECDH)를 통해 실행됩니다. 일반적으로 파일별 키는 NIST SP 800-56A에 서술된 단일 패스 디피-헬만 키 합의(Diffie-Hellman Key Agreement)를 사용해 파생된 키로 보호됩니다.

단일 패스 디피-헬만 키 합의(Diffie-Hellman Key Agreement)의 임시 공개 키는 래핑된 파일별 키와 함께 저장됩니다. KDF는 NIST SP 800-56A의 5.8.1에 서술된 연속 키 유도 함수(Concatenation Key Derivation Function)(승인된 대안 1)입니다. 여기서 AlgorithmID는 생략됩니다. PartyUInfo는 임시 공개 키이며 PartyVInfo는 정적 공개 키입니다. SHA-256은 해시 함수로 사용됩니다. 파일을 닫으면 바로 파일별 키는 메모리에서 지워집니다. 파일을 다시 열려면 Protected Unless Open 클래스의 개인 키 및 파일의 임시 공개 키를 사용해 공유 비밀이 다시 생성됩니다. 이 개인 키와 임시 공개 키를 사용하여 파일별 키의 래핑을 해제한 다음 파일의 암호화도 해제합니다.

Protected Until First User Authentication

(`NSFileProtectionCompleteUntilFirstUserAuthentication`): 이 클래스는 기기가 잠기더라도 암호화가 해제된 클래스 키가 메모리에서 삭제되지 않는다는 점을 제외하면 Complete Protection과 같은 방식으로 행동합니다. 이 클래스의 보호 방식은 데스크탑 풀 볼륨 암호화(Full-Volume Encryption)와 비슷한 특징을 가지고 있으며 재시동과 관련된 공격으로부터 데이터를 보호합니다. 데이터 보호 클래스에 할당되지 않은 모든 타사 앱 데이터는 기본으로 이 클래스에 할당됩니다.

No Protection

(`NSFileProtectionNone`): 이 클래스 키는 UID로만 보호되며 삭제할 수 있는 저장 장치(Effaceable Storage)에 보관됩니다. 이 클래스에 있는 파일을 암호화 해제하는 데 필요한 모든 키는 기기에 저장되어 있기 때문에 암호화는 빠른 원격 지우기의 이점만 제공합니다. 파일이 데이터 보호 클래스에 할당되지 않더라도 iOS 기기의 다른 모든 데이터와 마찬가지로 암호화된 형태로 보관됩니다.

데이터 보호 클래스 키

클래스A	Complete Protection	(NSFileProtectionComplete)
클래스B	Protected Unless Open	(NSFileProtectionCompleteUnlessOpen)
클래스C	Protected Until First User Authentication	(NSFileProtectionCompleteUntilFirstUserAuthentication)
클래스D	No Protection	(NSFileProtectionNone)

키체인 데이터 보호

대부분의 앱은 키 및 로그인 토큰 같은 작지만 민감한 데이터와 암호를 안전하게 처리해야 합니다. iOS 키체인은 이러한 항목을 안전하게 저장하는 방법을 제공합니다.

키체인은 파일 시스템에 저장되어 있는 SQLite 데이터베이스로 구현되어 있습니다. 데이터베이스는 단 하나이며, **securityd** 데몬이 각 프로세스 또는 앱의 키체인 항목 접근 권한을 결정합니다. 결과적으로 키체인 접근 API가 데몬으로 요청되고 데몬은 앱의 'Keychain-access-groups', 'application-identifier', 'application-group' 권한을 묻는 방식입니다. 단일 프로세스의 접근을 제한하는 대신에 접근 그룹을 허용해 앱 간에 키체인 항목이 공유될 수 있습니다.

키체인 항목은 같은 개발자가 개발한 앱 사이에서만 공유될 수 있습니다. 응용 프로그램 그룹 설정을 위해 Apple Developer Program에서 할당받은 접두사로 타사 앱에 접근 그룹을 설정하도록 하여 이렇게 관리할 수 있습니다. 접두사 요구 사항 및 응용 프로그램 그룹 특정성은 코드 서명, 권한 설정 프로파일 및 Apple Developer Program을 통해 강제로 적용됩니다.

키체인 데이터는 파일 데이터 보호에 사용했던 것과 비슷한 클래스 구조를 사용해 보호됩니다. 이 클래스는 파일 데이터 보호 클래스와 비슷하지만 별개의 키를 사용하고 서로 다른 이름의 API에 속해 있는 점이 다릅니다.

사용 가능 시기	파일 데이터 보호	키체인 데이터 보호
잠금 해제한 경우	NSFileProtectionComplete	kSecAttrAccessibleWhenUnlocked
잠겨있는 경우	NSFileProtectionCompleteUnlessOpen	없음
처음 잠금 해제한 경우	NSFileProtectionCompleteUntilFirstUserAuthentication	kSecAttrAccessibleAfterFirstUnlock
항상	NSFileProtectionNone	kSecAttrAccessibleAlways
암호 활성화됨	없음	kSecAttrAccessible WhenPasscodeSetThisDeviceOnly

백그라운드 새로 고침 서비스를 활용하는 앱은 백그라운드 업데이트 중에 접근해야 하는 키체인 항목에 대해 `kSecAttrAccessibleAfterFirstUnlock` 클래스를 사용할 수 있습니다.

`kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly` 클래스는 `kSecAttrAccessibleWhenUnlocked` 클래스와 똑같이 동작하지만 기기의 암호가 구성된 경우에만 사용 가능합니다. 이 클래스는 System keybag에만 존재하며 iCloud 키체인에 동기화되지 않고 백업되지도 않으며 Escrow keybag에도 포함되지 않습니다. 암호가 제거되거나 재설정된 경우 이 클래스 키를 삭제하여 이 항목을 사용할 수 없는 것으로 간주합니다.

다른 키체인 클래스는 '이 기기에서만'의 대응 항목을 가지고 있습니다. 대응 항목은 백업 중에 기기에서 복사되는 경우 UID로 항상 보호되며 백업이 다른 기기에 복원되는 경우에는 사용할 수 없습니다.

Apple은 보호되는 정보의 종류와 iOS에서 필요로 하는 경우에 따라 키체인 클래스를 선택함으로써 보안성과 사용성의 균형을 맞추었습니다. 예를 들어, VPN 인증서는 기기가 연결을 계속 유지하도록 항상 사용 가능하여야 합니다. 하지만 '이동할 수 없음'으로 구분되어 있기 때문에 다른 기기로 이동은 할 수 없습니다.

키체인 항목의 구성요소

접근 그룹과 함께 각 키체인 항목은 '생성일' 및 '최근 업데이트 날짜' 타임스탬프 등의 관리 메타데이터를 포함합니다.

또한 각 항목은 SHA-1 해시를 포함합니다. 해시는 암호화를 해제하지 않고 계정 및 서버 이름 등의 항목 검색을 허용하기 위해 사용됩니다. 마지막으로 키체인 항목은 다음과 같은 암호화 데이터를 포함합니다.

- 버전 번호
- ACL(접근 제어 목록) 데이터
- 항목의 보호 클래스를 나타내는 값
- 보호 클래스 키로 래핑된 항목별 키
- SecItemAdd에 전달된 항목을 설명하는 속성 사전, 바이너리 plist로 인코딩되고 항목별 키로 암호화됨

암호화 방식은 GCM(Galois/Counter Mode)의 AES 128입니다. 접근 그룹은 속성에 포함되고 암호화 중에 계산된 GMAC 태그로 보호됩니다.

iOS에서 생성된 키체인 항목에 대해 다음과 같은 클래스 보호 방식이 강제로 적용됩니다.

항목	접근 가능 시기
Wi-Fi 암호	처음 잠금 해제한 경우
Mail 계정	처음 잠금 해제한 경우
Exchange 계정	처음 잠금 해제한 경우
VPN 암호	처음 잠금 해제한 경우
LDAP, CalDAV, CardDAV	처음 잠금 해제한 경우
소셜 네트워크 계정 토큰	처음 잠금 해제한 경우
Handoff 알림 암호화 키	처음 잠금 해제한 경우
iCloud 토큰	처음 잠금 해제한 경우
홈 공유 암호	잠금 해제한 경우
나의 iPhone 찾기 토큰	항상
음성 메시지	항상
iTunes 백업	잠금 해제한 경우, 이동할 수 없음
Safari 암호	잠금 해제한 경우
Safari 책갈피	잠금 해제한 경우
VPN 인증서	항상, 이동할 수 없음
Bluetooth® 키	항상, 이동할 수 없음
Apple 푸시 알림 서비스 토큰	항상, 이동할 수 없음
iCloud 인증서 및 개인 키	항상, 이동할 수 없음
iMessage 키	항상, 이동할 수 없음
구성 프로파일을 통해 설치된 인증서 및 개인 키	항상, 이동할 수 없음
SIM PIN	항상, 이동할 수 없음

키체인 접근 제어

키체인은 접근 제어 목록(ACL)을 통해 접근성과 인증 요구사항에 대한 정책을 설정할 수 있습니다. Touch ID, Face ID 또는 기기 암호 입력을 통해 인증하는 경우에만 키체인 항목에 접근할 수 있도록 명시하여 사용자의 직접 입력이 필요한 조건을 확립할 수 있습니다. 또한, 항목이 추가된 이후에 Touch ID 또는 Face ID 등록이 변경되지 않도록 명시하여 항목 접근을 제한할 수도 있습니다. 이러한 제한 사항 덕분에 해커가 지문을 추가해도 키체인 항목에 접근할 수 없습니다. ACL은 보안 엔클레이브 내에서 평가되어 명시된 제약 조건이 충족되는 경우에만 커널에 공개됩니다.

Safari에 저장된 암호 접근

iOS 앱은 암호 자동 완성 기능을 위해 Safari에 저장된 키체인 항목과 통신하는 데 다음 두 가지 API를 사용할 수 있습니다.

- `SecRequestSharedWebCredential`
- `SecAddSharedWebCredential`

앱 개발자 및 웹 사이트 관리자 모두가 허가를 하고 사용자가 동의를 하는 경우에만 접근이 허가됩니다. 앱 개발자는 앱 내에 권한을 포함시켜 Safari에 저장된 암호에 대한 접근을 요청합니다. 해당 권한은 관련 웹 사이트의 전체 주소 도메인 이름을 나열합니다. 이 웹 사이트는 승인한 앱의 고유 앱 식별자 목록을 파일로 만들어 서버에 저장해 두어야 합니다. com.apple.developer.associated-domains 권한을 가진 앱이 설치된 경우 iOS는 목록에 있는 웹 사이트에 TLS 요청을 보내 file/apple-app-site-association을 요청합니다. 설치되는 앱의 앱 식별자가 목록 파일에 있다면 iOS는 웹 사이트와 앱이 신뢰 관계를 가진 것으로 표시합니다. 신뢰 관계를 통해서만 위의 두 가지 API가 사용자에게 요청을 보내고 사용자는 반드시 동의를 선택해야 앱에 암호를 공개, 업데이트 또는 삭제할 수 있습니다.

iOS는 사용자가 iOS 키보드의 QuickType 막대에 있는 '키' 어포던스를 탭하여 저장된 사용자 이름 및 암호를 앱의 인증서 관련 필드에 입력하도록 허용합니다. 또한 동일한 apple-app-site-association 메커니즘을 활용하여 앱과 웹 사이트를 강력하게 연동합니다. 이 인터페이스는 사용자가 앱에 인증서를 공개하는 것에 동의하기 전까지는 어떠한 인증서 정보도 유출하지 않습니다. iOS에서 웹 사이트 및 앱을 신뢰할 수 있다고 표시한 경우 QuickType 막대도 해당 앱에 직접 인증 정보를 채울지 제안합니다. 이를 통해 사용자는 Safari에 저장된 인증서를 보안 과정이 동일하지만 API를 채택하지 않은 앱에 공개할지 선택할 수 있습니다.

Keybag

파일 및 키체인 데이터 보호 클래스에 대한 키는 Keybag에서 수집하고 관리합니다. iOS는 User, Device, Backup, Escrow 및 iCloud Backup의 Keybag을 사용합니다.

User keybag은 기기의 일반적인 동작에 사용되는 래핑된 클래스 키가 저장되는 공간입니다. 예를 들어 암호가 입력된 경우에는 `NSFileProtectionComplete` 키가 System keybag에서 로드되고 래핑이 해제됩니다. No Protection 클래스에 저장된 바이너리 plist이며 콘텐츠는 삭제할 수 있는 저장 장치(Effaceable Storage)에서 보관하는 키로 암호화되어 있습니다. Keybag에 전방향 안전성을 제공하기 위해 이 키는 사용자가 암호를 변경할 때마다 지워지고 다시 생성됩니다. `AppleKeyStore` 커널 확장 프로그램은 User keybag을 관리하며 기기의 잠금 상태에 대한 답변을 할 수 있습니다. AppleKeyStore는 User keybag의 모든 클래스 키가 접근 가능하고 성공적으로 래핑이 해제된 경우에만 기기가 잠금 해제된 것을 보고합니다.

Device keybag은 기기 특정 데이터와 관련된 동작을 위해 사용되는 래핑된 클래스 키를 저장하는 데 사용됩니다. 공용으로 구성된 iOS 기기의 경우 사용자가 로그인하기 전에 인증서에 대한 접근 권한이 필요하기도 합니다. 그러므로 사용자의 암호로 보호되지 않은 keybag이 필요합니다. iOS에서는 사용자별 파일 시스템 콘텐츠에 대한 개별 암호화를 지원하지 않습니다. 즉, 시스템에서는 파일별 키를 랩핑하는 데 Device keybag의 클래스 키를 사용합니다. 하지만 키체인의 경우 User keybag의 클래스 키를 사용하여 사용자 키체인에 있는 항목을 보호합니다. 단일 사용자용(기본 구성)으로 구성된 iOS 기기에서는 Device keybag과 User keybag이 통합되어 하나로 운영되고 사용자의 암호로 보호됩니다.

Backup keybag은 iTunes에서 암호화된 백업을 생성하고 백업을 기기가 백업된 컴퓨터에 저장한 경우에 생성됩니다. 새로운 keybag은 새로운 키 세트와 함께 생성되며 백업된 데이터는 이러한 새로운 키로 다시 암호화됩니다. 위에서 설명한 것처럼 이동할 수 없는 키체인 항목은 UID에서 파생된 키로 래핑된 상태라 원래 백업된 기기에는 복원할 수 있지만 다른 기기에서는 접근할 수 없습니다.

이 keybag은 iTunes에서 설정한 암호로 보호되며 PBKDF2(Password-Based Key Derivation Function 2)로 천만 번의 반복을 거칩니다. 반복 횟수가 크지만 특정 기기에 연관되지 않아 이론상으로는 동시에 다수의 컴퓨터에 가하는 무작위 대입 공격이 Backup keybag을 공격할 수 있습니다. 하지만 충분히 강력한 암호로 이러한 위협을 완화할 수 있습니다.

만약 사용자가 iTunes 백업을 암호화하지 않으면 백업 파일은 데이터 보호 클래스에 상관없이 암호화되지 않습니다. 하지만 키체인은 UID에서 파생된 키로 보호됩니다. 이 때문에 백업 암호가 설정된 경우에만 키체인 항목을 새로운 기기로 이동할 수 있습니다.

Escrow keybag은 iTunes 동기화와 MDM에 사용됩니다. 이 keybag은 사용자 암호 입력을 요청하지 않고 iTunes가 백업 및 동기화를 하도록 허용하고 MDM 솔루션에서 원격으로 사용자 암호를 지울 수 있도록 허용합니다. Escrow keybag은 iTunes와 동기화하는 데 사용한 컴퓨터에 저장되거나 기기를 관리하는 MDM 솔루션에 저장됩니다.

또한 Escrow keybag은 잠재적으로 데이터의 모든 클래스에 대한 접근이 요구되는 기기 동기화에서 개선된 사용자 환경을 제공합니다. 암호로 잠겨있는 기기가 처음으로 iTunes에 연결되면 사용자에게 암호 입력을 요청합니다. 그런 다음 기기는 기기에 사용된 것과 같은 클래스 키를 포함한 Escrow keybag을 생성하고 새롭게 생성된 키로 Escrow keybag을 보호합니다. Escrow keybag과 Escrow keybag을 보호하는 키는 기기에 저장되어 있던

Protected Until First User Authentication 클래스의 데이터와 함께 기기와 호스트(또는 기기와 서버)로 분리됩니다. 이런 이유로 재시동 후 처음으로 iTunes에 연결하고 백업을 할 때 암호를 입력해야 합니다.

OTA 소프트웨어 업데이트의 경우 업데이트를 시작할 때 사용자에게 암호 입력을 요청합니다. 암호가 입력되면 일회용 잠금 해제 토큰이 안전하게 생성됩니다. 이 토큰은 업데이트 후에 User keybag을 잠금 해제합니다. 사용자의 암호를 입력하지 않고 이 토큰을 생성할 수는 없으며 이전에 생성된 토큰은 사용자의 암호가 변경된 경우 무효화됩니다.

일회용 잠금 해제 토큰은 소프트웨어 업데이트의 자동 또는 수동 설치 모두에 필요합니다. 보안 엔클레이브, Keybag의 UUID 및 보안 클레이브의 UID에 있는 모노토닉 카운터의 현재 값에서 파생된 키를 사용해 이 토큰이 암호화됩니다.

보안 엔클레이브의 일회용 잠금 해제 토큰 카운터가 증가하면 모든 토큰이 무효화됩니다. 토큰 카운터는 토큰이 사용된 경우, 재시동된 기기를 처음으로 잠금 해제하는 경우, 소프트웨어 업데이트가 취소된 경우(사용자 또는 시스템에서 취소) 또는 토큰에 대한 정책 타이머가 만료된 경우에 증가됩니다.

수동 소프트웨어 업데이트에 대한 일회용 잠금 해제 토큰은 20분 후에 만료됩니다. 보안 엔클레이브에서 내보낸 이 토큰은 삭제할 수 있는 저장 장치(Effaceable Storage)에 기록됩니다. 기기가 20분 안에 재시동되지 않는 경우 정책 타이머가 카운터를 증가시킵니다.

자동 소프트웨어 업데이트는 사용자가 업데이트 알림을 받았을 때 '나중에 설치하기'를 선택하면 설정됩니다. 이 때에 응용 프로그램 프로세서는 일회용 잠금 해제 토큰을 보안 엔클레이브에 최대 8시간 동안 살려둘 수 있습니다. 그리고 정해진 시간이 지나면 정책 타이머가 카운터를 증가시킵니다.

iCloud Backup keybag은 Backup keybag과 비슷합니다. 이 keybag에 있는 모든 클래스 키는 비대칭(Protected Unless Open Data Protection 클래스와 같이 Curve25519를 사용함)이기 때문에 iCloud 백업이 백그라운드에서 실행될 수가 있습니다. No Protection을 제외한 모든 데이터 보호 클래스에서는 암호화된 데이터를 기기로부터 읽고 iCloud로 전송합니다. 해당되는 클래스 키는 iCloud 키가 보호합니다. 키체인 클래스 키는 UID에서 파생된 키로 래핑되며 iTunes 백업을 암호화 해제하는 것과 같은 방식입니다. 또한 비대칭 Keybag은 iCloud 키체인의 키체인 복원 작업 중 백업에도 사용됩니다.

보안 인증 및 프로그램

참고: iOS 보안 인증, 확인 및 지침에 대한 최신 정보를 보려면 아래 사이트로 이동하십시오. support.apple.com/ko-kr/HT202739

ISO 27001 및 27018 인증

Apple의 인프라, 개발 및 작업에 대한 정보 보안 관리 시스템은 ISO 27001 및 ISO 27018 인증을 받았으며 2017년 7월 11일 화요일에 작성된 적용성 보고서(Statement of Applicability) v2.1에 따라 Apple School Manager, iCloud, iMessage, FaceTime, 관리되는 Apple ID 및 iTunes U와 같은 제품과 서비스를 지원합니다. Apple은 영국 표준 규격 협회(British Standards Institution)에서 인증한 ISO 표준을 따릅니다. BSI 웹 사이트는 ISO 27001 및 ISO 27018 준수 인증을 받았습니다. 해당 인증서를 보려면 아래 사이트로 이동하십시오.

www.bsigroup.com/ko-KR/our-services/certification/certificate-and-client-directory/search-results/?searchkey=company=apple&licencenumber=IS+649475

www.bsigroup.com/ko-KR/our-services/certification/certificate-and-client-directory/search-results/?searchkey=company=Apple&licencenumber=PII%20673269

암호화 확인(FIPS 140-2)

iOS의 암호화 모듈은 iOS 6부터 각 릴리즈마다 미연방 정보 처리 표준(FIPS) 140-2 레벨 1을 준수하는 것으로 계속 검증 받아 왔습니다. iOS 운영 체제가 출시되면 Apple은 주요 릴리즈마다 CMVP에 다시 검증 받기 위해 모듈을 제출합니다. 이 프로그램은 iOS 암호화 서비스 및 승인된 알고리즘을 적절히 활용하는 Apple 앱과 타사 앱에 대한 암호화 작업의 무결성을 검증합니다.

CC 인증(ISO 15408)

Apple은 iOS 9이 출시된 이후로 CC 인증 프로그램 하에 아래 항목에 대해 주요 iOS 릴리즈마다 iOS 인증을 받았습니다.

- 모바일 기기 기본 보호 프로파일
- VPN IPSec 클라이언트 보호 프로파일
- MDM Agent용 확장 패키지
- Wireless LAN 클라이언트용 확장 패키지

iOS 11에는 다음 인증이 추가되었습니다.

- 응용 프로그램 소프트웨어 보호 프로파일
- 이메일 클라이언트용 확장 패키지
- 웹 브라우저용 확장 패키지

Apple은 앞으로 출시되는 주요 iOS 버전에도 동일한 인증을 받을 예정입니다. Apple은 국제 기술 커뮤니티(ITC) 내에서 중요 모바일 보안 기술 평가에 중점을 두고 현재 사용할 수 없는 공동 보호 프로파일(cPP) 개발에 적극적인 역할을 하고 있습니다. 또한 Apple은 현재 사용 가능한 새로운 버전과 업데이트된 버전의 보호 프로파일(cPP)에 대한 인증을 평가하고 받기 위해 계속 노력하고 있습니다.

CSfC(Commercial Solutions for Classified)

경우에 따라 Apple은 또한 CSfC(Commercial Solutions for Classified) 프로그램 컴포넌트 목록에 포함시키기 위해 iOS 플랫폼과 여러 가지 서비스를 제출했습니다. Apple 플랫폼과 서비스가 CC 인증을 통과하면 이 또한 CSfC 프로그램 구성요소 목록에 포함될 수 있도록 제출됩니다.

최근 등록된 구성요소를 보려면 아래 사이트로 이동하십시오.
www.nsa.gov/resources/everyone/csfc/components-list

보안 구성 설명서

Apple은 전 세계의 정부와 협력하여 고위험 환경에 있어서 더욱 안전한 환경을 유지하는 일명 '기기 강화 방법(device hardening)'을 위한 지침 및 권장 사항을 제시하는 설명서를 마련했습니다. 이 설명서는 보호를 강화하기 위해 iOS에 내장된 기능을 구성하고 활용하는 방법에 대한 정의되고 검토된 정보를 제공합니다.

앱 보안

앱은 현대 모바일 보안 아키텍처에 있어 가장 중요한 요소 중 하나입니다. 앱이 사용자에게 놀라운 생산성을 제공하지만 동시에 제대로 관리되지 않는다면 시스템 보안, 안정성 및 사용자 데이터에 부정적인 영향을 줄 가능성이 있습니다.

이런 이유로 iOS에서는 사용자 데이터를 보호하기 위해 앱을 서명 및 확인하고 샌드박스되었는지 확인하는 보호 계층을 제공합니다. 이러한 요소는 앱을 위한 안정적이고 안전한 플랫폼을 제공하여 수천 명의 개발자가 시스템 무결성을 해치지 않으면서 수 십만개의 앱을 iOS에 제공할 수 있도록 합니다. 그리고 사용자는 바이러스, 악성 코드 또는 인증되지 않은 공격에 대한 걱정 없이 이러한 앱을 iOS 기기에서 사용할 수 있습니다.

앱 코드 서명

iOS 커널이 시작되면 어떤 사용자 프로세스 및 앱을 실행할지를 제어합니다. iOS는 모든 앱이 알려지고 승인된 출처를 가지며 함부로 변경되지 않았음을 확인하기 위해 실행되는 모든 코드가 Apple이 발급한 인증서를 사용해 서명되도록 요구합니다. Mail 및 Safari와 같이 기기에 설치되어 제공되는 앱은 Apple이 서명했습니다. 타사 앱 또한 Apple이 발급한 인증서를 사용해 확인되고 서명되어야 합니다. 의무 코드 서명은 신뢰 체인의 개념을 OS에서부터 앱으로 확장하고 타사 앱이 서명되지 않은 코드 리소스를 로드하거나 자체 수정 코드를 사용하는 것을 방지합니다.

앱을 개발하거나 iOS 기기에 앱을 설치하기 위해서 개발자들은 Apple에 등록하고 Apple Developer Program에 가입해야 합니다. Apple은 각 개발자의 실제 신원이 개인인지 기업인지 확인한 후에 인증서를 발급합니다. 개발자는 App Store에 배포할 앱을 이 인증서를 사용하여 서명하고 배포할 수 있습니다. 결과적으로 App Store의 모든 앱은 신원 확인이 가능한 개인 또는 조직에서 제출하여 악성 앱 개발을 제지하는 데 기여합니다. 또한 Apple에서 앱을 검토하여 설명대로 작동하는지 명백한 오류 또는 다른 문제가 있는지 확인합니다. 이미 논의한 기술 이외에도 이러한 큐레이션 프로세스를 통해 고객은 구매하는 앱의 품질을 신뢰할 수 있습니다.

iOS는 개발자들이 앱 내에 프레임워크를 내장할 수 있도록 허용해 앱 자체에서 사용되거나 앱에 내장된 확장 프로그램에서 사용할 수 있도록 합니다. 시스템 및 다른 앱에서 타사 코드를 주소 공간 내에 로드하는 것을 방지하기 위해 시스템은 시작 시점에 프로세스가 링크하는 모든 동적 라이브러리의 코드 서명 확인을 수행합니다. 이 확인 절차는 Apple이 발급한 인증서에서 추출한 팀 식별자(Team ID)를 통해 수행됩니다. 팀 식별자는 예를 들어 1A2B3C4D5F와 같은 10자로된 알파벳 숫자 문자열입니다. 프로그램은 시스템과 함께 탑재된 플랫폼 라이브러리 또는 동일한 팀 식별자가 주 실행 파일로서 코드 서명에 있는 모든 라이브러리에 링크하게 됩니다. 시스템의 한 부분으로 탑재되는 실행 파일은 팀 식별자가 없기 때문에 시스템 자체와 함께 탑재되는 라이브러리에 링크하게 됩니다.

기업체는 또한 기업 내부용 앱을 작성하여 조직 내부에서 사용할 목적으로 직원에게 배포할 수 있습니다. 기업 및 조직은 DUNS 번호를 사용해 ADEP(Apple Developer Enterprise Program)에 신청할 수 있습니다. Apple이 신원 및 자격을 확인한 다음 신청을 승인합니다. ADEP의 구성원이 된 조직은 조직이 인증한 기기에서 기업 내부용 앱 사용을 허용하는 권한 설정 프로파일을 신청할 수 있습니다. 사용자가 기업 내부용 앱을 사용하려면 권한 설정 프로파일을 설치해야 합니다. 이를 통해 조직이 의도한 사용자만 iOS 기기에 해당 앱을 설치하도록 할 수 있습니다. MDM을 통해 설치된 앱은 조직 및 기기 간의 관계가 이미 구축되었기 때문에 절대적으로 신뢰할 수 있습니다. MDM을 통해 설치된 앱이 아닌 경우 사용자는 설정에서 앱의 권한 설정 프로파일을 승인해야 합니다. 조직에서는 또한 알 수 없는 개발자의 앱을 사용자가 승인할 수 없게 제한할 수 있습니다. 기업용 앱이 처음 실행되는 경우 기기는 Apple로부터 앱을 실행할 수 있다는 확인을 받아야 합니다.

다른 모바일 플랫폼과는 달리 iOS는 사용자가 서명되지 않은 잠재적인 악성 앱을 웹 사이트를 통해 설치하거나 신뢰할 수 없는 코드를 실행하도록 허용하지 않습니다. 런타임에서 코드 서명은 실행 가능한 모든 메모리 페이지가 로드된 상태 그대로인지 확인하여 앱이 설치되거나 마지막으로 업데이트되고 난 이후 변경되지 않았는지 확인합니다.

런타임 프로세스 보안

앱이 승인된 출처에서 전송된 것이 확인되면 iOS는 해당 앱이 다른 앱 또는 시스템 전체를 해치는 것을 방지하기 위해 고안된 보안책을 시행합니다.

모든 타사 앱은 '샌드박스'되어 있어 다른 앱이 저장한 파일에 접근하거나 기기에 변경 사항을 만들 수 없습니다. 이런 방식으로 앱이 다른 앱에서 저장한 정보를 수집하거나 변경하는 것을 방지합니다. 각 앱은 파일에 대한 고유 홈 디렉토리를 가지며 앱이 설치될 때에 임의로 할당됩니다. 타사 앱이 다른 디렉토리의 정보에 접근해야 하는 경우 명백하게 iOS에서 제공하는 서비스를 통해서만 접근할 수 있습니다.

또한 시스템 파일 및 리소스는 사용자가 설치한 앱으로부터 보호됩니다. 타사 앱과 마찬가지로 iOS의 대부분 기능은 권한이 없는 'mobile' 사용자로 실행됩니다. 전체 OS 파티션은 읽기 전용으로 마운트되어 있습니다. 원격 로그인 서비스와 같은 불필요한 툴은 시스템 소프트웨어에 없기 때문에 API에서는 앱이 다른 앱 또는 iOS 자체를 수정하기 위해 권한을 확대하는 것을 승인하지 않습니다.

타사 앱에서 사용자 정보와 iCloud 및 확장성과 같은 기능에 접근할 경우 선언된 권한을 사용해 제어됩니다. 권한은 쌍으로 된 키 값으로 앱에 로그인하여 UNIX 사용자 ID와 같은 런타임 요소를 능가하는 승인을 허용합니다. 권한은 디지털로 서명되어 있기 때문에 변경이 불가능합니다. 권한은 또한 시스템 앱과 데몬에서 광범위하게 사용되어 특정 권한이 필요한 작업을 수행합니다. 그렇지 않은 경우 root로 실행하기 위한 프로세스를 요청합니다. 이를 통해 손상된 시스템 앱 또는 데몬이 권한을 확대할 가능성을 크게 줄입니다.

추가적으로 앱은 시스템에서 제공한 API를 통해서만 백그라운드 프로세스를 실행할 수 있습니다. 이를 통해 앱 성능을 저하시키거나 배터리 사용 시간을 현저하게 줄이지 않으면서 계속 작동할 수 있게 됩니다.

주소 공간 배치 난수화(ASLR)는 메모리 변형 버그 공격을 막습니다. 내장된 앱은 ASLR을 사용하여 실행 시에 모든 메모리 영역을 무작위로 할당합니다. 실행 가능한 코드, 시스템 라이브러리 및 관련 프로그래밍 구성의 임의로 할당된 메모리 주소는 수많은 복잡한 공격의 가능성을 줄입니다. 예를 들어 RTL(Return-To-Libc) 공격은 스택 및 시스템 라이브러리의 메모리 주소를 조종해 기기를 속여 악성 코드를 실행하도록 할 수 있습니다. 이러한 것들을 임의로 배치하면 특히 다수의 기기에 대한 공격은 더욱 힘들어지게 됩니다. iOS 개발 환경인 Xcode는 자동으로 ASLR 지원을 켠 상태로 타사 프로그램을 컴파일합니다.

iOS에서는 메모리 페이지를 실행 불가능으로 표시하는 ARM의 실행 방지(XN) 기능을 사용해 더욱 강력한 보호 기능을 제공합니다. 실행 및 쓰기가 모두 가능하게 표시된 메모리 페이지는 완벽하게 제어된 조건에서만 앱에서 사용 가능합니다. 커널이 Apple 전용 동적 코드 서명 권한의 존재 여부를 확인합니다. 게다가 하나의 mmap 호출만이 실행 및 쓰기가 가능한 페이지를 요청할 수 있어, 이 경우에도 무작위의 주소가 주어집니다. Safari는 이 기능을 JavaScript JIT 컴파일러에 사용합니다.

확장 프로그램

iOS에서는 앱이 **확장 프로그램**을 제공하여 다른 앱으로 기능을 제공할 수 있습니다. 확장 프로그램은 특정 목적을 위해 서명된 실행 가능한 바이너리로서 앱 내에 패키지되어 있습니다. 시스템은 설치 시 자동으로 확장 프로그램을 인식하고 일치하는 시스템을 사용하는 다른 앱에서 확장 프로그램을 사용하도록 허용합니다.

확장 프로그램을 지원하는 시스템 영역을 **확장 포인트**라고 부릅니다. 각각의 확장 포인트는 API를 제공하며 해당 영역의 정책을 시행합니다. 시스템에서는 확장 포인트 지정 매칭규칙을 기반해 확장 프로그램의 사용 가능 여부를 판단합니다. 시스템은 필요한 경우 자동으로 확장 프로그램 프로세스를

실행하며 확장 프로그램의 수명을 관리합니다. 권한을 사용해 특정 시스템 앱이 확장 프로그램을 사용하지 못하게 제한할 수 있습니다. 예를 들어 오늘 보기 위젯이 알림 센터에만 나타나고 공유 확장 프로그램이 공유 패널에서만 사용 가능하도록 합니다. 확장 포인트에는 오늘 위젯, 공유, 사용자 설정 동작, 사진 편집, 도큐멘트 제공자 및 사용자 설정 키보드가 있습니다.

확장 프로그램은 자신의 주소 공간에서 실행됩니다. 확장 프로그램과 이를 활성화한 앱 간의 통신은 시스템 프레임워크를 통한 프로세스 간 통신을 사용합니다. 확장 프로그램과 앱은 서로 간의 파일 또는 메모리 공간에 대한 접근 권한이 없습니다. 확장 프로그램은 다른 확장 프로그램, 확장 프로그램을 포함하는 앱 및 확장 프로그램을 사용하는 앱으로부터 분리되도록 디자인되어 있습니다. 다른 타사 앱처럼 샌드박스되어 확장 프로그램을 포함하는 앱의 컨테이너와 분리된 컨테이너를 가지고 있습니다. 하지만 확장 프로그램을 포함하는 앱과 개인 정보 보호 제어에 대한 권한을 같이 공유합니다. 그렇기 때문에 사용자가 앱에서 연락처 접근을 허용하면 앱에 내장된 확장 프로그램에도 이 권한이 확장되지만 해당 앱이 활성화한 확장 프로그램에는 권한이 허용되지 않습니다.

사용자 설정 키보드는 사용자에 의해 전체 시스템에 활성화되는 특별한 유형의 확장 프로그램입니다. 활성화되면 키보드 확장 프로그램이 암호 입력 및 보안 텍스트 보기를 제외한 모든 텍스트 필드에 사용됩니다. 사용자 데이터 전송을 제한하기 위해 사용자 설정 키보드는 기본적으로 매우 제한된 샌드박스에서 실행되어 네트워크, 프로세스를 대신해 네트워크 작업을 실행하는 서비스, 입력한 데이터를 추출할 수 있는 확장 프로그램을 허용하는 API를 차단합니다. 사용자 설정 키보드 개발자는 확장 프로그램이 오픈 액세스 권한을 가지도록 요청할 수 있으며, 사용자가 동의하는 경우 시스템이 확장 프로그램을 기본 샌드박스에서 실행할 수 있습니다.

MDM 솔루션에 등록된 기기, 도큐멘트 및 키보드 확장 프로그램은 Managed Open In 규칙을 따릅니다. 예를 들어 MDM 솔루션은 사용자가 관리되는 앱에서 관리되지 않는 도큐멘트 제공자에게 도큐멘트를 보내는 것을 막을 수 있습니다. 또는 관리되지 않은 키보드를 관리되는 앱에서 사용하는 것을 막을 수 있습니다. 추가적으로 앱 개발자는 타사 키보드 확장 프로그램이 자신의 앱에서 사용되는 것을 막을 수 있습니다.

앱 그룹

앱 그룹의 일부로 구성된 경우 특정한 개발자 계정이 소유한 앱 및 확장 프로그램은 콘텐츠를 공유할 수 있습니다. 개발자가 Apple Developer 포털에 적절한 그룹을 생성하고 원하는 앱과 확장 프로그램을 포함할 수 있습니다. 앱 그룹으로 구성된 경우에 앱은 다음에 접근할 수 있습니다.

- 저장 공간을 위한 볼륨 상의 공유된 컨테이너(그룹에 속한 앱이 최소 한 개라도 설치되어 있는 경우 기기에 계속 남음)
- 공유된 환경설정
- 공유된 키체인 항목

Apple Developer 포털은 각각의 앱 그룹 ID가 전체 앱 생태계에서 고유함을 보장합니다.

앱의 데이터 보호

iOS 소프트웨어 개발 키트(SDK)는 API 전체 모음을 제공하므로 타사 및 내부 개발자가 데이터 보호를 손쉽게 적용하여 가장 높은 수준으로 앱을 보호할 수 있도록 합니다. 데이터 보호는 파일 및 데이터베이스 API(NSFileManager, CoreData, NSData 및 SQLite 등)에 사용할 수 있습니다.

Mail 앱 데이터베이스(첨부 파일 포함), 관리되는 책, Safari 책갈피, 앱 실행 이미지 및 위치 데이터 또한 키를 통해 암호화된 상태로 보관됩니다. 또한 이 키는 기기에서 사용자가 설정한 암호로 보호됩니다. 캘린더(첨부 파일 제외), 연락처, 미리 알림, 메모, 메시지 및 사진 앱은 Protected Until First User Authentication 클래스를 구현합니다.

특정 데이터 보호 클래스에 할당되지 않은 사용자가 설치한 앱은 기본으로 Protected Until First User Authentication 클래스에 할당됩니다.

액세서리

Made for iPhone, iPad, iPod touch(MFi) 라이선스 프로그램은 심사를 통과한 액세서리 생산 업체에게 iPod 액세서리 프로토콜(iAP) 및 필수 지원 하드웨어 구성요소에 대한 권한을 부여합니다.

MFi 액세서리가 Lightning 커넥터 또는 Bluetooth를 통해 iOS 기기와 통신을 하는 경우 기기는 액세서리에 Apple이 인증한 제품인지를 묻고 Apple이 제공한 인증서를 받으면 기기에서 확인합니다. 그리고 기기가 보내는 확인 요청에 액세서리는 서명된 응답을 보내야만 합니다. 이 프로세스는 전적으로 Apple이 액세서리 생산 업체에 제공하는 커스텀 IC(직접 회로)에 의해 처리되며 액세서리 자체에서는 이 과정을 인지하지 못합니다.

액세서리는 다른 전송 방식과 기능에 대한 접근을 요청할 수 있습니다. 예를 들어, Lightning 케이블을 통한 디지털 오디오 스트림 또는 Bluetooth를 통해 제공되는 위치 정보 등이 있습니다. 인증 IC는 승인된 액세서리만 기기에 대한 전체 접근 권한을 가질 수 있도록 합니다. 액세서리가 인증을 지원하지 않으면 액세서리는 시리얼(UART) 오디오 재생 제어의 일부 및 아날로그 오디오에 접근할 수 있는 권한만을 가지도록 제한됩니다.

AirPlay도 인증 IC를 활용해 수신 기기가 Apple에 승인을 받았는지 확인합니다. AirPlay 오디오 및 CarPlay 비디오 스트림은 CTR 모드의 AES-128을 사용해 액세서리와 기기간의 통신을 암호화하는 MFi-SAP(보안 연계 프로토콜)을 활용합니다. ECDH 키 교환(Curve25519)을 사용하여 임시 키가 교환되고 인증 IC의 1024비트 RSA 키를 STS(Station-to-Station) 프로토콜의 부분으로 사용해 서명됩니다.

HomeKit

HomeKit는 iCloud 및 iOS 보안을 활용해 Apple에 공개하지 않으면서도 개인 데이터를 보호하고 동기화할 수 있는 홈 자동화 인프라를 제공합니다.

HomeKit 신원

HomeKit 신원 및 보안은 Ed25519 공개-개인 키 쌍을 기반합니다. Ed25519 키 쌍은 각 HomeKit 사용자를 위해 iOS 기기에 생성되며 HomeKit 신원으로 사용됩니다. 이 키 쌍은 iOS 기기 간, iOS 기기 및 액세서리 간 통신을 인증하는 데 사용됩니다.

키는 키체인에 저장되며 암호화된 키체인 백업에만 포함됩니다. 또한 iCloud 키체인을 사용해 다른 기기에 이 키를 동기화할 수 있습니다.

HomeKit 액세서리의 통신

HomeKit 액세서리도 각각의 Ed25519 키 쌍을 생성해 iOS 기기와의 통신합니다. 액세서리가 초기화 설정으로 복원되는 경우 새로운 키 쌍이 생성됩니다.

iOS 기기와 HomeKit 액세서리 간의 관계는 SRP(Secure Remote Password) 프로토콜(3072비트)을 사용한 키 교환으로 구축됩니다. 키는 액세서리 제조업체에서 제공하는 8자리 코드를 활용하여 사용자가 iOS 기기에 입력하면 HKDF-SHA-512에서 파생된 키와 함께 ChaCha20-Poly1305 AEAD를 사용해 암호화됩니다. 액세서리의 MFi 인증은 설정 중에도 확인됩니다.

사용 중에 iOS 기기와 HomeKit 액세서리가 통신을 하면 위에서 설명한 프로세스를 통해 교환한 키를 활용하여 서로를 인증합니다. 각 세션은 STS(Station-to-Station) 프로토콜을 사용해 구축되며 세션별 Curve25519 키를 기반으로 HKDF-SHA-512에서 파생된 키를 사용해 암호화됩니다. 이는 IP기반 및 Bluetooth LE(저전력 Bluetooth) 액세서리에도 적용됩니다.

로컬 데이터 저장 공간

HomeKit는 홈, 액세서리, 장소 및 사용자에 대한 정보를 사용자의 iOS 기기에 저장합니다. 저장된 데이터는 사용자의 HomeKit 신원 키에서 파생된 키에 임의 nonce를 추가하여 암호화됩니다. 추가적으로 HomeKit 데이터는 데이터 보호 클래스 Protected Until First User Authentication를 사용해 저장됩니다. HomeKit 데이터는 암호화된 백업에만 백업됩니다. 예를 들어 암호화되지 않은 iTunes 백업에는 HomeKit 데이터가 포함되지 않습니다.

기기 및 사용자 간의 데이터 동기화

HomeKit 데이터는 iCloud와 iCloud 키체인을 사용해 사용자의 iOS 기기 간에 동기화될 수 있습니다. HomeKit 데이터는 사용자의 HomeKit 신원에서 파생된 키와 임의 nonce를 사용해 동기화 중에 암호화됩니다. 동기화 중에 데이터는 불투명한 Blob으로 처리됩니다. iCloud에 저장된 가장 최신의 Blob은 동기화를 활성화하지만 다른 용도에는 사용되지 않습니다. Blob은 사용자의 iOS 기기에서만 사용할 수 있는 키로 암호화되었기 때문에 Blob 콘텐츠는 전송 중에나 iCloud 저장 공간에서나 접근이 불가능합니다.

HomeKit 데이터는 또한 같은 홈에 있는 여러 사용자간에도 동기화됩니다. 이 프로세스는 iOS 기기 및 HomeKit 액세서리 간에 사용된 것과 같은 인증 및 암호화를 사용합니다. 인증은 사용자가 홈에 추가되었을 때 기기 간에 교환된 Ed25519 공개 키를 기반으로 합니다. 새로운 사용자가 홈에 추가되면 이후의 모든 통신이 STS(Station-to-Station) 프로토콜 및 세션별 키를 사용해 인증 및 암호화됩니다.

HomeKit에서 처음으로 홈을 생성한 사용자나 편집 권한을 가진 사용자만이 새로운 사용자를 추가할 수 있습니다. 새로운 사용자의 공개 키를 통해 소유자의 기기가 액세서리를 구성하므로 액세서리가 새로운 사용자로부터 인증을 받고 명령을 받을 수 있습니다. 편집 권한을 가진 사용자가 새로운 사용자를 추가하면 작업을 완료하기 위하여 해당 프로세스가 홈 허브로 위임됩니다.

사용자가 iCloud에 로그인하면 Apple TV에서 HomeKit를 사용하도록 권한을 설정하는 프로세스가 자동으로 수행됩니다. 해당 iCloud 계정은 이중 인증을 활성화해야 합니다. Apple TV와 소유자가 사용하는 기기는 iCloud를 통해 임시 Ed25519 공개 키를 교환합니다. 소유자가 사용하는 기기와 Apple TV가 동일한 로컬 네트워크 상에 있는 경우, 임시 키를 사용하여 STS(Station-to-Station) 프로토콜 및 세션별 키를 통한 안전한 로컬 네트워크 연결을 구축합니다. 이 프로세스는 iOS 기기 및 HomeKit 액세서리 간에 사용된 것과 같은 인증 및 암호화를 사용합니다. 이 안전한 로컬 연결을 통해 소유자가 사용하는 기기에서 사용자의 Ed25519 공개-개인 키 쌍을 Apple TV로 전송합니다. 그런 다음 해당 키는 Apple TV와 HomeKit 액세서리 간의 통신, Apple TV와 HomeKit 홈에 속한 다른 iOS 기기 간의 통신을 보호하는 데 사용됩니다.

사용자가 여러 기기를 가지고 있지 않고 홈에 새로운 사용자를 추가하지 않는다면 iCloud에 동기화되는 HomeKit 데이터는 없습니다.

홈 데이터 및 앱

앱의 홈 데이터 접근은 사용자의 개인 정보 보호 설정에서 제어할 수 있습니다. 연락처 앱, 사진 앱 및 다른 iOS 데이터 소스처럼 앱이 홈 데이터를 요청하는 경우 사용자에게 접근 승인을 요청합니다. 사용자가 승인하는 경우 앱은 방의 이름, 액세서리의 이름, 각 액세서리가 있는 방에 접근할 수 있습니다. 더 자세한 정보를 보려면 아래 사이트에서 HomeKit 개발자 설명서를 참조하십시오. developer.apple.com/homekit.

HomeKit 및 Siri

액세서리에 질의를 보내거나 액세서리를 제어하고 장소를 활성화시키기 위해 Siri를 사용할 수 있습니다. Siri는 홈 구성에 관해 최소한의 정보를 익명으로 제공받습니다. Siri가 명령을 인식하는데 필요한 방, 액세서리 및 장소의 이름이 제공됩니다. Siri에게 전송된 오디오는 특정 액세서리 또는 명령어를 나타낼 수 있지만 Siri 데이터는 HomeKit와 같은 Apple의 다른 기능과는 연결되지 않습니다. 자세한 정보를 보려면 이 도큐멘트의 인터넷 서비스 섹션에 있는 ‘Siri’를 참조하십시오.

HomeKit IP 카메라

HomeKit의 IP 카메라는 비디오 및 오디오 스트림에 접근하는 로컬 네트워크 상의 iOS 기기로 스트림을 직접 전송합니다. 스트림은 iOS 기기와 IP 카메라에서 무작위로 생성한 키를 사용하여 암호화됩니다. 이 키는 안전한 HomeKit 세션을 통해 카메라로 교환됩니다. iOS 기기가 로컬 네트워크상에 없는 경우 암호화된 스트림은 홈 허브를 통해 iOS 기기로 릴레이됩니다. 홈 허브는 iOS 기기와 IP 카메라 간에 릴레이하는 기능만 할 뿐 스트림의 암호화를 해제하지 않습니다. 앱이 사용자에게 HomeKit IP 카메라 비디오 보기를 표시하면 HomeKit는 독립된 시스템 프로세스에서 비디오 프레임을 안전하게 렌더링하기 때문에 앱에서는 해당 비디오 스트림에 접근하거나 스트림을 저장할 수 없습니다. 또한 앱은 이 스트림의 스크린샷을 캡처할 권한이 없습니다.

HomeKit 액세서리의 iCloud 원격 접근

Bluetooth 또는 Wi-Fi 통신을 사용할 수 없는 경우 HomeKit 액세서리는 iCloud에 직접 연결하여 액세서리를 제어하는 iOS 기기를 활성화합니다.

iCloud 원격 접근은 Apple에게 어떤 액세서리인지 또는 어떤 명령과 알림이 전송되는지에 대한 정보를 공개하지 않으면서 액세서리를 제어하고 알림을 보낼 수 있도록 세심하게 디자인되었습니다. HomeKit는 홈에 대한 어떤 정보도 iCloud 원격 접근을 통해 보내지 않습니다.

사용자가 iCloud 원격 접근을 사용해 명령을 보내면 액세서리 및 iOS 기기는 서로 인증하고 로컬 연결에서 설명한 것과 같은 절차를 사용해 데이터가 암호화됩니다. 통신 중에 콘텐츠는 암호화되며 Apple이 볼 수 없습니다. iCloud를 통한 주소 지정은 설정 단계에서 등록된 iCloud 식별자를 기반으로 이루어집니다.

iCloud 원격 접근을 지원하는 액세서리는 액세서리 설정 단계에서 권한이 설정됩니다. 권한 설정 절차는 사용자가 iCloud에 로그인하면서 시작됩니다. 다음으로 iOS 기기가 모든 Built for HomeKit 액세서리에 내장되어 있는 Apple 인증 보조 프로세서를 사용해 액세서리가 그 절차에 서명하도록 요청합니다. 또한 액세서리는 prime256v1 타원곡선 키를 생성하며 이 공개 키는 서명된 확인 요청 및 인증 보조 프로세서의 X.509 인증서와 함께 iOS 기기로 전송됩니다. 이는 iCloud 권한 설정 서버로부터 액세서리에 대한 인증서를 요구하는데 사용됩니다. 액세서리가 인증서를 저장하지만, 액세서리를 식별할 수 있는 정보는 포함하지 않고 액세서리가 HomeKit iCloud 원격 접근에 대한 접근 권한을 부여받은 것에 대한 정보만 포함하고 있습니다. 권한 설정을 시행하는 iOS 기기는 iCloud 원격 제어 서버에 연결하기 위해 필요한 URL 주소 및 다른 정보가 포함된 백을 액세서리에도 보냅니다. 이러한 정보는 특정 사용자 또는 액세서리에만 적용되지 않습니다.

각 액세서리는 iCloud 원격 접근 서버를 사용할 수 있는 사용자들의 목록을 등록합니다. 이 사용자들은 홈에 액세서리를 처음 추가한 사용자로부터 액세서리를 제어할 수 있는 권한을 부여받았습니다. 사용자는 iCloud 서버로부터 식별자를 부여받아 액세서리의 알림 메시지 및 응답을 전달하기 위한 목적으로 iCloud 계정에 매핑될 수 있습니다. 비슷하게 액세서리도 iCloud에서 발급한 식별자를 가지지만 이 식별자는 불투명한 상태로 액세서리 자체에 대한 어떤 정보도 공개하지 않습니다.

액세서리가 HomeKit iCloud 원격 접근 서버에 연결되면 액세서리는 인증서와 패스를 제시합니다. 패스는 다른 iCloud 서버에서 얻을 수 있으며 각 액세서리별로 고유하지 않습니다. 액세서리는 제조업체명, 모델 및 펌웨어 버전 정보를 포함시켜 패스를 요청합니다. 이 요청에는 사용자의 신원을 파악할 수 있거나 홈을 파악할 수 있는 정보는 포함되지 않습니다. 개인 정보 보호를 위해 패스 서버에 대한 연결은 인증받지 않습니다.

액세서리는 HTTP/2를 사용해 iCloud 원격 접근 서버에 연결하고 AES-128-GCM 및 SHA-256을 사용한 TLS v1.2를 통해 보호받습니다. 액세서리는 iCloud 원격 접근 서버에 대한 연결을 열어두어 수신 메시지를 받고 응답 및 발신 알림을 iOS 기기에 보낼 수 있습니다.

SiriKit

Siri는 iOS 확장 프로그램 메커니즘을 활용하여 타사 앱과 통신할 수 있습니다. Siri는 iOS 연락처와 기기의 현재 위치에 접근할 수 있지만 확장 프로그램을 포함하는 앱에서 가지고 있는 사용자 데이터(iOS에서 보호됨)에 대한 접근 권한을 먼저 확인한 다음 해당 정보에 대한 접근 권한 여부가 확인되면 해당 정보를 앱에 제공합니다. Siri는 사용자 질문 텍스트 중에서 관련된 부분만 확장 프로그램에 전달합니다. 예를 들어, iOS 연락처에 대한 접근 권한이 없는 앱이라면 Siri는 “결제 앱으로 엄마한테 10000원 지불해”와 같은 사용자 요청이 있는 경우에도 관계 정보를 해당 앱에 제공하지 않습니다. 이 경우에 확장 프로그램의 앱에서는 전달받은 부분적인 문장에서 ‘엄마’라는 텍스트만 볼 수 있습니다. 만약 iOS 연락처에 대한 접근 권한이 있는 앱이라면 사용자의 어머니에 대한 iOS 연락처 정보가 전달됩니다. “오빠가 훌륭해라고 엄마한테 메시지 앱으로 문자 보내”와 같이 연락처가 메시지 내용에 언급되는 경우에 Siri는 앱의 TCC에 상관없이 ‘오빠’를 관계 정보로 처리하지 않습니다. 해당 앱에서 사용자가 사용할 수 있는 단어를 Siri가 이해할 수 있도록 앱에서 보여주는 콘텐츠가 서버로 전송될 수도 있습니다.

"<앱 이름>을 사용해서 엄마 집까지 안내해 줘"와 같이 사용자의 요청이 연락처에 있는 상대의 위치 정보를 가져와야 하는 경우, Siri는 앱의 위치 접근 권한 또는 연락처 접근 권한에 관계없이 해당 요청에 한해서 위치 정보를 앱 확장 프로그램에 제공합니다.

런타임에서 Siri는 SiriKit를 사용하는 앱이 응용 프로그램 동작 사례에 맞는 특정 단어 세트를 제공하도록 허용합니다. 이러한 특정 단어 세트는 무작위 ID(이 도큐멘트의 Siri 섹션에서 설명됨)에 연결되어 있으며 동일한 수명을 가집니다.

HealthKit

HealthKit는 사용자의 허가 하에 건강 앱과 운동 앱의 데이터를 저장합니다. HealthKit는 또한 Bluetooth LE 심박수 측정기 및 동작 인식 보조 프로세서와 같은 건강 및 운동 기기와 직접 동작합니다.

건강 데이터

HealthKit는 키, 몸무게, 걸은 거리, 혈압 등의 사용자 건강 데이터를 저장하고 종합합니다. 이 데이터는 데이터 보호 클래스인 Complete Protection에 저장되어 사용자가 암호를 입력하거나 Touch ID 또는 Face ID를 사용해 기기를 잠금 해제할 경우에만 접근이 가능합니다.

또한 HealthKit는 앱의 접근 권한, HealthKit에 연결된 기기의 이름, 새로운 데이터가 사용 가능할 경우 앱 실행에 사용되는 스케줄 정보 등의 관리 데이터를 종합합니다. 이 데이터는 데이터 보호 클래스인 Protected Until First User Authentication에 저장됩니다.

임시 저널 파일은 기기가 잠겨지는 경우에 생성되는 건강 기록(예를 들어 사용자가 운동 중일 때)을 저장합니다. 이 파일은 데이터 보호 클래스인 Protected Unless Open에 저장됩니다. 기기가 잠금 해제되면 기본 건강 데이터베이스에 임시 저널 파일을 가져오고 병합이 완료되면 파일은 삭제됩니다.

건강 데이터는 iCloud에 저장할 수 있습니다. iCloud 저장 공간을 구성한 경우 건강 데이터는 기기 간에 동기화되며 저장된 데이터 또는 전송 중인 데이터를 보호하는 암호화를 통해 안전하게 보호됩니다. 건강 데이터는 암호된 iTunes 백업에만 포함됩니다. 암호화되지 않은 iTunes 백업 또는 iCloud 백업에는 포함되지 않습니다.

데이터 무결성

데이터베이스에 저장된 데이터는 메타데이터를 가지고 있어 각 데이터 기록의 출처를 추적할 수 있습니다. 이 메타데이터에는 앱 식별자가 포함되어 있어 기록을 저장한 앱을 식별할 수 있습니다. 추가적으로 선택적인 메타데이터 항목은 디지털 서명된 기록의 사본을 포함할 수 있습니다. 이로 인해 신뢰하는 기기에서 생성된 기록은 데이터 무결성을 유지할 수 있습니다. 디지털 서명에 사용된 포맷은 IETF RFC 5652에 명시된 암호 메시지 구문(CMS)입니다.

타사 앱에서의 접근

HealthKit API에 대한 접근은 권한을 통해 제어되며 앱은 데이터의 사용 방식에 대한 제한 사항을 따라야 합니다. 예를 들어 앱은 건강 데이터를 광고에 활용할 수 없습니다. 앱은 또한 건강 데이터 사용에 대해 상세히 설명한 개인정보 취급방침을 사용자에게 필수적으로 제공해야 합니다.

사용자는 개인 정보 보호 설정에서 건강 데이터에 접근하는 앱을 제어할 수 있습니다. 연락처 앱, 사진 앱 및 다른 iOS 데이터 소스처럼 앱이 건강 데이터에 대한 접근을 요청하는 경우 사용자에게 접근 승인을 요청합니다. 하지만 건강 데이터의 경우 앱은 건강 데이터의 각 유형에 대해서와 마찬가지로 읽기 및 쓰기 데이터에 대한 접근을 따로 부여받습니다. 사용자는 건강 앱의 소스 탭에서 접근을 허용한 건강 데이터를 보거나 접근 취소 및 권한 확인을 할 수 있습니다.

데이터 쓰기 권한이 부여된 경우 앱은 앱이 쓴 데이터를 읽을 수도 있습니다. 데이터 읽기 권한이 부여된 경우 앱은 모든 소스가 쓴 데이터를 읽을 수 있습니다. 하지만 앱은 다른 앱에 부여된 권한을 확인할 수 없습니다. 게다가 앱은 건강 데이터에 읽기 접근 권한이 있는지를 확인할 수 없습니다. 앱이 읽기 접근 권한을 가지지 않은 경우 모든 질의에 데이터 없음을 반환합니다. 데이터 없음은 데이터베이스가 비어있을 때 반환하는 결과와 같습니다. 이렇게 함으로써 앱이 사용자가 추적하는 데이터 유형을 학습하여 사용자의 건강 상태를 추측하는 것을 방지합니다.

의료 정보

건강 앱은 사용자에게 의료 정보 양식을 제공해 긴급 상황에서 중요하게 사용될 수 있는 정보를 작성하는 옵션을 제공합니다. 이 정보는 수동으로 입력되고 업데이트되며 건강 데이터베이스의 정보와는 동기화되지 않습니다.

의료 정보는 잠금 화면에서 긴급상황 버튼을 눌러 볼 수 있습니다. 기기에 저장된 이 정보는 데이터 보호 클래스인 No Protection을 사용하므로 기기 암호를 입력하지 않고도 접근이 가능합니다. 의료 정보는 선택적 기능으로 사용자가 안전과 개인 정보 보호 중 선택하여 활성화할 수 있습니다.

ReplayKit

ReplayKit는 개발자가 앱에 녹화 및 라이브 방송 기능을 추가하도록 하는 프레임워크입니다. 또한, 사용자가 기기의 전면 카메라와 마이크를 사용하여 녹화 영상과 라이브 방송에 주석을 달 수 있게 합니다.

동영상 녹화

동영상 녹화에는 다음과 같이 여러 가지 보안 계층이 구축되어 있습니다.

- **권한 요청 알림:** 녹화를 시작하기 전에 ReplayKit가 사용자 동의 요청 알림을 나타내 사용자가 화면을 녹화하고 마이크와 전면 카메라를 사용하는 목적을 사용자가 확인하도록 합니다. 이 알림은 앱 프로세스마다 한 번씩 나타나며 앱이 8분 이상 백그라운드에서 실행되는 경우 알림이 다시 나타납니다.
- **화면 및 오디오 캡처:** 화면 및 오디오 캡처는 앱 프로세스가 아닌 ReplayKit의 데몬 **replayd**에서 이루어집니다. 이로 인해 녹화된 콘텐츠는 앱 프로세스에서 절대 접근할 수 없습니다.
- **동영상 생성 및 저장:** 동영상 파일은 ReplayKit의 보조 시스템에서만 접근할 수 있는 디렉토리에 작성되며 다른 앱에서는 절대 접근할 수 없습니다. 이 때문에 타사 개발자는 사용자의 동의 없이 녹화 영상을 사용할 수 없습니다.
- **최종 사용자 미리보기 및 공유:** 사용자는 ReplayKit에서 제공된 UI를 통해 동영상을 미리 보거나 공유할 수 있습니다. 해당 UI는 iOS 확장 프로그램 인프라를 통해 프로세스 밖에서 제공되며 생성된 동영상 파일에 대한 접근 권한을 가집니다.

방송

- **화면 및 오디오 캡처:** 방송 중의 화면 및 오디오 캡처 메커니즘은 **replayd**에서 사용하는 동영상 녹화 메커니즘과 동일합니다.
- **방송 확장 프로그램:** 타사 서비스에서 ReplayKit 방송에 참여하려고 하는 경우 com.apple.broadcast-services 엔드포인트로 구성된 아래와 같은 두 개의 새로운 확장 프로그램을 생성해야 합니다.
  - 사용자가 방송을 설정하도록 하는 UI 확장 프로그램
  - 비디오 및 오디오 데이터를 서비스의 백엔드 서버로 업로드하는 업로드 확장 프로그램

아키텍처는 방송된 비디오 및 오디오 콘텐츠 전용 ReplayKit에 호스팅 앱이 접근할 수 없도록 하고 타사 방송 확장 프로그램은 접근할 수 있도록 합니다.

- **방송 선택기:** 사용하려는 방송 서비스를 선택할 수 있도록 ReplayKit에서는 개발자가 앱에 추가 가능한 뷰 컨트롤러(UIActivityViewController와 유사함)를 제공합니다. 뷰 컨트롤러는 UIRemoteViewController SPI를 통해 구현되는 ReplayKit 프레임워크 내장 확장 프로그램입니다. 또한, 호스팅 앱 외부의 프로세스입니다.
- **업로드 확장 프로그램:** 방송 중에 비디오와 오디오를 처리하기 위해 타사 방송 서비스에서 구현하는 업로드 확장 프로그램은 아래와 같이 두 가지 방법으로 콘텐츠를 받을 수 있습니다.
  - 저용량으로 인코딩된 MP4 클립
  - Raw 포맷으로 인코딩되지 않은 샘플 버퍼

- **MP4 클립 처리:** 처리 모드에서 저용량으로 인코딩된 MP4 클립은 **replayd**에서 생성되며 ReplayKit의 보조 시스템에서만 접근할 수 있는 비공개 위치에 저장됩니다. 동영상 클립이 생성되면 **replayd**에서 XPC 기반의 NSExtension 요청 SPI를 통해 동영상 클립의 위치를 타사 업로드 확장 프로그램에 전달합니다. 또한 **replayd**는 확장 프로그램 요청이 있는 동안 업로드 확장 프로그램이 특정 동영상 클립에 접근할 수 있도록 하는 일회용 샌드박스 토큰을 생성하여 업로드 확장 프로그램에 전달합니다.
- **샘플 버퍼 처리:** 처리 모드에서 비디오 및 오디오 데이터는 직렬화되어 직접 XPC 연결을 통해 타사 업로드 확장 프로그램에 실시간으로 전달됩니다. 비디오 데이터는 비디오 샘플 버퍼에서 IOSurface 대상체를 추출하여 인코딩된 다음, XPC 대상체로 안전하게 인코딩되어 XPC를 통해 타사 확장 프로그램으로 전송되어 다시 IOSurface 대상체로 안전하게 디코딩됩니다.

보안 메모

메모 앱에는 특정 메모의 콘텐츠를 보호하는 보안 메모 기능이 있습니다. 보안 메모는 사용자가 입력한 암호(암호문구)로 암호화되며 iOS, macOS 및 iCloud 웹 사이트에서 보안 메모를 보려면 이 암호가 필요합니다.

사용자가 메모를 보호하면 PBKDF2 및 SHA256을 통하여 사용자의 암호에서 16바이트 키가 파생됩니다. 메모의 콘텐츠는 AES-GCM을 사용하여 암호화됩니다. Core Data 및 CloudKit에 새로운 기록이 생성되어 암호화된 메모, 태그 및 초기화 벡터를 저장합니다. 기존의 메모 기록은 삭제되고 암호화된 데이터는 기록되지 않습니다. 첨부 파일도 같은 방식으로 암호화됩니다. 지원되는 첨부 파일은 이미지, 스케치, 표, 지도 및 웹 사이트입니다. 다른 유형의 첨부 파일을 포함하는 메모는 암호화할 수 없으며 지원되지 않는 첨부 파일은 보안 메모에 추가될 수 없습니다.

사용자가 보안 메모를 보거나 생성하기 위하여 암호를 입력하는 데 성공하면, 메모 앱에서 보안 세션을 엽니다. 세션이 열려 있는 동안 사용자는 암호를 입력하거나 Touch ID 또는 Face ID를 사용하지 않고도 다른 메모를 보거나 보호할 수 있습니다. 하지만 일부 메모에서 다른 암호를 사용하는 경우, 해당 보안 세션은 현재 입력된 암호로 보호된 메모에만 적용됩니다. 다음과 같은 경우 보안 세션이 닫힙니다.

- 사용자가 메모 앱에서 지금 잠금 버튼을 탭한 경우.
- 메모 앱이 백그라운드로 전환된지 3분 이상 지난 경우.
- 기기가 잠긴 경우.

사용자가 암호를 잊어버려도 기기에서 Touch ID 또는 Face ID가 활성화된 상태라면 보안 메모를 보거나 추가로 메모를 보호할 수도 있습니다. 또한, 암호를 세 번 이상 틀리게 입력하면 메모 앱에서 사용자가 입력한 힌트를 보여줍니다. 암호를 변경하려면 사용자는 현재 설정된 암호를 알아야 합니다.

현재 설정된 암호를 잊어버렸다면 사용자는 암호를 재설정할 수 있습니다. 암호 재설정 기능으로 사용자는 새로운 암호로 보안 메모를 새로 생성할 수는 있지만 이전 암호로 보호한 메모는 볼 수 없습니다. 이전 암호로 보호한 메모는 사용자가 이전 암호를 기억하는 경우에만 볼 수 있습니다. 암호를 재설정하려면 사용자의 iCloud 계정 암호가 필요합니다.

공유 메모

메모는 다른 사람과 공유할 수 있습니다. 공유 메모는 엔드 투 엔드 암호화를 사용하지 않습니다. Apple은 사용자가 메모에 넣은 모든 텍스트 또는 첨부 파일에 대해 CloudKit로 암호화한 데이터 유형을 사용합니다. 자료는 CKRecord에서 암호화된 키로 항상 암호화됩니다. 생성일 및 수정일 등의 메타데이터는 암호화되지 않습니다. 공유 참여자가 서로의 데이터를 암호화하거나 암호화를 해제하는 프로세스는 CloudKit에서 관리합니다.

Apple Watch

Apple Watch는 iOS를 위해 개발된 보안 기능과 기술을 사용하여 기기의 데이터, 쌍으로 연결된 iPhone과의 통신 및 인터넷 통신을 보호합니다. 해당 기술에는 데이터 보호 및 키체인 접근 제어 등의 기술이 포함됩니다. 또한 사용자 암호는 기기 UID와 연결되어 암호화 키를 생성합니다.

Apple Watch와 iPhone 간의 페어링은 BTLE 링크 공유 비밀을 따르는 공개 키 교환을 대역 외(OOB) 프로세스를 사용해 보호합니다. Apple Watch는 iPhone에서 카메라를 사용해 캡처할 수 있는 움직이는 패턴을 표시합니다. 이 패턴은 BTLE 4.1 대역 외(OOB) 페어링에 사용되는 암호화된 비밀을 포함하고 있습니다. 필요한 경우 표준 BTLE 패스키 엔트리가 폴백 페어링 방식으로 사용됩니다.

BTLE 세션이 구축되면 Apple Watch와 iPhone은 IDS에서 조정된 프로세스를 사용해 키를 교환합니다(IDS는 이 도큐멘트의 iMessage 섹션에서 설명합니다). 키가 교환되면 Bluetooth 세션 키는 삭제되고 Apple Watch와 iPhone 간의 모든 통신은 IDS를 사용해 암호화되며 암호화된 Bluetooth, Wi-Fi 및 셀룰러 링크는 2차 암호화 계층을 제공합니다. 트래픽이 손상된 경우 키 롤링을 15분 간격으로 활용해 노출을 제한합니다.

스트리밍 데이터가 필요한 앱을 지원하기 위해 이 도큐멘트의 인터넷 서비스 섹션에 있는 'FaceTime'에서 설명하는 방식을 사용해 암호화가 제공됩니다. 암호화는 인터넷에 직접 연결하거나 쌍으로 연결된 iPhone에서 제공하는 IDS 서비스를 이용합니다.

Apple Watch는 이 도큐멘트의 암호화 및 데이터 보호 섹션에서 설명한 것과 같이 하드웨어 기반 암호화된 저장 장치와 파일 및 키체인 항목에 대한 클래스 기반의 보호를 구현합니다. 또한 키체인 항목에 대해서는 접근이 제어되는 Keybag이 사용됩니다. 시계와 iPhone 간의 통신에 사용되는 키 또한 클래스 기반의 보호를 사용합니다.

Apple Watch가 Bluetooth 범위 안에 있지 않다면 Wi-Fi 또는 셀룰러를 대신 사용할 수 있습니다. Wi-Fi 네트워크에 연결하는 데 사용하는 자격 증명(이전에 Apple Watch에 동기화되어 있어야 함)이 쌍으로 연결된 iPhone에 없는 경우 Apple Watch는 Wi-Fi 네트워크에 연결되지 않습니다. Apple Watch가 iPhone의 범위에서 벗어나면 iPhone에 새로 저장되는 네트워크 자격 증명은 Apple Watch로 전송되지 않습니다.

Apple Watch는 측면 버튼을 길게 눌러 수동으로 잠글 수 있습니다. 추가적으로 기기가 손목에서 벗겨진 경우 동작 휴리스틱(Heuristics) 기능을 통해 자동으로 기기를 잠급니다. Apple Watch가 잠기면 Apple Watch 암호를 입력해야만 Apple Pay를 사용할 수 있습니다. 손목 인식은 iPhone의 Apple Watch 앱을 사용해 끌 수 있습니다. 이 설정은 또한 MDM 솔루션을 사용해 강제로 적용할 수 있습니다.

시계를 착용한 경우에 쌍으로 연결된 iPhone에서도 시계를 잠금 해제할 수 있습니다. 이는 페어링 중에 구축된 키가 연결 구축을 인증하여 가능한 기능입니다. iPhone에서는 시계가 데이터 보호 키를 잠금 해제하는데 사용할 수 있는 키를 보냅니다. 시계 암호는 iPhone으로 전송되지 않으며 iPhone에서 알 수 없습니다. 이 기능은 iPhone에서 Apple Watch 앱을 사용하여 끌 수 있습니다.

Apple Watch는 한 번에 한 개의 iPhone에만 연결될 수 있습니다. 연결을 해제하면 iPhone에서 Apple Watch에 저장된 모든 콘텐츠와 데이터를 지우도록 명령을 보냅니다.

쌍으로 연결된 iPhone에서 나의 iPhone 찾기를 활성화하면 Apple Watch에서도 활성화 잠금을 사용할 수 있습니다. 활성화 잠금은 Apple Watch가 분실 또는 도난되었을 때 다른 사람이 사용하거나 판매하기 어렵게 만듭니다. 활성화 잠금은 쌍으로 연결된 Apple Watch를 연결 해제, 삭제, 재활성화할 경우 사용자의 Apple ID 및 암호를 요구합니다.

네트워크 보안

iOS 기기에 저장된 데이터를 보호하기 위해 Apple이 사용하는 기본 제공 안전 장치 이외에도, 많은 네트워크 보안책이 마련되어 있어 조직에서 iOS 기기에서 또는 기기로부터 전송되는 정보를 안전하게 보호할 수 있습니다.

모바일 사용자는 세계 어느 곳에서든 기업 네트워크에 접근할 수 있어야 하므로 모바일 사용자가 인증을 받고 사용자의 데이터가 전송 중에 보호되도록 하는 것이 중요합니다. iOS는 권한을 부여받고 인증받은 암호화된 통신에 대해 표준 네트워크 프로토콜을 사용하며 개발자에게 접근 권한을 제공합니다. 이러한 보안 목적을 달성하기 위해 iOS는 Wi-Fi 및 셀룰러 데이터 네트워크 연결의 최신 표준과 입증된 기술을 통합하였습니다.

다른 플랫폼에서는 공개 통신 포트를 공격으로부터 보호하기 위해서는 방화벽 소프트웨어가 필요합니다. iOS는 수신 포트를 제한하고 telnet, 쉘, 웹 서버와 같은 필요없는 네트워크 유틸리티를 제거하여 공격 노출을 줄였습니다. 따라서 iOS에서는 방화벽 소프트웨어가 추가로 필요하지 않습니다.

TLS

iOS는 전송 계층 보안(TLS v1.0, TLS v1.1, TLS v1.2)과 DTLS를 사용합니다. AES-128과 AES-256을 모두 지원하며 완전 순방향 비밀성을 갖춘 암호 모음을 선호합니다. Safari, 캘린더, Mail 및 기타 인터넷을 사용하는 앱은 자동으로 이러한 프로토콜을 사용해 기기와 네트워크 서비스 간에 암호화된 통신 채널을 활성화합니다. CFNetwork와 같은 상위 수준 API는 개발자가 TLS를 앱에 쉽게 적용할 수 있으며 SecureTransport와 같은 하위 수준 API는 세부 제어 권한을 제공합니다. CFNetwork는 SSLv3를 허가하지 않으며 Safari와 같은 WebKit를 사용하는 앱은 SSLv3 연결이 금지되어 있습니다.

iOS 11 및 macOS High Sierra부터 사용자가 신뢰하지 않은 경우 SHA-1 인증서를 더 이상 TLS 연결에 대해 허용하지 않습니다. 2048비트 이하의 RSA 키를 지닌 인증서도 허용하지 않습니다. RC4 대칭 암호 모음은 iOS 10과 macOS Sierra에서 제거되었습니다. 기본적으로 SecureTransport API로 구현된 TLS 클라이언트 또는 서버는 RC4 암호 모음이 활성화되어 있지 않고 RC4 암호 모음만 사용할 수 있는 경우에는 연결할 수 없습니다. 보안 강화를 위해 RC4를 요구하는 서버 또는 앱은 최신의 안전한 암호 모음을 사용하도록 업그레이드해야 합니다.

앱 전송 보안

앱 전송 보안은 기본적인 연결 요구사항을 제공하여 NSURLConnection, CFURL 또는 NSURLSession API을 사용하는 경우 앱이 안전한 연결을 위한 모범 사례를 따르도록 합니다. 기본적으로 앱 전송 보안은 특히 GCM 또는 CBC 모드의 ECDHE_ECDSA_AES 및 ECDHE_RSA_AES와 같이 전방향 안정성을 제공하는 암호 모음만을 선택하도록 암호 선택을 제한합니다. 앱에서는 도메인별로 전방향 안정성 요구사항을 비활성화할 수 있습니다. 비활성화된 경우에는 사용 가능한 암호 세트에 RSA_AES가 추가됩니다.

서버는 반드시 TLS v1.2 및 전방향 안전성을 지원해야 하며 인증서는 반드시 유효한 것으로 SHA-256로 서명되어 있거나 최소 2048비트 RSA 키 또는 256비트 타원곡선 키로 서명된 것이 좋습니다.

앱에서 앱 전송 보안을 오버라이드하지 않는 이상 이러한 요구사항에 부합하지 않는 네트워크 연결은 실패하게 됩니다. 유효하지 않은 인증서를 사용하는 경우 무조건 실패하게 되고 연결을 잃게 됩니다. 앱 전송 보안은 iOS 9 이상 버전용으로 컴파일된 앱에 자동으로 적용됩니다.

VPN

VPN(가상 사설 통신망)과 같은 보안 네트워크 서비스는 일반적으로 최소한의 설정과 구성으로 iOS 기기에서 작동합니다. VPN 서버와 연결된 iOS 기기는 다음과 같은 프로토콜 및 인증 방식을 지원합니다.

- 공유 비밀에서 인증한 IKEv2/IPSec, RSA 인증서, ECDSA 인증서, EAP-MSCHAPv2 또는 EAP-TLS
- App Store의 적절한 클라이언트 앱을 사용한 SSL-VPN
- 암호, RSA SecurID 또는 CRYPTOCard로 사용자가 인증하거나 공유 비밀 및 인증서로 컴퓨터가 인증하는 Cisco IPSec
- MS-CHAPV2 암호, RSA SecurID 또는 CRYPTOCard로 사용자가 인증하거나 공유 비밀로 컴퓨터가 인증하는 L2TP/IPSec

iOS는 다음을 지원합니다.

- 인증서 기반의 인증을 사용하는 네트워크의 **VPN On Demand**를 지원합니다. VPN 구성 프로파일을 사용해 VPN 연결이 요구되는 도메인을 IT 정책에서 명시할 수 있습니다.
- **Per App VPN**을 지원해 VPN 연결을 더욱 전문적으로 활용을 할 수 있습니다. MDM은 Safari에서 각 관리되는 앱 또는 특정 도메인에 대한 연결을 지정할 수 있습니다. 이렇게 하면 사용자의 개인 데이터를 제외한 안전한 데이터가 기업 네트워크로 전송 및 발송될 수 있습니다.
- **Always-on VPN**은 MDM을 통해 관리되고 Apple Configurator 2, 기기 등록 프로그램 또는 Apple School Manager를 통해 감독하는 기기에 구성할 수 있습니다. 이를 통해 사용자는 셀룰러 또는 Wi-Fi 네트워크에 연결할 경우 보안을 활성화하기 위해 VPN을 켤 필요가 없어집니다. Always-on VPN을 사용하는 조직은 모든 IP 트래픽을 조직으로 터널링하여 기기 트래픽을 완전히 제어할 수 있습니다. 기본 터널링 프로토콜인 IKEv2은 데이터 암호화를 통해 트래픽 전송을 보호합니다. 조직에서는 조직의 기기에서 받거나 보내는 트래픽을 모니터링 및 필터링하고 조직 네트워크 내의 데이터를 보호하며 기기의 인터넷 연결을 제한할 수 있습니다.

Wi-Fi

iOS는 기업용 WPA2를 포함한 업계 표준 Wi-Fi 프로토콜을 지원하여 무선 기업 네트워크에 인증된 접근을 제공합니다. 기업용 WPA2는 128비트 AES 암호화를 사용하여 Wi-Fi 네트워크 연결을 통해 통신할 때, 데이터가 보호될 수 있는 가장 높은 수준의 보안 품질을 사용자에게 보장합니다. 802.1X를 지원하여 iOS 기기는 다양하고 폭넓은 RADIUS 인증 환경과 통합됩니다. iPhone 및 iPad는 EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0, PEAPv1 및 LEAP를 포함한 802.1X 무선 인증 방식을 지원합니다.

iOS는 데이터를 보호할 뿐 아니라 802.11w에 속한 보호된 관리 프레임 서비스를 통해 WPA2 수준으로 유니캐스트 및 멀티캐스트 관리 프레임도 보호합니다. iPhone 6 및 iPad Air 2 이후 모델에서는 PMF를 지원합니다.

Wi-Fi 네트워크와 연결되지 않은 상태일 때 Wi-Fi 스캔을 수행하는 경우 iOS는 무작위 MAC(매체 접근 제어) 주소를 사용합니다. 선호하는 Wi-FI 네트워크를 찾아 연결하거나 위치 기반 미리 알림 또는 Apple 지도에서 위치 수정 시 지오펜스를 사용하는 앱에 대한 위치 서비스를 지원하기 위해 스캔을 수행합니다. 선호하는 Wi-Fi 네트워크에 연결하려고 할 때 수행되는 Wi-Fi 스캔은 무작위가 아닙니다.

기기가 Wi-Fi 네트워크와 연결된 상태가 아니며 기기의 프로세서가 잠자기 상태일 때 ePNO (향상된 선호하는 네트워크 오프로드) 스캔을 수행하는 경우 iOS는 무작위 MAC 주소를 사용합니다. ePNO 스캔은 지오펜스를 사용하는 앱에 대해 위치 서비스를 사용하는 경우에 실행됩니다(예를 들어 기기가 특정 장소 근처에 있는지 판단하는 위치 기반 미리 알림).

기기의 MAC 주소는 이제 Wi-Fi 네트워크에 연결 해제된 경우에 변경되기 때문에 기기가 셀룰러 네트워크에 연결되어 있더라도 Wi-Fi 트래픽 관찰자가 기기를 계속해서 추적하기 위해 MAC 주소를 사용할 수 없습니다. Apple은 Wi-Fi 제조업체에 iOS Wi-Fi 스캔은 무작위 MAC 주소를 사용하며 Apple이나 제조업체 모두 이러한 무작위 MAC 주소를 추측할 수 없다고 공지하였습니다. Wi-Fi MAC 주소 무작위 생성 기능은 iPhone 4s 이전 모델에서는 지원되지 않습니다.

iPhone 6s 또는 이후 모델에서는 알려진 Wi-Fi의 가려진 속성이 자동으로 공개되고 업데이트됩니다. Wi-Fi 네트워크의 SSID(서비스 세트 식별자)가 알려지면 iOS 기기는 프로브를 전송할 때 요청 사항에 SSID를 포함하지 않습니다. 이렇게 하면 해당 기기에서 가려지지 않은 네트워크의 이름을 알리지 않게 됩니다.

네트워크 프로세서 펌웨어의 취약성으로부터 기기를 보호하기 위해, Wi-Fi 및 베이스밴드 등의 네트워크 인터페이스는 응용 프로그램 프로세서 메모리에 제한된 접근 권한을 가집니다. 네트워크 프로세서에 접속하는 데 USB 또는 SDIO를 사용하는 경우, 네트워크 프로세서는 응용 프로그램 프로세서에 DMA(직접 메모리 접근) 트랜잭션을 시작할 수 없습니다. PCIe를 사용하는 경우, 각 네트워크 프로세서는 분리된 PCIe 버스에 위치해 있습니다. 각 PCIe 버스에 있는 IOMMU는 네트워크 프로세서의 DMA가 네트워크 패킷 또는 제어 구조가 들어 있는 메모리 페이지에 접근하는 것을 제한합니다.

Bluetooth

iOS의 Bluetooth는 개인 데이터에 대한 불필요한 접근을 늘리지 않으면서도 유용한 기능을 제공하기 위해 설계되었습니다. iOS 기기는 Encryption Mode 3, Security Mode 4 및 Service Level 1 연결을 지원하며 iOS는 다음과 같은 Bluetooth 프로파일을 지원합니다.

- 핸즈프리 프로파일(HFP 1.5)
- 전화번호부 접근 프로파일(PBAP)
- 메시지 액세스 프로파일(MAP)
- 고급 오디오 배포 프로파일(A2DP)
- 오디오/비디오 원격 제어 프로파일(AVRCP)
- 개인 영역 네트워크 프로파일(PAN)
- 휴먼 인터페이스 기기 프로파일(HID)
- 프로파일에 대한 지원은 기기마다 다릅니다.

자세한 정보를 보려면 아래 사이트로 이동하십시오.
support.apple.com/ko-kr/HT204387.

단일 로그인

iOS는 단일 로그인(SSO)을 통해 기업 네트워크 인증을 지원합니다. SSO는 Kerberos 기반 네트워크에서 동작하여 서비스 접근에 허용된 사용자인지 인증합니다. SSO는 Safari 보안 세션에서부터 타사 앱에 이르기까지 수많은 네트워크 활동에 사용될 수 있습니다. 인증서 기반 인증(PKINIT) 또한 지원됩니다.

iOS SSO는 SPNEGO 토큰과 HTTP 합의 프로토콜을 이용하여 Kerberos 기반 인증 게이트웨이 및 Kerberos 티켓을 지원하는 Windows 통합 인증 시스템과 동작합니다. SSO 지원은 오픈 소스 Heimdal 프로젝트를 기반으로 합니다.

다음과 같은 암호화 유형이 지원됩니다.

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

Safari가 SSO를 지원하여 표준 iOS 네트워크 API를 사용하는 타사 앱이 이를 사용해 구성을 설정할 수 있습니다. SSO를 구성하기 위해 iOS는 구성 프로파일 페이로드를 지원하여 MDM 솔루션이 필요한 설정을 내려보낼 수 있도록 허용합니다. 이 설정에는 사용자 계정 이름(Active Directory 사용자 계정을 뜻함) 및 Kerberos 영역 설정과 SSO 사용을 허용할 앱 또는 Safari 웹 URL 구성을 포함합니다.

AirDrop 보안

AirDrop을 지원하는 iOS 기기는 Bluetooth LE와 Apple이 만든 피어 투 피어 Wi-Fi 기술을 사용해 파일 및 정보를 근처 기기(OS X 10.11 이상 버전이 설치된 AirDrop 지원 Mac을 포함하여)에 보낼 수 있습니다. 인터넷 연결 또는 Wi-Fi 액세스 포인트를 사용하지 않고 기기 간에 직접 통신하기 위해 Wi-Fi 무선 통신을 사용합니다.

사용자가 AirDrop을 활성화하면 2048비트 RSA 신원이 기기에 저장됩니다. 추가적으로 사용자의 Apple ID에 등록된 이메일 주소 및 전화번호를 기반으로 AirDrop 신원 해시가 생성됩니다.

사용자가 AirDrop을 사용해 항목을 공유하는 경우 기기는 Bluetooth LE를 통해 AirDrop 신호를 보냅니다. AirDrop이 켜져 있으며 가까운 거리에 있고 잠자기 상태가 아닌 다른 기기는 이 신호를 인식하고 축소된 버전의 신원 해시를 보내 응답합니다.

AirDrop은 '연락처만'이 공유 기본값으로 설정되어 있습니다. 사용자는 AirDrop에서 '모두'로 설정해 모두와 공유하거나 기능을 완전히 끌 수도 있습니다. '연락처만' 모드에서는 수신된 신원 해시를 연락처 앱에 있는 사람의 해시와 비교합니다. 일치하는 연락처가 발견되면 전송하는 기기는 피어 투 피어 Wi-Fi 네트워크를 생성하고 Bonjour를 사용해 AirDrop 연결을 알립니다. 이 연결을 사용하여 수신하는 기기는 전체 신원 해시를 전송하는 기기에 보냅니다. 전체 해시가 연락처의 내용과 일치하는 경우 받는 사람의 이름과 사진(연락처 앱에 있는 경우)이 AirDrop 공유 시트에 표시됩니다.

AirDrop을 사용하는 경우 전송하는 사용자는 항목을 공유하려는 사람을 선택할 수 있습니다. 전송하는 기기에서는 수신하는 기기와 암호화된 연결(TLS)를 시작하여 iCloud 신원 인증서를 교환합니다. 인증서의 신원은 각 기기의 연락처 앱에서 확인됩니다. 그리고 수신하는 기기에서는 신원이 확인된 사람 또는 기기에서 전송된 항목을 수신할지를 확인받습니다. 만약 여러 기기에 전송하기 위해 선택한 경우라면 이 프로세스는 각 기기마다 반복됩니다.

'모두' 모드에서는 같은 프로세스가 사용되지만 연락처에서 일치하는 사람을 찾지 못하는 경우 수신하는 기기는 AirDrop 전송 시트에 설정 > 일반 > 정보 > 이름에서 설정한 기기 이름과 실루엣으로 표시됩니다.

조직에서는 MDM 솔루션을 사용해 관리하는 기기 및 앱의 AirDrop 사용을 제한할 수 있습니다.

Wi-Fi 암호 공유

Wi-Fi 암호 공유를 지원하는 iOS 기기는 AirDrop이 Wi-Fi 암호를 한 기기에서 다른 기기로 전송하는 것과 유사한 메커니즘을 사용합니다.

사용자가 Wi-Fi 네트워크(요청자)를 선택하고 Wi-Fi 암호에 대한 메시지를 받으면 Apple 기기는 Wi-Fi 암호가 필요하다는 Bluetooth LE 알림을 표시합니다. 선택된 Wi-Fi 네트워크의 암호가 있으며 가까운 거리에 있고 잠자기 상태가 아닌 다른 Apple 기기는 Bluetooth LE를 사용하여 요청하는 기기에 연결합니다.

Wi-Fi 암호(수여자)가 있는 기기는 요청자의 연락처 정보를 요구하며, 요청자는 AirDrop과 유사한 메커니즘을 사용하여 신원을 증명해야 합니다. 신원이 증명되면 수여자는 요청자에게 64자 PSK를 전송합니다. 이는 네트워크에 연결할 때에도 사용할 수 있습니다.

조직에서는 MDM 솔루션을 사용해 관리하는 기기 및 앱의 Wi-Fi 암호 공유 사용을 제한할 수 있습니다.

Apple Pay

사용자는 Apple Pay를 사용하여 지원하는 iOS 기기와 Apple Watch에서 간편하고 안전하게 개인 정보를 공개하지 않고 스토어나 앱, Safari를 통해 웹에서 결제할 수 있습니다. 하드웨어 및 소프트웨어에 통합된 보안성을 제공하며 사용이 쉽습니다.

Apple Pay는 또한 사용자 개인 정보를 보호할 수 있도록 디자인되었습니다. Apple Pay는 사용자와 관련이 있을 수 있는 거래 내역 정보를 수집하지 않습니다. 결제 거래는 사용자, 판매처 및 카드 발급처 간에 이루어집니다.

Apple Pay 구성요소

보안 요소: 보안 요소는 업계 표준의 Java Card 플랫폼을 사용하는 인증된 칩으로서 전자 결제에 대한 금융 업계의 요구사항을 준수합니다.

NFC 컨트롤러: NFC 컨트롤러는 NFC(근거리 무선 통신) 프로토콜을 처리하며 응용 프로그램 프로세서와 보안 요소 간의 통신 및 보안 요소와 POS 터미널 간의 통신을 전달합니다.

Wallet: Wallet은 신용 카드, 직불 카드, 적립 카드 및 매장 카드를 추가하고 관리하며 Apple Pay를 사용하여 결제를 진행할 수 있습니다. 사용자는 Wallet에서 사용자의 카드와 카드 발급처, 발급처의 개인정보 취급방침, 최근 거래 내역 등의 추가 정보를 볼 수 있습니다. 사용자는 또한 설정 지원 및 설정 앱의 Apple Pay에서 카드를 추가할 수 있습니다.

보안 엔클레이브: 보안 엔클레이브는 iPhone, iPad 및 Apple Watch에서 인증 프로세스를 관리하며 결제가 진행될 수 있도록 활성화합니다.

Apple Watch에서는 기기가 잠금 해제되어야 하며 사용자가 측면 버튼을 이중 클릭해야 합니다. 이중 클릭이 인식되면 응용 프로그램 프로세서를 직접 통하지 않고 보안 요소 또는 보안 엔클레이브(사용 가능한 경우)로 전달됩니다.

Apple Pay 서버: Apple Pay 서버는 보안 요소에 저장된 기기 계정 번호와 Wallet에 저장된 신용 카드 및 직불 카드의 설정 및 권한 설정을 관리합니다. Apple Pay 서버는 기기 및 결제 네트워크 서버 모두와 통신합니다. Apple Pay 서버는 또한 앱 내 결제의 결제 승인서를 다시 암호화합니다.

Apple Pay가 보안 요소를 사용하는 방법

보안 요소는 Apple Pay를 관리하기 위해 특별히 디자인된 애플릿을 호스트합니다. 또한 결제 네트워크가 승인한 결제 애플릿을 포함합니다. 결제 네트워크 또는 카드 발급처는 키를 사용하여 신용 카드, 직불 카드 또는 선불 카드 데이터를 암호화하고 결제 애플릿으로 전송합니다. 이 키는 결제 네트워크 및 결제 애플릿의 보안 도메인에서만 알고 있습니다. 이 카드 데이터는 결제 애플릿 안에 저장되며 보안 요소의 보안 기능을 사용하여 보호됩니다. 거래 중에는 터미널이 전용 하드웨어 버스의 NFC 컨트롤러를 통해 보안 요소와 직접 통신합니다.

Apple Pay가 NFC 컨트롤러를 사용하는 방법

보안 요소의 게이트웨이인 NFC 컨트롤러는 모든 비접촉식 결제 거래가 기기와 가까운 거리에 있는 POS 터미널을 통해 진행되도록 합니다. NFC 컨트롤러는 범위 내의 터미널이 보내는 결제 요청만을 비접촉식 거래로 인식합니다.

카드 소지자가 Touch ID 또는 암호를 사용하여 결제를 승인하거나 잠금 해제된 Apple Watch에서 측면 버튼을 이중 클릭하여 결제를 승인하면 보안 요소의 결제 애플릿이 준비한 비접촉식 응답이 컨트롤러에 의해 NFC 필드로 단독 전달됩니다. 그 결과 비접촉식 거래의 결제 승인 세부 사항은

로컬 NFC 필드에 포함되어 응용 프로그램 프로세서에 절대 공개되지 않습니다. 그에 반해서 앱 내 결제 및 웹 상 결제의 결제 승인 세부 사항은 응용 프로그램 프로세서에 전달됩니다. 하지만 Apple Pay 서버의 보안 요소가 암호화를 먼저 진행합니다.

신용 카드, 직불 카드 및 선불 카드 권한 설정

사용자가 Apple Pay에 신용 카드, 직불 카드 또는 선불 카드를 추가하는 경우(매장 카드 포함) Apple은 카드 발급처나 카드 발급처의 공인 서비스 제공업체에 카드 정보를 사용자 계정 및 기기와 같은 기타 정보와 함께 안전하게 전송합니다. 이 정보를 사용해 카드 발급처는 Apple Pay에 카드를 추가할 수 있는지를 판단합니다.

Apple Pay에서는 카드 권한 설정 프로세스의 하나로서 다음 세 번의 서버측 요청을 사용해 카드 발급처 또는 네트워크와 통신을 주고 받습니다. 필수 입력 필드, 카드 확인, 링크 및 권한 설정이 필요합니다. 카드 발급처 또는 네트워크는 이러한 요청을 사용해 Apple Pay에 카드를 추가하거나 카드를 확인 및 승인할 수 있습니다. 이러한 클라이언트와 서버 간 세션은 TLS v1.2를 사용해 암호화됩니다.

전체 카드 번호는 기기 또는 Apple 서버에 저장되지 않습니다. 대신에 고유 기기 계정 번호가 생성되어 암호화되며 보안 요소에 저장됩니다. 이 고유 기기 계정 번호는 Apple이 접근할 수 없는 방식으로 암호화되어 있습니다. 기기 계정 번호는 기기마다 고유하고 일반적인 신용 카드 또는 직불 카드 번호와는 달라 카드 발급처에서는 마그네틱 카드, 전화 통화 또는 웹 사이트에서의 사용을 금지할 수 있습니다. 보안 요소에 있는 기기 계정 정보는 iOS 및 watchOS에서 분리되어 있으며 절대로 Apple 서버에 저장되지 않고 iCloud에 백업되지 않습니다.

Apple Watch에서 사용하는 카드는 iPhone의 Apple Watch 앱을 통해 Apple Pay 권한이 설정됩니다. Apple Watch에서 사용하기 위해 카드 권한 설정을 하려면 시계가 Bluetooth 통신 범위 내에 있어야 합니다. Apple Watch에서 사용하기 위해 특별히 등록된 카드는 고유 기기 계정 번호를 가지게 됩니다. 기기 계정 번호는 Apple Watch의 보안 요소 내에 저장됩니다. 다음은 신용 카드, 직불 카드 또는 선불 카드를 Apple Pay에 권한 설정하는 세 가지 방법입니다.

- Apple Pay에 카드를 수동으로 추가하기
- iTunes Store 계정에 저장된 신용 카드 또는 직불 카드를 Apple Pay에 추가하기
- 카드 발급처의 앱을 사용해 카드 추가하기

Apple Pay에 신용 카드 또는 직불 카드를 수동으로 추가하기

카드를 수동으로 추가하려면 권한 설정 프로세스 진행을 위해 매장 카드, 이름, 신용 카드 번호, 만료일 및 CVV 등을 제공해야 합니다. 설정, Wallet 앱 또는 Apple Watch 앱에서 사용자가 직접 입력하거나 기기에 있는 카메라를 사용해 정보를 입력할 수 있습니다. 카메라가 카드 정보를 캡처하면 Apple은 이름, 카드 번호 및 만료일 정보를 채웁니다. 카드 사진은 절대 기기 또는 사진 보관함에 저장되지 않습니다. 모든 필드가 작성되면 카드 확인 프로세스를 통해 CVV를 제외한 필드를 확인합니다. 정보는 암호화되어 Apple Pay 서버로 전송됩니다.

카드 확인 프로세스에서 사용 약관 ID를 반환하면 Apple은 카드 발급처의 사용 약관을 다운로드하여 사용자에게 표시합니다. 사용자가 사용 약관에 동의하면 Apple은 링크 및 권한 설정 프로세스에 CVV 정보와 함께 동의한 약관의 ID를 전송합니다. 추가적으로 링크 및 권한 설정 프로세스의 일환으로 Apple은 기기의 정보를 카드 발급처 또는 네트워크에 공유합니다. 공유되는 정보에는 iTunes 및 App Store 계정 활동에 대한 정보(예: iTunes에서 오래된 거래 내역이 있는지 여부), 기기에 대한 정보(예: 기기의 전화번호, 이름 및 모델 그리고 Apple Pay 설정을 위해 연결되어 있는 iOS 기기 정보), 카드를 추가했을 때 사용자의 대략적인 위치(위치 서비스를 활성화한 경우)가 포함되어 있습니다. 이 정보를 사용해 카드 발급처는 Apple Pay에 카드를 추가할 수 있는지를 판단합니다.

링크 및 권한 설정 프로세스가 완료되면 다음 두 가지 작업이 진행됩니다.

- 기기가 신용 카드 및 직불 카드를 표시하는 Wallet 패스 파일 다운로드를 시작합니다.
- 기기가 카드를 보안 요소로 바인딩합니다.

패스 파일에는 카드 사진을 다운로드할 수 있는 URL, 연락처 정보, 관련 카드 발급처의 앱 및 지원 기능과 같은 카드에 대한 메타데이터가 포함됩니다. 또한 보안 요소 개인화 완료 여부, 카드가 발급처에 의해 정지되었는지의 여부 또는 해당 카드로 Apple Pay를 통해 결제하기 전에 추가 확인이 필요한지 여부 등의 정보를 포함하는 패스 상태도 포함합니다.

iTunes Store 계정에 저장된 신용 카드 또는 직불 카드를 Apple Pay에 추가하기

iTunes에 저장된 신용 카드 또는 직불 카드를 사용하려면 사용자는 Apple ID 암호를 다시 입력하도록 요청받을 수 있습니다. 그리고 카드 번호를 iTunes에서 가져오게 되며 카드 확인 프로세스가 시작됩니다. 카드가 Apple Pay 사용하기에 적합하다면 기기는 사용 약관을 다운로드하고 사용자에게 표시한 다음 약관 ID와 카드 보안 번호를 링크 및 권한 설정 프로세스로 전송합니다. iTunes 계정에 저장된 카드는 추가 확인이 이루어질 수 있습니다.

카드 발급처의 앱을 사용해 신용 카드 또는 직불 카드 추가하기

앱에서 Apple Pay 사용이 가능한 경우 앱과 거래처 서버에 대해 키가 설정됩니다. 이 키는 거래처에 보내는 카드 정보를 암호화하는 데 사용되며 iOS 기기에서 정보를 읽을 수 없도록 방지합니다. 이러한 권한 설정 작업 흐름은 이전에 설명한 수동으로 카드를 추가하는 경우와 비슷하지만 일회용 암호가 CVV 대신에 사용된다는 점이 다릅니다.

추가 확인

카드 발급처는 신용 카드 또는 직불 카드가 추가 확인이 필요한지 여부를 결정할 수 있습니다. 카드 발급처가 제공하는 방법에 따라 사용자는 추가 확인 방법을 선택할 수 있습니다. 예를 들어 문자 메시지, 이메일, 고객 서비스 전화, 또는 승인된 타사 앱을 통한 확인 방법이 있습니다. 문자 메시지나 이메일의 경우 사용자는 카드 발급처가 가지고 있는 연락처 정보에서 선택합니다. 사용자는 전송된 코드를 Wallet, 설정 또는 Apple Watch 앱에 입력합니다. 고객 서비스 전화 또는 앱을 사용한 확인 방법은 카드 발급처에서 직접 프로세스를 진행합니다.

결제 승인

보안 엔클레이브를 사용하는 기기에서 보안 요소는 보안 엔클레이브에서 인증을 받는 경우에만 결제를 허용합니다. iPhone이나 iPad에서는 사용자가 Touch ID, Face ID 또는 기기 암호로 인증하였다는 사실을 확인하는 것도 포함합니다. 사용 가능한 경우 Touch ID 또는 Face ID가 기본 인증 방식으로 사용되지만 암호도 사용할 수 있습니다. 지문 인식 시도를 세 번 실패했거나 얼굴 인식 시도를 두 번 실패한 경우 암호 사용을 제안하며 다섯 번 실패한 경우에는 암호를 요구합니다. 또한 Touch ID 또는 Face ID를 구성하지 않았거나 Apple Pay에서 사용하도록 설정하지 않은 경우에도 암호를 요구합니다. Apple Watch에서는 암호를 입력해 기기를 잠금 해제한 다음 측면 버튼을 이중 클릭하여 결제를 허용해야 합니다.

보안 엔클레이브와 보안 요소 간의 통신은 시리얼 인터페이스를 통해 이루어지는데 보안 요소는 NFC 컨트롤러에 연결되고 컨트롤러는 응용 프로그램 프로세서와 연결된 상태여야 합니다. 직접 연결되지는 않지만 보안 엔클레이브와 보안 요소는 제조 과정에서 권한이 설정된 공유 페어링 키를 사용해 안전하게 통신합니다. 통신의 암호화 및 인증은 AES를 기반으로 하며 재전송 공격을 방지하기 위해 양쪽에서 사용한 암호화 nonce도 사용됩니다. 페어링 키는 UID키의 보안 엔클레이브와 보안 요소의 고유 식별자 내에서 생성됩니다. 그런 다음 페어링 키는 제조 과정 중에 보안 엔클레이브에서 하드웨어 보안 모듈(HSM)로 안전하게 전송됩니다. HSM은 페어링 키를 보안 요소에 삽입하는 데 필요한 중요 자료를 가지고 있습니다.

사용자가 거래를 승인하는 경우 보안 엔클레이브는 인증 유형에 대해 서명된 데이터와 거래 유형에 대한 세부 사항(비접촉식 또는 앱 내 결제)을 권한 무작위(AR: Authorization Random) 값과 연관된 보안 요소로 전송합니다. 사용자가 처음으로 신용 카드 권한을 설정하고 Apple Pay까지 활성화하는 경우 보안 엔클레이브에서 AR을 생성하며 보안 엔클레이브의 암호화 및 안티 롤백 메커니즘으로 보호합니다. 또한 페어링 키를 통해 보안 요소로 안전하게 전송됩니다. 새로운 AR 값을 받게 되는 경우 보안 요소는 이전에 추가한 카드를 삭제됨으로 표시합니다.

보안 요소에 추가된 신용 카드, 직불 카드 및 선불 카드는 카드를 추가한 때와 동일한 페어링 키와 AR 값을 사용하여 보안 요소가 승인된 경우에만 사용할 수 있습니다. 이렇게 하여 다음과 같은 시나리오에서 iOS가 보안 엔클레이브의 AR 복사본을 유효하지 않음으로 표시하여 보안 엔클레이브에 카드 사용이 불가능하다고 전달합니다.

- 암호가 비활성화됨
- 사용자가 iCloud에서 로그아웃
- 사용자가 모든 콘텐츠 및 설정 지우기를 선택
- 기기가 복구 모드에서 복원됨

Apple Watch에서 카드가 유효하지 않음으로 표시되는 경우

- 시계의 암호가 비활성화됨
- 시계가 iPhone과 연결 해제됨
- 손목 인식이 꺼짐

페어링 키와 현재 AR 값의 복사본을 사용하여 보안 요소는 보안 엔클레이브에서 받은 승인을 확인한 후 비접촉식 결제를 위한 결제 애플릿을 활성화합니다. 또한 이 프로세스는 암호화된 결제 데이터를 앱 내 거래의 결제 애플릿에서 가져오는 경우에도 적용됩니다.

특정 거래 동적 보안 코드

결제 애플릿에서 생성된 모든 결제 거래는 특정 거래 동적 보안 코드와 기기 계정 번호를 포함합니다. 이러한 일회용 코드는 카운터와 키를 사용해 계산됩니다. 카운터는 새로운 거래가 있는 경우 증가하는 값이며, 키는 개인화 중에 결제 애플릿에서 권한이 설정되어 결제 네트워크 또는 카드 발급처에서 알고 있는 값입니다. 이러한 코드를 계산하는 데 결제 방법에 따라 다음과 같은 다른 데이터도 사용될 수 있습니다.

- 결제 애플릿에서 생성한 무작위 숫자
- NFC 거래의 경우 터미널에서 생성한 또 다른 무작위 숫자

  또는

- 앱 내 거래의 경우 서버에서 생성한 또 다른 무작위 숫자

이러한 보안 코드는 결제 네트워크 및 카드 발급처에 제공되어 거래를 확인하는 데 도움을 줍니다. 보안 코드의 길이는 처리되는 거래의 유형에 따라 다를 수 있습니다.

Apple Pay를 사용한 비접촉식 결제

iPhone이 켜져 있고 NFC 필드를 인식하는 경우 사용자에게 적절한 신용 카드, 직불 카드, 선불 카드 또는 설정 앱에서 관리되는 기본 카드가 제시됩니다. 사용자는 Wallet 앱으로 이동해 신용 카드 또는 직불 카드를 선택할 수도 있고 기기가 잠겨 있는 경우에는 홈 버튼을 이중 클릭하여 선택할 수도 있습니다.

그런 다음, 사용자는 Touch ID, Face ID 또는 암호를 사용해 인증하여 결제 정보를 전달합니다. Apple Watch가 잠금 해제된 경우 측면 버튼을 이중 클릭하면 결제용 기본 카드가 활성화됩니다. 사용자가 승인하기 전에는 어떠한 결제 정보도 전송되지 않습니다. 사용자가 승인하게 되면 기기 계정 번호 및 특정 거래 동적 보안 코드를 사용하여 결제가 진행됩니다. Apple 또는 사용자의 기기는 실제 신용 카드 및 직불 카드 번호를 거래처에 보내지 않습니다. Apple은 거래가 이루어진 대략적인 시간 및 장소와 같은 거래 정보를 익명으로 받을 수 있으며 이를 통해 Apple Pay 및 다른 Apple 제품과 서비스를 개선합니다.

앱 내에서 Apple Pay로 결제하기

Apple Pay는 iOS 앱 내에서 결제하거나 watchOS 3에서 작동하는 Apple Watch 앱 내에서 결제할 때에도 사용할 수 있습니다. 사용자가 앱 내에서 Apple Pay를 사용해 결제하면 Apple은 암호화된 거래 정보를 받고 개발사 특정 키로 그 정보를 다시 암호화하여 개발사 또는 거래처에 보냅니다. Apple Pay는 추정 결제 금액과 같은 거래 정보를 익명으로 유지합니다. 이러한 정보를 통해서는 사용자의 신원을 확인할 수 없으며 사용자의 구매 항목에 대한 정보는 절대 포함되지 않습니다.

앱이 Apple Pay 결제 거래를 개시하면 거래처보다 먼저 Apple Pay 서버가 기기로부터 암호화된 거래 정보를 받습니다. 그리고 Apple Pay는 거래 정보를 거래처 특정 키로 다시 암호화하고 거래처에 거래 정보를 릴레이합니다.

앱에서 결제를 요청하는 경우 API를 호출하여 기기가 Apple Pay를 지원하는지 여부와 거래처에서 승인하는 결제 네트워크에서 사용할 수 있는 신용 카드 또는 직불 카드를 사용자가 가지고 있는지 여부를 판단합니다. 그리고 앱은 거래를 진행하고 완료하기 위한 모든 정보를 요청하는데 청구 주소, 배송 주소 및 연락처 등의 정보가 이에 포함됩니다. 앱은 또한 iOS에 Apple Pay 시트를 요청하여 사용해야 하는 카드 등의 필요한 정보를 요청합니다.

이 단계에서 앱은 시/도, 우편번호 정보를 받아 최종 배송비를 계산합니다. 하지만 사용자가 Touch ID, Face ID 또는 기기 암호로 결제를 승인하기 전에는 요청한 모든 정보가 제공되지 않습니다. 결제가 승인된 경우 Apple Pay 시트에 표시된 정보는 거래처에 전송됩니다.

사용자가 결제를 승인하면 Apple Pay 서버에 요청을 보내 암호화 nonce를 받습니다. 이는 매장 내 거래에서 사용된 NFC 터미널이 반환한 값과 비슷한 값입니다. 다른 거래 데이터와 함께 nonce는 보안 요소로 전송되어 결제 승인서를 생성하며 승인서는 Apple 키를 통해 암호화됩니다. 암호화된 결제 승인서가 보안 요소에서 발견되면 Apple Pay 서버로 전송되어 암호화가 해제되고 인증서의 nonce를 보안 요소가 보낸 nonce와 비교하여 확인한 다음 결제 승인서를 거래처 ID의 거래처 키를 통해 다시 암호화합니다. 암호화된 승인서는 다시 기기로 전송되고 API를 통해 앱에 전달됩니다. 그리고 앱은 거래 진행을 위해 거래처 시스템에 승인서를 보냅니다. 승인서를 받은 거래처는 개인 키를 사용해 암호화를 해제하고 거래를 진행합니다. 이 부분에서 거래처는 또한 Apple 서버의 서명을 통해 해당 거래가 자신을 위한 것인지 확인할 수 있습니다.

API는 지원하는 거래처 ID를 명시하는 권한을 요구합니다. 앱은 또한 서명을 위해 보안 요소에 보낼 주문 번호 또는 고객 신원과 같은 추가 데이터를 포함시켜 거래가 다른 고객에게 넘어가지 않도록 합니다. 이 작업은 PKPaymentRequest에 applicationData를 명시할 수 있는 앱 개발자가 수행합니다. 이 데이터의 해시는 암호화된 결제 데이터에 포함되어 있습니다. 그래서 거래처는 자신의 applicationData 해시가 결제 데이터의 정보와 일치하는지를 확인해야 합니다.

웹 또는 Handoff를 통해 Apple Pay로 결제하기

Apple Pay는 웹 사이트에서 결제할 때에도 사용할 수 있습니다. iOS 10 이상이 설치된 iPhone과 iPad는 웹에서 Apple Pay로 결제가 가능합니다. 또한, macOS Sierra 이상의 경우 Mac에서 Apply Pay 결제를 시작하고 동일한 iCloud 계정을 사용하여 Apple Pay가 활성화된 iPhone이나 Apple Watch에서 결제를 완료할 수 있습니다.

웹에서 Apple Pay를 사용하려면 웹 사이트가 Apple에 등록되어야 합니다. Apple 서버에서 도메인 이름 확인을 수행하고 TLS 클라이언트 인증서를 발행합니다. Apple Pay를 지원하는 웹 사이트는 HTTPS를 통해 콘텐츠를 제공해야 합니다. 웹 사이트는 결제 거래가 있을 때마다 Apple이 발행한 TLS 클라이언트 인증서를 사용하여 Apple 서버로 안전하고 고유한 거래처 세션을 얻어야 합니다. 거래처 세션 데이터는 Apple에서 서명합니다. 거래처 세션 서명이 확인되면 웹 사이트에서 사용자가 Apple Pay가 활성화된 기기를 사용하는지 여부와 신용 카드, 직불 카드 또는 선불 카드가 기기에 활성화되어 있는지 여부를 확인할 수 있습니다. 다른 세부 사항은 공유되지 않습니다. 사용자가 이러한 정보를 공유하는 것을 원하지 않는다면 iOS 및 macOS의 Safari 개인 정보 보호 설정에서 Apple Pay 확인을 비활성화할 수 있습니다.

거래처 세션이 확인되면 사용자가 앱 내에서 결제할 때와 동일한 보안 및 개인 정보 보호 방안이 사용됩니다.

Mac에서 iPhone이나 Apple Watch로 Handoff하는 경우 Apple Pay는 암호화된 엔드 투 엔드 IDS 프로토콜을 사용하여 결제 관련 정보를 사용자의 Mac에서 인증을 시도하는 기기로 전송합니다. IDS는 사용자의 기기 키를 사용하여 암호화를 수행하기 때문에 다른 기기에서는 결제 관련 정보의 암호화를 해제할 수 없으며 해당 키는 Apple에 공유되지 않습니다. Apple Pay Handoff 기기를 발견하는 절차는 일부 메타데이터와 함께 사용자 신용 카드의 유형과 고유 ID 확인을 포함합니다. 사용자 카드의 기기별 고유 계정 번호는 공유되지 않으며 사용자의 iPhone 또는 Apple Watch에 계속 안전하게 보관됩니다. Apple은 또한 사용자가 최근에 사용한 연락처, 배송 및 청구 주소를 iCloud 키체인을 통해 안전하게 전송합니다.

사용자가 iPhone의 Touch ID, Face ID 또는 암호를 사용하거나 Apple Watch의 측면 버튼을 이중 클릭하여 결제를 승인하면 웹 사이트 거래처별 인증서에 맞추어 암호화된 결제 토큰이 사용자의 iPhone이나 Apple Watch에서 Mac으로 안전하게 전송된 다음 거래처 웹 사이트로 전달됩니다.

기기는 가까운 거리에 있어야 결제를 요청하고 완료할 수 있습니다. Bluetooth LE 광고를 통해 거리를 확인합니다.

적립 카드

iOS 9 이상에서 Apple Pay는 거래처 적립 카드의 호환되는 NFC 터미널 전송에 대한 부가 가치 서비스(VAS) 프로토콜을 지원합니다. VAS 프로토콜은 거래처 터미널에 구현할 수 있으며 NFC를 사용해 지원되는 Apple 기기와 통신합니다. VAS 프로토콜은 짧은 거리에서 작동하며 Apple Pay 거래의 한 부분으로서 적립 카드 정보 전송 등의 고객 관련 서비스를 제공합니다.

NFC 터미널은 카드 요청을 보내 카드 정보를 수신하도록 시작할 수 있습니다. 만약 사용자가 매장 식별자가 있는 카드를 가지고 있다면 카드 사용 승인을 요청받습니다. 거래처에서 암호화를 지원하는 경우 카드 정보, 타임스탬프 및 일회용 무작위 ECDH P-256 키가 거래처의 공개 키와 함께 사용되어 터미널로 전송되는 카드 데이터의 암호화 키를 파생합니다. 만약 거래처가 암호화를 지원하지 않는다면 적립 카드 정보가 전송되기 전에 기기를 터미널 근처로 다시 가져가도록 사용자에게 요청합니다.

Apple Pay Cash

iOS 11.2 및 watchOS 4.2부터 iPhone, iPad, Apple Watch에서 Apple Pay를 통해 다른 사용자와 돈을 보내거나 받고, 송금을 요청할 수 있습니다. 사용자가 돈을 받으면 Apple Pay Cash 계좌에 해당 금액이 추가되며, 승인된 모든 기기에서 사용자의 Apple ID로 로그인하여 Wallet 앱이나 설정 > Wallet 및 Apple Pay에서 금액을 확인할 수 있습니다.

개인 간 거래 및 Apple Pay Cash를 사용하려면 사용자는 Apple Pay Cash 호환 기기에서 iCloud 계정으로 로그인해야 하며, iCloud 계정에 이중 인증을 설정해 두어야 합니다.

Apple Pay Cash를 설정하면 신용 카드 및 직불 카드를 등록했을 때와 동일한 정보가 Apple 제휴 은행인 Green Dot Bank와 공유되고, 다른 Apple 부서에서는 알 수 없도록 따로 정보를 보관하고 처리함으로써 사용자의 개인 정보를 보호하기 위해 설립된 Apple의 자회사 Apple Payments Inc.와 공유될 수 있습니다. 이 정보는 문제 해결, 사기 방지 및 규제 목적으로만 사용됩니다.

메시지 앱을 사용하거나 Siri에게 요청하여 사용자 간에 송금을 요청하고 이체할 수 있습니다. 사용자가 송금을 시도할 때 iMessage는 Apple Pay 시트를 표시합니다. 항상 Apple Pay Cash 잔액이 가장 먼저 사용됩니다. 필요한 경우 추가 금액은 사용자의 Wallet에 등록된 두 번째 신용 카드 또는 직불 카드에서 가져옵니다.

Wallet의 Apple Pay Cash 카드는 Apple Pay를 통해 스토어, 앱 및 웹에서 결제 시 사용할 수 있습니다. 또한 Apple Pay Cash 계정에 있는 금액을 은행 계좌로 이체할 수 있습니다. 다른 사용자로부터 송금을 받을 수 있을 뿐만 아니라, Wallet의 직불 카드 또는 선불 카드에서 Apple Pay Cash 계정으로 입금할 수 있습니다.

거래가 완료되면 Apple Payments Inc.는 문제 해결, 사기 방지 및 규제 등을 위해 사용자의 거래 데이터를 보관 및 사용할 수 있습니다. 다른 Apple 부서는 사용자가 누구에게 송금을 했는지, 누구로부터 송금을 받았는지 또는 Apple Pay Cash 카드로 어디서 구입을 진행했는지 등의 정보에 대해 알 수 없습니다.

사용자가 Apple Pay로 송금하거나, Apple Pay Cash 계정에 입금하거나, 은행 계좌로 이체할 때 Apple Pay 서버에 요청을 보내 암호화 nonce를 받습니다. 이는 앱 내에서 Apple Pay에 반환한 값과 비슷한 값입니다. 다른 거래 데이터와 함께 nonce는 보안 요소로 전송되어 결제 서명을 생성합니다. 결제 서명이 보안 요소에서 발견되면 Apple Pay 서버로 전달됩니다. Apple Pay 서버는 nonce 및 결제 서명을 통해 거래의 인증, 무결성 및 정확성을 확인합니다. 확인 후 금액이 이체되며 사용자에게 거래가 완료되었음을 알립니다.

다음의 경우 거래에 신용 카드 또는 직불 카드가 사용될 수 있습니다.

- Apple Pay Cash에 입금
- 다른 사용자에게 송금
- Apple Pay Cash 잔액이 부족한 경우 보조 결제 수단으로 사용

그런 다음, 위에서 설명한 이체 서명에 더하여 암호화된 결제 승인서도 생성되고 Apple Pay 서버에 전송됩니다. 이는 앱 및 웹 사이트에서 Apple Pay로 결제할 때와 유사합니다.

Apple Pay Cash 계정의 잔액이 특정 금액을 초과하거나 비정상적 활동이 감지되면 사용자에게 신원을 확인하라는 메시지가 표시됩니다. 사회 보장 번호 또는 질문에 답변하기(예: 예전에 살았던 곳의 도로 주소)와 같이 사용자의 신원을 확인하기 위해 제공된 정보는 Apple 파트너사에 안전하게 전송되고 파트너사의 키를 사용하여 암호화됩니다. Apple은 이 데이터의 암호화를 해제할 수 없습니다.

Suica 카드

일본의 경우 지원되는 iPhone 및 Apple Watch 모델에서 Apple Pay Wallet에 Suica 카드를 등록할 수 있습니다. 실물 카드에서 디지털 Wallet으로 금액 및 정기권을 전송하거나, Suica 앱에서 Wallet에 새로운 Suica 카드의 권한을 설정할 수 있습니다. Wallet에 Suica 카드를 등록하고 나면 사용자는 자신의 무기명 Suica 카드, MySuica 또는 정기권이 포함된 카드로 매장에서 결제하거나 대중교통을 이용할 수 있습니다.

등록된 Suica 카드는 사용자의 iCloud 계정에 연결됩니다. 사용자가 Wallet에 두 장 이상의 카드를 등록하는 경우, Apple 또는 승차권 발급처에서 사용자의 개인 정보 및 카드 간 계정 정보를 연결하는 게 가능할 수 있습니다. 예를 들어 MySuica 카드가 무기명 Suica 카드에 연결될 수 있습니다. Suica 카드 및 거래는 일련의 계층형 암호화 키로 보호됩니다.

실물 카드에서 Wallet으로 금액을 전송하는 과정에서 사용한 카드가 무기명 Suica 카드인 경우 사용자는 카드 일련번호의 마지막 네 자리를 입력해야 합니다. MySuica 카드 또는 정기권이 포함된 카드인 경우, 사용자는 생년월일을 입력하여 카드 소유자임을 증명해야 합니다. iPhone에서 Apple Watch로 승차권을 전송할 때 두 기기 모두 온라인 상태여야 합니다.

Wallet 또는 Suica 앱을 통해 신용 카드 또는 선불 카드로 잔액을 충전할 수 있습니다. Apple Pay를 사용 시 잔액을 다시 불러올 때의 보안에 관한 내용은 이 도큐멘트의 '앱 내에서 Apple Pay로 결제하기' 섹션에 설명되어 있습니다.

Suica 앱에서 Suica 카드 권한 설정 과정에 관한 내용은 이 도큐멘트의 '카드 발급처의 앱을 사용해 신용 카드 또는 직불 카드 추가하기' 섹션에 설명되어 있습니다.

승차권 발급처는 실물 카드를 인증하고 사용자가 입력한 데이터를 확인하기 위한 암호화 키를 가집니다. 확인이 끝나면 시스템은 보안 요소에 대한 기기 계정 번호를 생성할 수 있으며 Wallet에 잔액이 전송되고 새로 추가된 패스를 활성화할 수 있습니다. 실물 카드에서 Wallet으로 권한 설정을 마치면 실물 카드는 사용할 수 없습니다.

각 권한 설정의 마무리 단계에서 Suica 잔액은 암호화되어, 보안 요소의 지정된 애플릿에 저장됩니다. 승차권 발급처는 잔액 거래를 위해 카드 데이터에 암호화 작업을 수행하는 키를 가집니다.

기본적으로 사용자는 빠른 승차 기능을 통해 요금 충전 또는 탑승 시 Touch ID, Face ID 및 암호 요청없이 빠르게 사용할 수 있습니다. Express Mode가 활성화되어 있는 근처 비접촉식 카드 리더기의 경우 최근 방문한 역, 거래 내역, 추가 티켓 등의 정보에 접근할 수 있습니다. Wallet 및 Apple Pay 설정에서 빠른 승차 기능을 비활성화하여 Touch ID, Face ID 또는 암호 인증을 요청하도록 변경할 수 있습니다.

다음과 같은 방법으로 사용자는 다른 Apple Pay 카드로 Suica 카드를 정지하거나 제거할 수 있습니다.

- 나의 iPhone 찾기를 사용하여 원격으로 기기 지우기
- 나의 iPhone 찾기에서 분실 모드 활성화
- MDM 원격 지우기 작동
- Apple ID 계정 페이지에서 모든 카드 제거
- iCloud.com에서 모든 카드 제거
- Wallet에서 모든 카드 제거

Apple Pay 서버는 정기권 발급처에 Suica 카드 비활성화를 고지합니다. 기기를 지우려고 시도했을 때 해당 기기가 오프라인 상태면 익일 오전 12시 1분(일본 표준시)이 되기 전까지는 일부 터미널에서 해당 Suica 카드 사용이 가능할 수 있습니다.

사용자가 Suica 카드를 제거하더라도 잔액을 복구할 수 있습니다. 해당 잔액을 익일 오전 5시 (일본 표준시) 이후에 동일한 Apple ID를 사용하여 서명된 기기에 추가할 수 있습니다.

기기가 오프라인인 경우에는 Suica 카드를 정지할 수 없습니다.

카드 정지, 제거 및 삭제하기

사용자는 나의 iPhone 찾기를 사용해 기기를 분실 모드로 설정하여 iPhone, iPad 및 watchOS 3이 설치된 Apple Watch에서 Apple Pay 사용을 정지시킬 수 있습니다. 사용자는 또한 나의 iPhone 찾기를 사용하거나 iCloud.com을 방문하거나 Wallet을 사용해 기기에서 직접 Apple Pay의 카드를 제거 및 삭제할 수 있습니다. 카드는 Apple Watch에서 직접 제거하거나 iCloud 설정 또는 iPhone의 Apple Watch 앱에서 제거할 수 있습니다. 기기에 저장된 카드를 사용하여 결제하는 기능이 정지될 수 있고 셀룰러 또는 Wi-Fi 네트워크에 연결되어 있지 않고 기기가 오프라인 상태에서도 카드 발급처 또는 관련 결제 네트워크에 의해 카드가 Apple Pay에서 지워질 수 있습니다. 사용자는 또한 카드 발급처에 연락하여 Apple Pay의 카드를 정지하거나 지울 수 있습니다.

추가적으로 사용자가 '모든 콘텐츠 및 설정 지우기'를 하거나 나의 iPhone 찾기에서 기기를 지우거나, 복구 모드에서 기기를 복원하여 기기 전체가 지워지는 경우 iOS는 보안 요소로 명령을 보내 모든 카드를 삭제됨으로 표시합니다. 삭제됨으로 표시하면 카드는 즉시 사용 불가능 상태로 변경되고, Apple Pay 서버에 연결되면 보안 요소에서 카드가 완전히 삭제됩니다. 보안 엔클레이브는 별도로 AR을 유효하지 않음으로 표시하여 이미 등록한 카드에 대한 결제 승인을 불가능하게 만듭니다. 기기가 온라인 상태가 되면 기기는 Apple Pay 서버로 접근을 시도하여 보안 요소에 있는 모든 카드가 삭제된 것을 확인합니다.

인터넷 서비스

강력한 Apple ID 암호 생성하기
Apple ID는 iCloud, FaceTime 및 iMessage를 포함한 여러 서비스에 연결하는 데 사용됩니다. 사용자가 강력한 암호를 생성할 수 있도록 새로운 계정은 다음의 암호 속성을 포함해야 합니다.

- 최소 8자
- 문자 최소 한 개
- 대문자 최소 한 개
- 숫자 최소 한 개
- 3자 이상의 동일한 문자를 연속으로 포함하지 않음
- 계정 이름과 동일하지 않아야 함

Apple은 iMessage, FaceTime, Siri 제안, iCloud, iCloud 백업 및 iCloud 키체인 등의 강력한 서비스를 구축하여 사용자들이 기기를 더 잘 활용하고 생산성을 높일 수 있도록 도와줍니다.

이러한 인터넷 서비스는 iOS가 플랫폼 전반에서 지키려고 하는 하나의 보안 목표를 기반으로 구축되었습니다. 보안 목표에는 기기에 저장된 데이터의 안전한 처리, 무선 네트워크를 통해 전송되는 데이터의 안전한 처리, 사용자의 개인 정보 보호, 정보와 서비스에 대한 악의적이거나 인증되지 않은 접근 방지 등이 포함됩니다. 각 서비스는 전반적인 iOS의 사용성을 저하시키지 않는 강력한 보안 아키텍처를 사용합니다.

Apple ID

Apple ID는 iCloud, iMessage, FaceTime, iTunes Store, iBooks Store, App Store 등과 같은 Apple 서비스에 로그인하는 데 사용하는 계정입니다. 사용자가 Apple ID를 안전하게 보호하여 해당 계정에 인증되지 않은 접근을 방지하는 것이 중요합니다. 이에 도움이 되도록 Apple은 강력한 암호를 요구합니다. 암호의 길이가 최소 8자이어야 하고 문자와 숫자를 둘 다 포함해야 하며 3자 이상의 동일한 연속 문자를 포함하지 않아야 하고 일반적으로 사용되는 암호를 사용할 수 없습니다. 사용자는 이 기준을 넘어서 문자와 구두점을 더 추가해 암호를 강화할 수 있습니다. Apple은 또한 사용자에게 세 가지 보안 질문을 설정하도록 요청합니다. 보안 질문을 통해 사용자가 계정 정보를 변경하거나 잊어버린 암호를 재설정하려는 경우 계정 소유자의 신원을 확인합니다.

또한 Apple은 해당 계정에 중요한 변경사항이 적용되는 경우 사용자에게 이메일 및 푸시 알림을 보냅니다. 예를 들어, 암호나 청구 정보가 변경된 경우 또는 Apple ID가 새로운 기기에 로그인하는 데 사용된 경우가 있습니다. 모르는 내용이 있는 경우 사용자는 Apple ID 암호를 즉시 변경하도록 안내를 받게 됩니다.

추가로 Apple은 사용자의 계정을 보호하기 위해 고안된 다양한 정책과 절차를 채용합니다. 여기에는 로그인 재시도 횟수 제한과 암호 재설정 횟수 제한, 해킹 발생 시 이를 확인하기 위한 적극적인 사기 방지 모니터링 및 고객 보안에 영향을 미치는 새로운 정보에 맞추어 정책을 조정하기 위한 Apple의 주기적인 정책 검토가 포함됩니다.

이중 인증

사용자의 계정을 더욱 안전하게 보호하기 위해서 Apple에서는 Apple ID를 한 번 더 보호하는 이중 인증을 제공합니다. 이중 인증은 다른 사람이 암호를 알더라도 계정 소유자만 계정에 접근할 수 있도록 개발되었습니다.

이중 인증을 사용하면 사용자의 iPhone, iPad 또는 Mac과 같이 신뢰할 수 있는 기기에서만 사용자의 계정을 사용할 수 있습니다. 새로운 기기에서 처음으로 로그인하려는 경우 두 가지 정보가 필요한데, 하나는 Apple ID 암호이고, 다른 하나는 사용자의 신뢰할 수 있는 기기에 자동으로 표시되거나 신뢰할 수 있는 전화번호로 발송되는 6자리 확인 코드입니다. 확인 코드를 입력하면 사용자는 새로운 기기를 신뢰하며 새로운 기기가 로그인하기에 안전하다고 확인하게 됩니다. 암호만으로는 사용자의 계정에 접근하지 못하므로 이중 인증은 사용자의 Apple ID와 사용자가 Apple에 저장한 모든 개인 정보에 대한 보안을 향상합니다. 이중 인증은 또한 iOS, macOS, tvOS, watchOS 및 Apple 웹 사이트에서 사용하는 인증 시스템에 직접 통합되어 있습니다.

이중 인증에 관한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
support.apple.com/ko-kr/HT204915.

2단계 확인

2013년부터 Apple에서는 2단계 확인이라는 유사한 보안 방식도 제공하고 있습니다. 2단계 확인을 활성화하면 사용자는 신뢰할 수 있는 기기로 전송된 임시 코드를 통해 신원을 확인해야 합니다. 신원이 확인되지 않으면 Apple ID 계정 정보에 대한 변경사항이 허용되지 않거나 iCloud, iMessage, FaceTime 또는 Game Center에 로그인을 할 수 없고 또한 새로운 기기에서 iTunes Store, iBooks Store 또는 App Store의 항목을 구매할 수 없습니다. 또한 암호를 잊었거나 신뢰할 수 있는 기기에 접근하지 못하는 경우에 사용할 수 있도록 14자의 복구 키가 제공되며 사용자는 이를 안전한 장소에 보관해야 합니다. 대부분의 새로운 사용자에게 이중 인증을 권장하는 반면 일부 경우에는 2단계 확인을 사용하도록 권장합니다.

Apple ID 이중 인증에 관한 자세한 내용을 보려면 아래 사이트로 이동하십시오.
support.apple.com/ko-kr/HT204152

관리되는 Apple ID

관리되는 Apple ID는 Apple ID와 유사한 기능을 가지고 있지만 교육 기관에서 소유하고 제어합니다. 해당 기관에서는 암호를 재설정하거나 구입을 제한하고 FaceTime 및 메시지와 같은 통신을 제한하거나 교직원, 교사 및 학생에 대한 역할별 권한을 설정할 수 있습니다.

관리되는 Apple ID를 사용하는 경우, Apple Pay, iCloud 키체인, HomeKit 및 나의 iPhone 찾기 등의 일부 Apple 서비스를 사용할 수 없게 됩니다.

관리되는 Apple ID에 대한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
help.apple.com/schoolmanager

관리되는 Apple ID 감사하기

관리되는 Apple ID는 감사를 지원하여 교육 기관에서 법적 규정과 개인 정보 보호 규정을 준수하도록 합니다. 임직원, 교사 또는 관리자의 계정은 특정 관리되는 Apple ID에 대한 감사 권한을 가질 수 있습니다. 감사원은 학교 조직 체계에 따라 자신의 아래에 속한 계정만 모니터링할 수 있습니다. 강사는 학생을 모니터링할 수 있고, 관리자는 강사와 학생을 감사할 수 있고, 임직원은 관리자, 강사, 학생을 감사할 수 있는 방식입니다.

감사를 하는 경우 Apple School Manager에서 인증서가 요청되고 감사가 요청된 관리되는 Apple ID에만 접근 가능한 특수 계정이 발급됩니다. 감사 권한은 7일 후에 만료됩니다. 해당 기간 동안 감사원은 사용자가 iCloud 또는 CloudKit가 활성화된 앱에 저장한 콘텐츠를 읽거나 수정할 수 있습니다. 모든 감사 접근 요청은 Apple School Manager에 기록됩니다. 기록에는 감사원의 신원 정보, 감사원이 접근을 요청한 관리되는 Apple ID, 요청 시간 및 감사의 시행 여부가 나타납니다.

관리되는 Apple ID 및 개인 기기

또한, 관리되는 Apple ID는 개인이 소유한 iOS 기기 및 Mac 컴퓨터에서도 사용할 수 있습니다. 학생은 교육 기관에서 발행한 관리되는 Apple ID와 Apple ID의 이중 인증 절차에서 2차 인증 확인 역할을 하는 가정용 추가 암호를 사용하여 iCloud에 로그인합니다. 관리되는 Apple ID를 개인 기기에 사용하는 경우 iCloud 키체인을 사용할 수 없으며 해당 교육 기관에서 FaceTime이나 메시지와 같은 기타 기능을 제한할 수도 있습니다. 학생이 로그인하여 생성한 모든 iCloud 도큐멘트는 이 섹션에서 위에 설명된 바와 같은 감사의 대상이 됩니다.

iMessage

Apple의 iMessage는 iOS 기기, Apple Watch 및 Mac 컴퓨터를 위한 메시지 전송 서비스입니다. iMessage는 문자, 사진, 연락처 및 위치와 같은 첨부 파일을 지원합니다. 등록된 모든 기기에 메시지가 나타나므로 사용자는 어떤 기기에서도 대화를 이어갈 수 있습니다. iMessage는 APNS(Apple 푸시 알림 서비스)를 폭넓게 사용합니다. Apple은 메시지 또는 첨부 파일 내용을 기록하지 않으며 해당 콘텐츠는 엔드 투 엔드 암호화 기술로 안전하게 보호되어 보낸 사람과 받는 사람을 제외한 누구도 해당 콘텐츠에 접근할 수 없습니다. Apple은 해당 데이터의 암호화를 해제할 수 없습니다.

사용자가 기기에서 iMessage를 켜면 해당 서비스에서 사용할 수 있는 두 쌍의 키가 생성됩니다. 두 쌍의 키는 각각 암호화용인 RSA 1280비트 키와 서명용인 NIST P-256 커브의 ECDSA 256비트 키입니다. 두 쌍의 키에 대한 개인 키는 기기의 키체인에 저장되고 공개 키는 Apple의 디렉토리 서비스(IDS)로 전송됩니다. IDS에 전송된 공개 키는 해당 기기의 APNS 주소, 사용자의 전화번호 또는 이메일 주소와 연결됩니다.

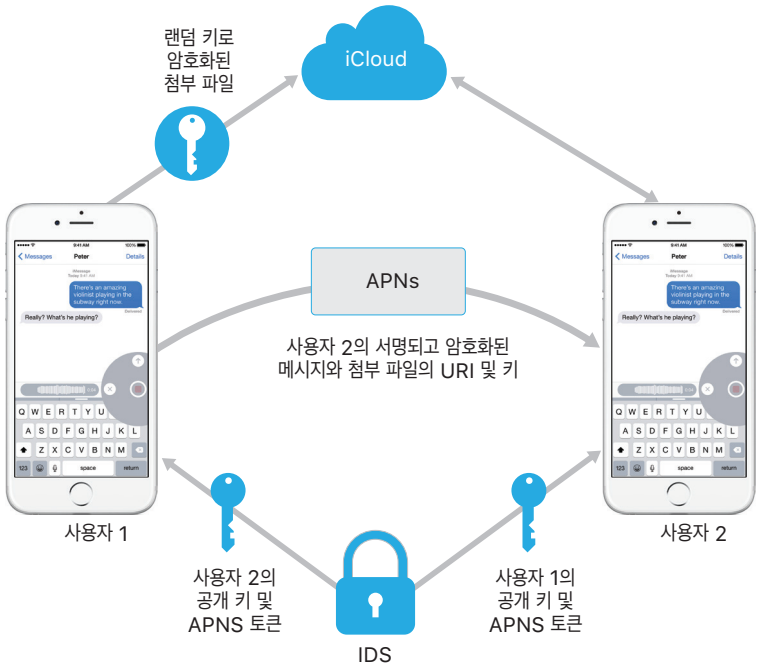
사용자가 iMessage를 사용하도록 추가 기기를 활성화하면 암호화 및 서명 공개 키, APNS 주소 및 관련 전화번호가 디렉토리 서비스에 추가됩니다. 또한 사용자는 더 많은 이메일 주소를 추가할 수 있으며 이메일 주소는 확인 링크를 통해 확인됩니다. 전화번호는 통신 사업자 네트워크와 SIM을 통해 확인됩니다. 일부 네트워크의 경우 SMS 사용이 요구됩니다(SMS 요금이 부과되는 경우에는 확인 대화상자가 사용자에게 표시됩니다). iMessage 외에도 FaceTime 및 iCloud와 같은 몇몇 시스템 서비스의 경우 전화번호 확인이 필요할 수 있습니다. 사용자의 등록된 기기는 모두 새로운 기기, 전화번호 또는 이메일 주소가 추가될 때 알림 메시지를 표시합니다.

iMessage가 메시지를 주고 받는 방법

사용자는 이메일 주소 또는 이름을 입력하여 새로운 iMessage 대화를 시작할 수 있습니다. 전화번호 또는 이메일 주소를 입력하면 기기가 IDS에 연결하여 해당 번호 또는 주소와 관련한 모든 기기에 대한 공개 키와 APNS 주소를 가져옵니다. 사용자가 이름을 입력하면 기기는 우선 사용자의 연락처 앱을 이용하여 해당 이름과 관련한 전화번호와 이메일 주소를 수집한 다음 IDS에서 공개 키와 APNS 주소를 가져옵니다.

사용자가 보내는 메시지는 수신자의 각 기기에서 개별적으로 암호화됩니다. 수신하는 기기의 공개 RSA 암호화 키는 IDS에서 가져옵니다. 또한 전송하는 기기는 각각의 수신하는 기기에 대해 무작위로 88비트 값을 생성하고 HMAC-SHA256 키로 사용하여 전송자와 수신자의 공개 키와 평문에서 파생된 40비트 값을 구성합니다. 88비트 값과 40비트 값을 조합하여 128비트 키가 만들어지고 이 키는 CTR 모드에서 AES에 사용되어 메시지를 암호화합니다. 40비트 값은 수신하는 쪽에서 암호화 해제된 평문의 무결성을 확인하기 위해 사용합니다. 메시지별 AES 키는 RSA-OAEP를 사용하여 수신하는 기기의 공개 키로 암호화됩니다. 그리고 암호화된 메시지 텍스트와 암호화된 메시지 키의 조합은 SHA-1으로 해시되며, 해당 해시는 전송하는 기기의 개인 서명 키를 사용하여 ECDSA로 서명됩니다. 결과로 나온 메시지는 수신하는 기기당 하나이며 암호화된 메시지 텍스트, 암호화된 메시지 키와 보낸 사람의 디지털 서명으로 구성됩니다. 그런 다음 전송을 위해 APNS로 발송됩니다. 타임스탬프나 APNS 라우팅 정보와 같은 메타데이터는 암호화되지 않습니다. APNS와의 통신은 전방향 안전 TLS 채널을 사용하여 암호화됩니다.

APNS는 iOS 버전에 따라 최대 4KB 또는 16KB의 크기의 메시지만 릴레이할 수 있습니다. 메시지 텍스트가 너무 길거나 사진과 같은 첨부 파일이 포함되어 있는 경우 첨부 파일은 CTR 모드의 AES와 무작위로 생성된 256비트 키로 암호화되어 iCloud에 업로드됩니다. 첨부 파일의 AES 키, 해당 URI(Uniform Resource Identifier) 및 암호화된 형태의 SHA-1 해시는 아래의 다이어그램에 나타난 대로 일반 iMessage 암호화를 통해 기밀성과 무결성을 보호하면서 수신자에게 iMessage 콘텐츠로 전송됩니다.



그룹 대화에서는 각 수신자와 수신하는 기기가 이 과정을 반복합니다.

수신하는 쪽의 각 기기는 APNS에서 메시지 사본을 받고 필요한 경우 iCloud에서 첨부 파일을 검색합니다. 가능한 경우, 보낸 사람의 전화번호 또는 이메일 주소가 수신자의 연락처와 일치하면 이름이 표시됩니다.

모든 푸시 알림과 마찬가지로 메시지는 전송되는 경우 APNS에서 삭제됩니다. 그러나 다른 APNS 알림과 달리 iMessage 메시지는 오프라인 기기에 전송을 위해 대기합니다. 메시지는 현재 최대 30일 동안 저장됩니다.

FaceTime

FaceTime은 Apple의 영상 및 음성 통화 서비스입니다. iMessage와 마찬가지로 FaceTime 통화 또한 APNS(Apple 푸시 알림 서비스)를 사용하여 사용자의 등록된 기기에 초기 연결을 구축합니다. FaceTime 통화의 음성/영상 콘텐츠는 엔드 투 엔드 암호화 기술로 안전하게 보호되어 보낸 사람과 받는 사람을 제외한 누구도 해당 콘텐츠에 접근할 수 없습니다. Apple은 해당 데이터의 암호화를 해제할 수 없습니다.

초기 FaceTime 연결은 사용자의 등록 기기 간에 데이터 패킷을 릴레이하는 Apple 서버 인프라를 통해 구축되었습니다. 릴레이되는 연결을 통한 APNS(Apple 푸시 알림 서비스) 알림 및 STUN(Session Traversal Utilities for NAT) 메시지를 사용하여 기기는 신원 인증서를 확인하고 각 세션에 대한 공유 비밀을 구축합니다. 공유 비밀은 SRTP(Secure Real Time Protocol)를 통해 스트리밍되는 미디어 채널의 세션 키를 가져오는 데 사용됩니다. SRTP 패킷은 Counter Mode의 AES-256 및 HMAC-SHA1을 사용하여 암호화됩니다. 초기 연결 및 보안 설정 후 FaceTime은 STUN 및 ICE(Internet Connectivity Establishment)를 사용하여 기기간에 피어 투 피어 연결을 구축합니다(가능한 경우).

iCloud

iCloud는 사용자의 연락처, 캘린더, 사진, 도큐멘트 등을 저장하고 사용자의 모든 기기에서 자동으로 해당 정보를 최신으로 유지합니다. 또한 타사 업체가 iCloud를 사용하여 도큐멘트 뿐만 아니라 앱 데이터의 키 값을 개발자가 정의한 대로 저장하고 동기화할 수도 있습니다. 사용자는 Apple ID로 로그인하고 사용할 서비스를 선택하여 iCloud를 설정합니다. IT 관리자는 MDM 구성 프로파일을 통해 나의 사진 스트림, iCloud Drive 및 iCloud 백업을 포함하여 iCloud 기능을 비활성화할 수 있습니다. 해당 서비스는 저장 중인 내용에 대해 알지 못하며 모든 파일 콘텐츠를 바이트 모음처럼 동일한 방법으로 처리합니다.

각 파일은 청크로 나뉘어지고 SHA-256을 이용하는 각 청크의 콘텐츠에서 파생된 키와 AES-128을 사용하는 iCloud에서 암호화됩니다. 파일의 메타데이터와 키는 Apple이 사용자의 iCloud 계정에 저장합니다. 파일의 암호화된 청크는 사용자의 신원을 확인할 수 있는 정보 없이 S3 및 Google 클라우드 플랫폼 같은 타사 저장 공간 서비스를 사용하여 저장됩니다.

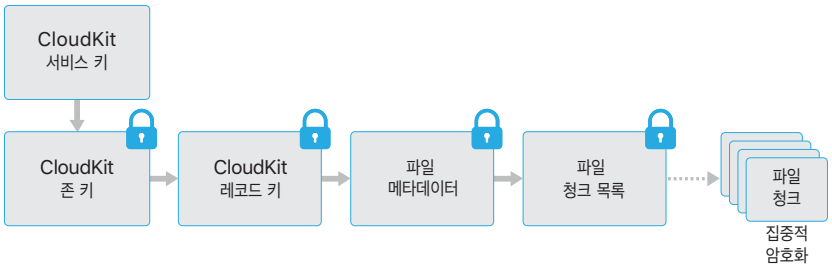
iCloud Drive

iCloud Drive는 계정 기반의 키를 추가하여 iCloud에 저장된 도큐멘트를 보호합니다. 기존 iCloud 서비스와 마찬가지로 파일 콘텐츠를 청크로 나누어 암호화하고 타사 서비스를 사용하여 암호화된 청크를 저장합니다. 하지만 파일 콘텐츠 키는 iCloud Drive 메타데이터와 함께 저장된 레코드 키로 래핑됩니다. 그리고 레코드 키는 사용자의 iCloud Drive 서비스 키로 보호됩니다. 그런 다음 iCloud Drive 서비스 키는 사용자의 iCloud 계정과 함께 저장됩니다. 사용자는 iCloud에서 인증하여 iCloud 도큐멘트 메타데이터에 접근할 수 있지만 iCloud Drive 저장 공간의 보호된 부분을 공개하기 위해서는 iCloud Drive 서비스 키 또한 가지고 있어야 합니다.

CloudKit

CloudKit를 사용하면 앱 개발자가 키 값 데이터, 구조적 데이터 및 iCloud의 자료를 저장할 수 있습니다. CloudKit에 대한 접근은 앱 권한을 사용하여 제어할 수 있습니다. CloudKit는 공개 및 개인 데이터베이스를 모두 지원합니다. 공개 데이터베이스는 앱의 모든 사본에서 주로 일반 자료용으로 사용되며 암호화되지 않습니다. 개인 데이터베이스는 사용자의 데이터를 저장합니다.

iCloud Drive와 마찬가지로 CloudKit는 계정 기반의 키를 사용하여 사용자의 개인 데이터베이스에 저장된 정보를 보호합니다. 또한, 다른 iCloud 서비스와 마찬가지로 타사 서비스를 사용해 파일을 청크로 나누고 암호화하여 저장합니다. CloudKit는 데이터 보호와 마찬가지로 키의 계층 구조를 이용합니다. 먼저 파일별 키는 CloudKit 레코드 키로 래핑됩니다. 그리고 레코드 키는 존 와이드 키 (zone-wide key)로 보호됩니다. 또한, 존 와이드 키는 사용자의 CloudKit Service 키로 보호됩니다. 마지막으로 CloudKit Service 키는 사용자의 iCloud 계정에 저장되며 사용자가 iCloud로 인증한 후에만 사용 가능합니다.



CloudKit 엔드 투 엔드 암호화

Apple Pay Cash, 사용자 키워드, Siri 인공지능 및 Siri야 기능은 iCloud 키체인 동기화에 의해 보호되는 CloudKit 서비스 키를 지닌 CloudKit 엔드 투 엔드 암호화를 사용합니다. 이러한 CloudKit 컨테이너의 경우 키 계층은 iCloud 키체인에서부터 시작됨에 따라 iCloud 키체인의 보안 특징을 공유합니다. 사용자가 신뢰하는 기기에서만 키를 사용할 수 있으며 Apple이나 제 3자는 접근할 수 없습니다. iCloud 키체인 데이터에 대한 접근 권한이 손실된 경우(이 도큐멘트의 '에스크로 보안' 섹션 참조), CloudKit에 있는 데이터는 재설정되며 신뢰하는 로컬 기기에서 데이터를 이용할 수 있는 경우 CloudKit에 데이터가 다시 업로드됩니다.

iCloud 백업

iCloud는 Wi-Fi를 통해 기기 설정값, 앱 데이터, 카메라 롤의 사진 및 비디오, 그리고 메시지 앱의 대화 내용을 포함한 정보를 매일 백업합니다. iCloud는 인터넷을 통해 콘텐츠를 보낼 때 암호화하고, 콘텐츠를 암호화된 포맷으로 저장하고, 인증용 보안 토큰을 사용하여 보호합니다. iCloud 백업은 기기가 잠겨 있고 전원에 연결되어 있으며 Wi-Fi로 인터넷에 연결되었을 때에만 작동합니다. iOS에서 사용하는 암호화 덕분에 시스템이 증분 자동 백업 및 복원을 실행하는 동안 데이터가 안전하게 유지될 수 있습니다.

iCloud는 다음과 같은 정보를 백업합니다.

- 구매한 음악, 동영상, TV 프로그램 앱 및 책의 기록. 사용자의 iCloud 백업은 사용자의 iOS 기기에 있는 구입한 콘텐츠에 대한 정보를 포함합니다. 구입한 콘텐츠 자체는 포함하지 않습니다. 사용자가 iCloud 백업으로 복원하면 iTunes Store, iBooks Store 또는 App Store에서 구입한 콘텐츠가 자동으로 다운로드됩니다. 일부 유형의 콘텐츠는 일부 국가 또는 지역에서만 다운로드할 수 있으며 환불되었거나 스토어에서 더 이상 사용할 수 없게 된 경우, 이전에 구입한 항목은 사용할 수 없게 됩니다. 전체 구입 내역은 사용자의 Apple ID에 연결되어 있습니다.
- 사용자의 iOS 기기에 저장된 사진 및 비디오. 사용자가 iOS 8.1 이상이 설치된 iOS 기기나 OS X 10.10.3 이상이 설치된 Mac에서 iCloud 사진 보관함을 활성화하면 iCloud에 이미 저장된 사진과 비디오는 사용자의 iCloud 백업에 포함되지 않습니다.
- 연락처, 캘린더 이벤트, 미리 알림 및 메모
- 기기 설정값
- 앱 데이터
- 통화 기록 및 벨소리
- 홈 화면 및 앱 구성
- HomeKit 구성
- HealthKit 데이터
- iMessage, 문자 메시지(SMS) 및 MMS 메시지(백업 중에 사용한 SIM 카드가 필요함)
- Visual Voicemail 암호(백업 중에 사용한 SIM 카드가 필요함)

기기가 잠겨있을 때 접근할 수 없는 데이터 보호 클래스에서 파일이 생성되면 파일별 키는 iCloud Backup keybag의 클래스 키를 사용하여 암호화됩니다. 파일이 원래 암호화된 상태로 iCloud에 백업됩니다. 데이터 보호 클래스인 No Protection에 있는 파일은 전송 중에 암호화됩니다.

iCloud Backup keybag은 각 데이터 보호 클래스에 대해 비대칭(Curve25519) 키를 포함합니다. 이 비대칭 키는 파일별 키를 암호화하는 데 사용됩니다. 백업 keybag 및 iCloud Backup keybag의 콘텐츠에 대한 자세한 정보를 보려면 이 도큐멘트의 암호화 및 데이터 보호 섹션에서 '키체인 데이터 보호'를 참조하십시오.

백업 세트는 사용자의 iCloud 계정에 저장되며 사용자 파일의 사본과 iCloud Backup keybag으로 구성됩니다. iCloud Backup keybag은 무작위 키로 보호됩니다. 이 무작위 키 또한 백업 세트와 함께 저장됩니다.(사용자의 iCloud 암호는 암호화에 이용되지 않으므로 iCloud 암호를 변경해도 기존 백업이 무효화되지 않습니다.)

사용자의 키체인 데이터베이스가 iCloud에 백업되는 동안 UID 관련 키로 보호됩니다. 이를 통해 키체인을 생성한 기기로만 키체인을 복구할 수 있으며, 이는 Apple을 포함한 어느 누구도 사용자의 키체인 항목을 읽을 수 없음을 의미합니다.

복원 시에 백업된 파일, iCloud Backup keybag 및 keybag용 키를 사용자의 iCloud 계정에서 가져옵니다. 그리고 iCloud Backup keybag은 keybag용 키를 사용해 암호화를 해제합니다. 그런 다음 keybag의 파일별 키를 사용하여 백업 세트에 있는 파일의 암호화를 해제합니다. 백업 세트의 파일은 파일 시스템에서 새로운 파일로 작성되어 데이터 보호 클래스별로 다시 암호화됩니다.

Safari에 통합된 iCloud 키체인
Safari는 자동으로 암호화된 강력한 무작위 문자열을 생성해 웹 사이트 암호로 사용할 수 있습니다. 이 암호는 키체인에 저장되고 다른 기기와 동기화됩니다. 키체인 항목은 Apple 서버를 통해 기기에서 기기로 전송되지만 Apple과 다른 기기가 해당 콘텐츠를 읽을 수 없는 방법으로 암호화됩니다.

iCloud 키체인

iCloud 키체인을 사용하면 사용자가 Apple에 정보를 노출하지 않고 iOS 기기와 Mac 컴퓨터 간에 암호를 안전하게 동기화할 수 있습니다. 강력한 개인 정보 보호와 보안을 포함하여 사용 편의성과 키체인 복구 능력이라는 목표는 iCloud 키체인의 디자인과 아키텍처에 큰 영향을 주었습니다. iCloud 키체인은 키체인 동기화와 키체인 복구라는 두 가지 서비스로 구성됩니다.

Apple은 사용자의 암호가 다음 조건에서도 계속 보호되도록 iCloud 키체인과 키체인 복구를 설계했습니다.

- 사용자의 iCloud 계정이 손상된 경우
- iCloud가 외부 또는 내부의 공격에 의해 손상된 경우
- 제 3자가 사용자 계정에 접근하는 경우

키체인 동기화

사용자가 iCloud 키체인을 처음으로 활성화하면 기기는 신뢰 서클을 구축하고 자체 동기화 ID를 생성합니다. 동기화 ID는 개인 키와 공개 키로 구성됩니다. 동기화 ID의 공개 키가 서클에 들어가면 서클은 두 번 서명됩니다. 먼저 동기화 ID의 개인 키로 서명된 다음 사용자의 iCloud 계정 암호에서 파생된 비대칭 타원형 키(P-256 사용)로 다시 서명됩니다. 또한 서클에 매개 변수(무작위 솔트 및 반복)가 저장되어 사용자의 iCloud 암호를 기반으로 하는 키를 생성하는 데 사용됩니다.

서명된 동기화 서클은 사용자의 iCloud 키 값 저장 공간 영역에 위치합니다. 사용자의 iCloud 암호를 알지 못하면 읽을 수 없고 해당 구성원의 동기화 ID의 개인 키가 없으면 유효하게 수정할 수 없습니다.

사용자가 다른 기기에서 iCloud 키체인을 켜면 iCloud에서 사용자가 이전에 구축한 동기화 서클의 구성원이 아님을 알립니다. 그리고 기기는 동기화 ID 키 쌍을 생성한 다음 서클의 멤버십을 요청하기 위한 신청 티켓을 생성합니다. 티켓은 기기의 동기화 ID의 공개 키로 구성되며 사용자에게 iCloud 암호로 인증할 것을 요청합니다. 그런 다음 iCloud에서 타원형 키 생성 매개 변수를 가져와 신청 티켓을 서명하는 데 사용되는 키를 생성합니다. 마지막으로 신청 티켓은 iCloud에 저장됩니다.

첫 번째로 서클에 등록된 기기가 신청 티켓의 도착 여부를 확인하면 새로운 기기의 서클 연결 요청을 확인할 수 있도록 사용자에게 알림을 표시합니다. 사용자가 iCloud 암호를 입력하면 신청 티켓을 일치하는 개인 키로 서명했음을 확인합니다. 이렇게 하여 서클 연결 요청을 생성한 사람이 요청한 시점에 사용자의 iCloud 암호를 입력하도록 확립합니다.

사용자가 새로운 기기를 승인하게 되면 첫 번째 기기는 동기화 서클에 새로운 구성원의 공개 키를 추가하고 사용자의 iCloud 암호에서 파생된 키 및 동기화 ID로 다시 서명합니다. 새로운 동기화 서클은 iCloud에 저장되며 또한 서클의 새로운 구성원이 비슷한 방식으로 서명을 합니다.

이제 서명 서클에 두 명의 구성원이 존재하게 되며 각 구성원은 상대의 공개 키를 가집니다. 그리고 iCloud 키 값 저장 공간을 통해 개별 키체인 항목을 교환하거나 필요에 따라 CloudKit에 저장할 수 있습니다. 서클 구성원이 동일한 항목을 가지고 있는 경우 수정일이 가장 최근인 항목이 동기화됩니다. 다른 구성원이 동일한 항목을 가지고 있고 수정일이 동일한 경우 항목을 건너뜁니다. 동기화된 각 항목은 암호화되며 사용자가 신뢰하는 사용자의 서클 내 기기만 암호화를 해제할 수 있습니다. 그렇기 때문에 Apple이 암호화를 해제할 수 없으며 다른 어떤 기기에서도 암호화를 해제할 수 없습니다.

이 과정은 새로운 기기가 동기화 서클에 연결될 때마다 반복됩니다. 예를 들어, 세 번째 기기가 연결될 때 사용자의 다른 기기 두 대에 확인 메시지가 나타납니다. 사용자는 해당 기기 중 하나에서 새로운 구성원을 승인할 수 있습니다. 새로운 상대가 추가되면 각 상대는 새로운 상대와 동기화하여 모든 구성원이 동일한 키체인 항목을 가지고 있는지 확인합니다.

하지만 전체 키체인은 동기화되지 않습니다. VPN ID와 같은 일부 항목은 특정 기기에만 해당되며 기기에서 벗어날 수 없습니다. 또한 `kSecAttrSynchronizable` 속성이 있는 항목만 동기화됩니다. Apple은 Wi-Fi 암호와 HomeKit 암호화 키뿐 아니라 Safari 사용자 데이터(사용자 이름, 암호 및 신용 카드 번호 포함)에 대해 이 속성을 설정했습니다.

또한, 기본적으로 타사 앱에서 추가된 키체인 항목은 동기화하지 않습니다. 개발자는 키체인에 항목을 추가할 때 `kSecAttrSynchronizable`을 설정해야 합니다.

키체인 복구

키체인 복구는 사용자에게 Apple에 키체인을 선택적으로 위탁하는 방법을 제공합니다. 하지만 Apple이 암호 및 다른 데이터를 읽는 것을 허용하지 않습니다. 기기를 하나만 가지고 있더라도 키체인 복구는 데이터 유실에 대한 안전망을 제공합니다. Safari를 사용해 웹 계정용으로 무작위로 강력한 암호를 생성하는 경우 해당 암호는 키체인에만 기록되기 때문에 유실 방지가 특히 중요합니다.

키체인 복구의 기본은 보조 인증과 보안 에스크로 서비스로서 Apple이 키체인 복구를 위해 특별히 개발한 기술입니다. 사용자의 키체인은 강력한 암호를 사용하여 암호화되고 에스크로 서비스는 일련의 엄격한 조건이 충족될 때에만 키체인의 사본을 제공합니다.

iCloud 키체인이 켜져 있을 때 이중 인증이 사용자의 계정에 활성화되어 있는 경우 기기 암호는 에스크로된 키체인을 복구하는 데 사용됩니다. 이중 인증이 설정되어 있지 않으면 사용자는 6자리 암호로 iCloud 보안 코드를 생성하도록 요청 받습니다. 사용자는 이중 인증 대신 자신만의 긴 코드를 지정하거나 기기에서 암호화된 무작위 코드를 생성하여 자체적으로 기록하고 유지하도록 할 수도 있습니다.

그런 다음 iOS 기기는 사용자의 키체인 사본을 내보내서 비대칭 keybag으로 키를 래핑하도록 암호화하여 사용자의 iCloud 키 값 저장 공간 영역에 배치합니다. keybag은 사용자의 iCloud 보안 코드와 에스크로 레코드를 저장하는 HSM(하드웨어 보안 모듈) 클러스터의 공개 키로 래핑됩니다. 이는 사용자의 iCloud 에스크로 레코드가 됩니다.

사용자가 자체의 보안 코드를 지정하거나 4자리 값을 사용하는 대신 암호화된 무작위 보안 코드를 사용하려는 경우 에스크로 레코드가 필요하지 않습니다. 대신 iCloud 보안 코드를 사용하여 무작위 키를 직접 래핑합니다.

보안 코드를 설정한 뒤에 사용자는 전화번호를 등록해야 합니다. 전화번호는 키체인 복구에서 인증의 보조 단계로 활용됩니다. 복구 시에 사용자는 SMS를 받게 되며 답장을 하지 않으면 복구를 진행할 수 없습니다.

에스크로 보안

iCloud는 인증된 사용자와 기기만 복구를 수행할 수 있도록 키체인 에스크로에 대한 보안 인프라를 제공합니다. 순서적으로 iCloud 뒤에 위치한 HSM(하드웨어 보안 모듈) 클러스터는 에스크로 레코드를 보호합니다. 이 도큐멘트에서 앞서 설명한 대로 각각의 클러스터는 관리 하에 에스크로 레코드를 암호화하는 데 사용되는 키를 가지고 있습니다.

키체인을 복구하려면 사용자는 iCloud 계정과 암호로 인증하고 등록된 전화번호로 전송된 SMS에 응답해야 합니다. 완료되면 사용자는 iCloud 보안 코드를 입력해야 합니다. HSM 클러스터는 SRP(Secure Remote Password) 프로토콜을 사용하여 사용자가 iCloud 보안 코드를 아는지 확인합니다. 코드 자체는 Apple에 전송되지 않습니다. 아래 설명처럼 클러스터의 구성원은 사용자가 레코드 가져오기에 허용된 최대 시도 횟수를 초과하지 않았는지 개별적으로 확인합니다. 구성원 과반수가 동의하는 경우 클러스터는 에스크로 레코드의 래핑을 해제하여 사용자의 기기로 전송합니다.

그런 다음, 기기는 iCloud 보안 코드를 사용하여 사용자의 키체인을 암호화하는 데 사용한 무작위 키의 래핑을 해제합니다. 해당 키를 사용하여 iCloud 키 값 저장 공간에서 가져온 키체인은 암호화가 해제되고 기기에서 복구됩니다. 에스크로 레코드는 인증하고 가져오는 데 10번의 시도만 허용됩니다. 여러 번 실패한 후에는 레코드가 잠기므로 추가 시도를 승인받으려면 사용자는 Apple 지원에 문의해야 합니다. 10번 실패하는 경우 HSM 클러스터가 에스크로 레코드를 파기하여 키체인이 영원히 유실됩니다. 이를 통해 키체인 데이터를 희생하지만 무작위 대입 공격으로 레코드를 가져오려는 시도를 방지합니다.

이러한 정책은 HSM 펌웨어에 구현되어 있습니다. 펌웨어의 변경을 허용하는 관리 접근 카드는 파기되었습니다. 펌웨어를 변경하거나 개인 키에 접근하기 위한 시도가 있는 경우 HSM 클러스터는 개인 키를 삭제합니다. 이렇게 되면 클러스터로 보호된 각 키체인의 소유자는 해당 에스크로 레코드가 유실되었음을 알리는 메시지를 받게 됩니다. 그리고 나서 다시 등록하도록 선택할 수 있습니다.

Siri

자연스럽게 말하는 것만으로 사용자는 Siri에게 메시지를 보내고, 회의 일정을 잡으며, 전화를 거는 등의 작업을 요청할 수 있습니다. Siri는 음성 인식, 텍스트 말하기 및 클라이언트 대 서버 모델을 사용하여 광범위한 요청에 응답합니다. Siri가 지원하는 작업은 최소한의 개인 정보만 이용하고 완벽하게 보호되도록 디자인되었습니다.

Siri를 켜면 기기가 음성 인식 및 Siri 서버를 사용하도록 무작위 ID를 생성합니다. 이러한 ID는 Siri 내에서만 사용되며 서비스 향상에 이용됩니다. 이후에 Siri를 끄면 기기는 Siri를 다시 켜는 경우에 사용하도록 새로운 무작위 ID를 생성합니다.

Siri의 기능을 사용하기 위해 기기의 사용자 정보 중 일부가 서버로 전송됩니다. 사용자 정보에는 음악 보관함에 대한 정보(노래 제목, 아티스트 및 재생목록), 미리 알림 목록의 이름, 연락처에 정의된 이름과 관계를 포함합니다. 서버와의 모든 통신은 HTTPS를 통해 이루어집니다.

Siri 세션이 시작되면 연락처에서 가져온 사용자의 성과 이름, 그리고 대략적인 지리적 위치가 서버로 전송됩니다. 이를 통해 Siri가 사용자의 이름을 사용해 답변하거나 날씨에 대한 질문처럼 대략적인 위치만 필요한 질문에 답변할 수 있습니다.

더욱 정확한 위치가 필요한 경우(예: 주변 영화관의 위치 확인) 서버는 기기에게 더욱 정확한 위치를 제공하도록 요청합니다. 사용자의 요청을 처리하기 위해 엄격히 필요한 경우에만 기본적으로 정보가 서버로 전송됩니다. 어떤 상황에서든 10분 동안 사용하지 않으면 세션 정보가 삭제됩니다.

Apple Watch에서 Siri를 사용하면 시계는 위에서 설명한대로 자체의 고유한 무작위 ID를 생성합니다. 하지만 사용자의 정보를 다시 전송하는 대신 쌍으로 연결된 iPhone의 Siri ID를 요청 시에 함께 보내 해당 정보에 대한 참조 자료를 제공합니다.

사용자가 말한 음성은 Apple의 음성 인식 서버로 전송됩니다. 받아쓰기만 포함되는 작업의 경우 인식된 텍스트가 해당 기기로 다시 전송됩니다. 다른 경우에 Siri는 텍스트를 분석하고 필요한 경우 기기와 연관된 프로파일의 정보와 결합합니다. 예를 들어, '엄마에게 메시지를 보내줘'라고 요청한 경우 연락처 앱에서 서버로 업로드한 관계와 이름이 활용됩니다. 그리고 식별된 동작에 대한 명령이 기기로 다시 전송되어 실행됩니다.

서버에서 전송된 명령을 통해 기기에서 여러 Siri 기능을 수행합니다. 예를 들어, 사용자가 Siri에게 수신 메시지를 읽도록 요청하는 경우, 서버가 기기에게 읽지 않은 메시지의 콘텐츠를 말하도록 명령합니다. 하지만 메시지의 콘텐츠와 보낸 사람은 서버로 전송되지 않습니다.

사용자 음성 녹음은 인식 시스템이 사용자의 음성을 잘 이해하기 위해 활용할 수 있도록 6개월간 저장됩니다. 6개월 후에는 식별자가 없는 다른 사본을 저장합니다. 저장된 정보는 Siri를 향상시키고 개발하기 위해 Apple이 최대 2년 동안 사용합니다. 식별자가 없는 일부 녹음, 대화 기록 및 관련 데이터는 Siri의 성능 향상 및 품질 개선을 위해 2년 이상 Apple에서 사용할 수 있습니다. 추가적으로 음악, 스포츠 팀과 선수, 비즈니스 또는 관심 지역 정보를 언급하는 일부 음성 녹음은 비슷하게 Siri를 향상시킬 목적으로 저장됩니다.

Siri는 또한 음성 구동 방식을 통해 핸즈프리로 작동시킬 수 있습니다. 음성 작동 감지 기능은 해당 기기의 기기 내에서 수행됩니다. 이 모드에서 Siri는 수신 오디오 패턴이 지정된 작동 구문의 음향과 충분히 일치할 때에만 활성화됩니다. 작동 구문이 감지되면 Siri 명령을 포함한 작동 구문 오디오가 추가 처리를 위해 Apple의 음성 인식 서버로 전송됩니다. Apple의 음성 인식 서버는 Siri를 통해 만들어진 사용자 음성 녹음과 동일한 규칙을 따릅니다.

연속성

연속성은 iCloud, Bluetooth 및 Wi-Fi 같은 기술을 활용하여 사용자가 한 기기에서 다른 기기로 하던 작업을 계속하고, 전화를 걸거나 받고, 문자 메시지를 주고 받으며, 셀룰러 인터넷 연결을 공유할 수 있도록 해줍니다.

Handoff

Handoff를 사용하면 사용자의 Mac과 iOS 기기가 서로 가까이에 있을 때 자동으로 한 기기에서 수행 중인 작업을 다른 기기로 전달할 수 있습니다. Handoff를 사용하면 사용자가 기기를 전환하고 즉시 작업을 이어갈 수 있습니다.

사용자가 Handoff를 지원하는 두 번째 기기에서 iCloud에 로그인하면 두 기기가 APNS를 사용하여 Bluetooth LE 4.0 대역 외(OOB) 페어링을 구축합니다. 개별 메시지는 iMessage와 비슷한 방법으로 암호화됩니다. 기기가 쌍으로 연결되면 각각은 기기의 키체인에 저장되는 대칭 256비트 AES 키를 생성합니다. 이 키는 Bluetooth LE 광고를 암호화하고 인증할 수 있습니다. 광고는 재전송 방지책과 함께 GCM 모드의 AES-256을 사용하여 기기의 현재 동작을 iCloud와 쌍으로 연결된 다른 기기에 전달합니다. 기기가 새로운 키에서 광고를 처음 받는 경우, 발신하는 기기에 Bluetooth LE 연결을 구축하고 광고 암호화 키 교환을 수행합니다. 이 연결은 iMessage 암호화 방법과 비슷한 방법으로 개별 메시지 암호화 방식과 표준 Bluetooth LE 4.0 암호화 방식을 사용하여 보호됩니다. 일부 상황에서 이러한 메시지는 Bluetooth LE 대신 APNS를 통해 전송됩니다. 동작 페이로드는 iMessage와 동일한 방법으로 보호되고 전송됩니다.

네이티브 앱과 웹 사이트 간의 Handoff

Handoff를 사용하면 iOS 네이티브 앱에서 앱 개발자가 합법적으로 제어하는 도메인의 웹 페이지를 재개할 수 있습니다. 또한 Handoff를 통해 네이티브 앱 사용자 동작을 웹 브라우저에서 재개할 수도 있습니다.

네이티브 앱이 개발자가 제어하지 않는 웹 사이트를 재개하지 못하도록 하려면 앱이 재개하려는 웹 도메인에 대한 합법적인 제어를 입증해야 합니다. 웹 사이트 도메인을 통한 제어는 공유 웹 인증서를 위한 메커니즘을 통해 구축됩니다. 자세한 내용은 이 도큐멘트의 암호화 및 데이터 보호 섹션에서 ‘Safari에 저장된 암호 접근’을 참조하십시오. 시스템은 사용자 동작 Handoff를 승인하도록 앱을 허용하기 전에 앱의 도메인 이름 제어가 유효한지 확인해야 합니다.

웹 페이지 Handoff의 소스는 Handoff API를 채택한 어떤 브라우저든 가능합니다. 사용자가 웹 페이지를 볼 때 시스템은 암호화된 Handoff 광고 바이트에서 웹 페이지의 도메인 이름을 광고합니다. 이 섹션에서 설명한 것처럼 사용자의 다른 기기만 광고 바이트의 암호화를 해제할 수 있습니다.

수신하는 기기에서 시스템은 설치된 네이티브 앱이 광고된 도메인 이름으로부터 Handoff를 승인함을 감지하고 해당 네이티브 앱 아이콘을 Handoff 옵션으로 표시합니다. Handoff 실행 시, 네이티브 앱은 전체 URL과 웹 페이지 제목을 받습니다. 하지만 브라우저에서 네이티브 앱으로 다른 정보는 전달되지 않습니다.

반대로 네이티브 앱은 Handoff를 수신하는 기기에 동일한 네이티브 앱이 설치되어 있지 않으면 폴백 URL을 지정할 수도 있습니다. 이 경우, 기본 브라우저가 Handoff API를 사용한다면 시스템은 사용자의 기본 브라우저를 Handoff 앱 옵션으로 표시합니다. Handoff가 요청되면 브라우저가 실행되고 소스 앱이 제공한 폴백 URL을 받게 됩니다. 폴백 URL을 네이티브 앱 개발자가 제어하는 도메인 이름으로 제한해야 한다는 요구사항은 없습니다.

대용량 데이터 Handoff

Handoff의 기본 기능에 추가로 일부 앱은 Apple이 만든 피어 투 피어 Wi-Fi 기술을 통해 (AirDrop과 비슷한 방법으로) 대용량의 데이터 전송을 지원하는 API도 사용할 수 있습니다. 예를 들어, Mail 앱은 이러한 API를 사용하여 대용량 첨부 파일을 포함한 임시 저장 메일의 Handoff를 지원합니다.

앱이 이 기능을 사용하면 두 기기 간의 교환이 Handoff에서와 마찬가지로 시작됩니다(이전 섹션 참조). 하지만 수신하는 기기는 Bluetooth LE를 사용하여 초기 페이로드를 받은 후에 Wi-Fi를 통해 새로운 연결을 시작합니다. 이 연결은 암호화(TLS)되어 iCloud 신원 인증서를 교환합니다. 그리고 인증서에서 신원을 사용자의 신원과 대조하여 확인합니다. 마지막으로 추가 페이로드 데이터가 전송이 완료될 때까지 이 암호화된 연결을 통해 전송됩니다.

공통 클립보드

공통 클립보드는 Handoff를 활용하여 사용자의 클립보드 콘텐츠를 안전하게 모든 기기에 전송할 수 있어 한 기기에서 복사한 콘텐츠를 다른 기기에서 붙여넣을 수 있는 기능입니다. 콘텐츠는 다른 Handoff 데이터와 동일한 방법으로 보호되며 앱 개발자가 공유를 거부하도록 선택하지 않은 이상 공통 클립보드와 자동으로 공유되도록 기본 설정됩니다.

또한 사용자가 클립보드 데이터를 앱에 붙여넣지 않더라도 앱은 클립보드 데이터에 대한 접근 권한을 가집니다. 공통 클립보드를 사용하면 데이터 접근 권한이 iCloud 로그인을 통해 연결된 사용자의 다른 기기까지 확대됩니다.

자동 잠금 해제

자동 잠금 해제를 지원하는 Mac 컴퓨터는 Bluetooth LE 및 피어 투 피어 Wi-Fi를 사용하여 사용자의 Apple Watch가 Mac을 안전하게 잠금 해제하도록 합니다. iCloud 계정에 연결된 Mac과 Apple Watch는 모두 TFA(이중 인증)를 사용해야 합니다.

Apple Watch가 Mac 잠금 해제를 활성화하면 자동 잠금 해제 ID를 사용하는 안전한 링크가 구축됩니다. Mac은 일회용 무작위 잠금 해제 비밀을 생성하여 이 링크를 통해 Apple Watch로 전송합니다. 비밀은 Apple Watch에 저장되며 Apple Watch가 잠금 해제된 경우에만 접근할 수 있습니다('데이터 보호 클래스' 섹션 참조). 마스터 엔트로피나 새로운 비밀은 사용자의 암호가 아닙니다.

잠금 해제 작업을 수행하는 동안 Mac은 Bluetooth LE를 통해 Apple Watch와 연결됩니다. 그렇게 되면 안전한 링크가 처음 활성화되었을 때 사용된 공유 키를 사용하여 두 기기 간에 안전한 링크가 구축됩니다. 연결된 Mac 및 Apple Watch는 피어 투 피어 Wi-Fi와 안전한 링크에서 파생된 안전한 키를 사용하여 두 기기 간의 거리를 파악합니다. 두 기기가 범위 안에 들어오면 안전한 링크를 사용하여 이미 공유된 비밀을 전송하여 Mac을 잠금 해제합니다. 잠금 해제가 완료되면 Mac은 현재 잠금 해제 비밀을 새로운 일회용 잠금 해제 비밀로 교체한 다음, 링크를 통해 새로운 잠금 해제 비밀을 Apple Watch로 전송합니다.

iPhone 셀룰러 통화 릴레이

Mac, iPad 또는 iPod touch가 사용자의 iPhone과 동일한 Wi-Fi 네트워크에 있으면 사용자의 iPhone 셀룰러 연결을 사용하여 전화를 걸고 받을 수 있습니다. 구성하려면 기기에서 동일한 Apple ID 계정을 사용하여 iCloud와 FaceTime 모두에 로그인해야 합니다.

수신 통화가 도착하면 설정된 기기 모두에서 Apple 푸시 알림 서비스를 통해 알림을 받게 됩니다. 각 알림은 iMessage와 동일한 엔드 투 엔드 암호화를 사용합니다. 동일한 네트워크에 있는 기기는 수신 통화 알림 UI를 나타냅니다. 전화를 받는 즉시 두 기기 간의 안전한 피어 투 피어 연결을 통해 사용자의 iPhone에서 나오는 오디오가 완벽하게 전송됩니다.

한 기기에서 전화를 받는 경우 가까이에 있는 iCloud로 연결된 기기의 벨소리가 Bluetooth LE 4.0 광고를 받아 바로 꺼집니다. 광고 바이트는 Handoff 광고와 같은 방식을 통하여 암호화됩니다.

발신 통화도 또한 Apple 푸시 알림 서비스를 통해 iPhone에 릴레이됩니다. 비슷한 방식으로 오디오는 기기 간에 안전한 피어 투 피어 링크를 통해 전송됩니다.

사용자가 FaceTime 설정에서 iPhone 셀룰러 통화를 끔으로써 기기에서 전화 통화 릴레이를 비활성화할 수 있습니다.

iPhone 문자 메시지 전달

문자 메시지 전달 기능을 통해 iPhone에서 받은 SMS 문자 메시지를 사용자의 등록된 iPad, iPod touch 또는 Mac에 자동으로 전송할 수 있습니다. 각 기기는 동일한 Apple ID 계정을 사용하여 iMessage 서비스에 로그인해야 합니다. 사용자의 신뢰 서클 내 기기에 문자 메시지 전달 기능이 켜져 있고 이중 인증이 활성화되어 있는 경우 자동으로 등록됩니다. 또는 iPhone에서 무작위 6자리 숫자 코드가 생성됩니다. 이 코드를 사용하여 각 기기를 등록할 수 있습니다.

기기가 연결되어 있으면 이 도큐멘트의 iMessage 섹션에서 설명된 방법을 이용하여 iPhone이 수신한 SMS 문자 메시지를 암호화하여 각 기기로 전달합니다. 답장은 동일한 방법을 통해 iPhone으로 전송되며 iPhone은 통신 사업자의 SMS 전송 방식을 사용하여 그 답장을 문자 메시지로 전송합니다. 문자 메시지 전달은 메시지 설정에서 켜거나 끌 수 있습니다.

Instant Hotspot

Instant Hotspot을 지원하는 iOS 기기는 Bluetooth LE를 사용해 동일한 iCloud 계정으로 로그인한 기기를 찾아 통신합니다. OS X Yosemite 이상 버전이 설치된 호환되는 Mac 컴퓨터는 동일한 기술을 통해 Instant Hotspot iOS 기기를 찾아 통신합니다.

사용자가 iOS 기기에서 Wi-Fi 설정 패널에 들어가는 경우 기기는 Bluetooth LE 신호를 내보냅니다. 신호는 동일한 iCloud 계정에 로그인된 모든 기기가 인증한 식별자를 포함합니다. 해당 식별자는 DSID(Destination Signaling Identifier)에서 생성됩니다. DSID는 iCloud 계정에 연결되어 있으며 주기적으로 교체됩니다. 동일한 iCloud 계정에 로그인된 다른 기기가 가까이에 있고 개인용 핫스팟을 지원하면 사용 가능을 나타내는 신호와 응답을 감지합니다.

사용자가 개인용 핫스팟을 사용할 수 있는 기기를 선택하면 해당 기기가 개인용 핫스팟을 켜도록 요청을 보냅니다. 해당 요청은 표준 Bluetooth LE 암호화를 사용하여 암호화된 링크를 통해 전송됩니다. 이는 iMessage 암호화와 비슷한 방법입니다. 그러면 기기는 개인용 핫스팟 연결 정보와 같은 메시지별 암호화를 사용하여 동일한 Bluetooth LE 링크를 통해 반응합니다.

Safari 제안, 검색에서 Siri 제안, 찾기, #이미지, News 앱 및 News를 지원하지 않는 국가에서의 News 위젯

Safari 제안, 검색에서의 Siri 제안, 찾기, #이미지 및 News앱을 지원하지 않는 국가에서의 News 위젯은 위키백과, iTunes Store, 지역 뉴스, 지도 검색 결과 및 App Store와 같은 소스처럼 기기를 벗어난 제안 사항을 보여주며 심지어 사용자가 입력하지 않아도 제안 사항을 제공합니다.

사용자가 Safari 주소 막대에 입력을 시작하거나 검색에서 Siri 제안을 열거나 사용할 때, 찾기를 사용할 때, #이미지를 열 때, News 앱에서 검색을 사용하거나 News를 지원하지 않는 국가에서 News 위젯을 사용하는 경우 다음과 같은 정보가 HTTPS로 암호화되어 Apple에 전송되고 사용자에게 알맞은 결과가 제공됩니다.

- 개인 정보 보호를 위해 15분마다 순환하는 ID
- 사용자 검색 질문
- 로컬에 캐시된 이전 검색 및 구문에 기반한 가장 가능성 높은 질문 완성
- 기기의 대략적인 위치(위치 기반 제안에 대해 위치 서비스가 켜진 경우) 위치를 흐리게 처리하는 정도는 현재 기기 위치의 예상 인구 밀도를 기반으로 합니다. 예를 들어, 사용자가 지리적으로 더 분산된 교외 지역은 더욱 흐리게 처리하는 반면에 사용자가 더 가깝게 밀집한 도시 중심 지역은 덜 흐리게 처리합니다. 또한, 사용자가 설정 앱에서 위치 기반 제안의 위치 서비스를 끄면 모든 위치 정보를 Apple에 전송하는 것을 비활성화할 수 있습니다. 위치 서비스를 끄면 Apple은 기기의 IP 주소를 사용하여 대략적인 위치를 추측할 수도 있습니다.
- 기기 유형 및 검색 위치(검색에서의 Siri 제안, Safari, 찾기, News 앱, 메시지 앱)
- 연결 유형
- 기기에서 가장 최근에 사용한 3개의 앱에 대한 정보(추가 검색 구문으로 사용됨) Apple이 관리하는 인기 앱 허용 목록에 포함되어 있고 지난 3시간 내에 접근한 앱만 포함됩니다.
- 기기의 인기 응용 프로그램 목록
- 언어, 지역 및 입력 설정
- 사용자의 기기에 음악 또는 비디오 구독 서비스에 접근할 수 있는 권한이 있는 경우 구독 서비스 이름과 구독 서비스 유형과 같은 정보가 Apple로 전송됩니다. 사용자 계정 이름, 번호 및 암호는 Apple로 전송되지 않습니다.
- 관심 주제를 요약하고 종합한 표현

사용자가 결과를 선택하거나 결과를 선택하지 않고 앱을 종료하면 차후 검색 제안 품질을 개선할 수 있도록 일부 정보가 Apple로 전송됩니다. 이러한 정보는 동일한 15분 세션 ID에만 연결되며 사용자의 신원 정보와는 연결되지 않습니다. 피드백은 앞서 설명한 내용 일부와 다음과 같은 상호 작용 정보를 포함합니다.

- 대화와 검색 네트워크 요청 사이의 타이밍
- 랭킹과 제안 표시 순서
- 결과 ID와 선택된 동작(결과가 로컬이 아닌 경우) 또는 선택된 로컬 결과의 카테고리(결과가 로컬인 경우)
- 사용자가 결과를 선택했는지를 표시하는 깃발

Apple은 18개월 동안 질문, 구문 및 피드백과 함께 제안 로그를 보유합니다. 일부 로그는 최대 5년까지 보관됩니다. 예를 들어 질문, 지역 설정, 도메인, 대략적 위치, 메트릭스 합계 등과 같은 로그입니다.

일부 경우에 제안은 일반적인 단어와 구문에 대한 질문을 적합한 파트너에게 전달하여 파트너의 검색 결과를 받아서 표시할 수 있습니다. Apple은 질문을 파트너에게 중개하여 파트너가 사용자의 IP 주소 또는 검색 피드백을 받지 않도록 합니다. 파트너와의 통신은 HTTPS를 통하여 암호화됩니다. 자주 묻는 질문의 경우 Apple에서는 파트너사가 검색 성능을 개선하도록 도시 단계 위치 정보, 기기 유형 및 클라이언트 언어 정보를 검색 구문으로서 파트너사에게 제공합니다. iOS 11에서 검색의 Siri 제안은 파트너사에 전송되지 않습니다.

제안의 성능을 지리적으로 개선하고 제안이 다양한 네트워크 유형에서 작동하도록 하기 위해 아래와 같은 정보를 세션 ID 없이 기록합니다.

- 부분적인 IP 주소(IPv4 주소의 마지막 옥텟 제외, IPv6 주소의 마지막 80비트 제외)
- 대략적인 위치
- 질문의 대략적인 시간
- 대기 시간/전송 시간
- 응답 크기
- 연결 유형
- 지역 설정
- 기기 유형 및 요청하는 앱

기기 제어

iOS는 유연한 보안 정책과 쉽게 시행하고 관리할 수 있는 설정을 지원합니다. 이를 통해 회사는 기업 정보를 보호하고 직원들이 개인 기기를 사용하더라도(예: BYOD(개인 기기 사용) 프로그램) 기업의 요구사항을 준수하도록 할 수 있습니다.

회사는 암호 보호, 구성 프로파일, 원격 지우기 및 타사 MDM 솔루션과 같은 방법을 사용하여 수많은 기기를 관리할 수 있고 직원이 개인용 iOS 기기를 사용하는 경우에도 기업 데이터를 안전하게 유지할 수 있습니다.

암호 보호

기본적으로 사용자의 암호는 숫자 PIN으로 설정됩니다. Touch ID 또는 Face ID를 지원하는 기기의 암호 길이는 최소 6자리입니다. 다른 기기에서는 암호 길이가 최소 4자리입니다. 설정 > 암호의 암호 옵션에서 사용자 지정 알파벳 숫자 코드를 선택하면 긴 알파벳 숫자 암호를 설정할 수 있습니다. 추측하거나 해킹하기 어려운 길고 복잡한 암호를 사용할 것을 권장합니다.

관리자는 MDM 또는 Exchange ActiveSync를 사용하거나 사용자에게 수동으로 구성 프로파일을 설치하도록 요청하여 복잡한 암호 요구사항 및 기타 정책을 시행할 수 있습니다. 다음과 같은 암호 정책을 사용할 수 있습니다.

- 단순 값 허용
- 알파벳 숫자 값 필요
- 암호 최소 길이
- 복잡한 문자의 최소 개수
- 최대 암호 사용 가능 기간
- 암호 기록
- 자동 잠금 시간
- 기기 잠금 유예 시간
- 최대 암호 입력 시도 횟수
- Touch ID 또는 Face ID 허용

각 정책에 관한 관리자용 정보를 더 보려면 아래 사이트로 이동하십시오.
help.apple.com/deployment/ios/#/apd4D6A472A-A494-4DFD-B559-D59E63167E43

각 정책에 관한 개발자용 정보를 더 보려면 아래 사이트로 이동하십시오.
developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef

iOS 페어링 모델

iOS는 페어링 모델을 사용하여 호스트 컴퓨터에서 기기로의 접근을 제어합니다. 페어링은 기기와 그에 연결된 호스트 간에 신뢰 관계를 구축합니다. 연결된 호스트는 공개 키 교환을 통해 나타납니다. iOS는 이 신뢰 관계의 서명을 사용하여 연결된 호스트에서 데이터 동기화와 같은 추가 기능을 사용할 수 있습니다.

iOS 9에서 페어링이 필요한 서비스는 사용자가 기기를 잠금 해제한 후에만 시작될 수 있습니다.

추가적으로 iOS 10에서 사진 동기화를 포함한 일부 서비스는 시작할 때 기기를 잠금 해제하도록 요청합니다.

iOS 11부터 마지막 잠금 해제 후 일정 시간이 지나면 서비스가 시작되지 않습니다.

페어링 과정을 진행하려면 사용자가 기기를 잠금 해제하고 호스트에서 보낸 페어링 요청을 승인해야 합니다. 또한 iOS 11부터 사용자는 암호를 입력해야 합니다. 사용자가 이 과정을 완료하면 호스트와 기기는 2048비트 RSA 공개 키를 서로 교환하여 저장합니다. 그러면 호스트는 기기에 저장된 Escrow keybag을 잠금 해제할 수 있는 256비트 키를 받게 됩니다(이 도큐멘트의 'Keybag' 섹션에서 Escrow keybag을 참조하십시오). 교환된 키는 암호화된 SSL 세션을 시작하는 데 사용됩니다. 기기가 보호된 데이터를 호스트로 전송하거나 서비스(iTunes 동기화, 파일 전송, Xcode 개발 등)를 시작하기 전에 이 세션을 시작해야 합니다. 기기는 암호화된 세션을 모든 통신에 사용하기 위해 Wi-Fi를 통해 호스트에서 연결을 요구합니다. 그렇기 때문에 기기는 이전에 USB로 연결되어야 합니다. 또한 페어링을 사용하면 여러 진단 기능을 사용할 수도 있습니다. iOS 9에서 페어링 기록이 6개월 이상 사용되지 않았다면 그 기록은 만료됩니다. iOS 11에서는 기간이 30일로 줄었습니다.

자세한 정보를 보려면 아래 사이트로 이동하십시오.
support.apple.com/ko-kr/HT203034

com.apple.pcapd와 같은 특정 서비스는 USB를 통해서만 작동할 수 있습니다. 추가적으로 com.apple.file_relay 서비스를 설치하려면 Apple이 서명한 구성 프로파일이 필요합니다.

iOS 11에서 Apple TV는 SRP(Secure Remote Password) 프로토콜을 사용하여 무선으로 페어링을 구축할 수 있습니다.

사용자는 '네트워크 설정 재설정' 또는 '위치 및 개인 정보 보호 재설정' 옵션을 사용하여 신뢰할 수 있는 호스트 목록을 지울 수 있습니다.

자세한 정보를 보려면 아래 사이트로 이동하십시오.
support.apple.com/ko-kr/HT202778

구성 적용

구성 프로파일은 관리자가 iOS 기기로 구성 정보를 배포할 수 있도록 해주는 XML 파일입니다. 사용자는 설치된 구성 프로파일의 설정을 변경할 수 없습니다. 만약 사용자가 구성 프로파일을 삭제하면 프로파일에서 만든 설정도 모두 제거됩니다. 이러한 방식으로 관리자는 정책을 Wi-Fi 및 데이터 접근에 접목하여 설정을 적용할 수 있습니다. 예를 들어, 이메일 구성을 제공하는 구성 프로파일은 기기 암호 정책을 설정할 수도 있습니다. 그런 경우, 암호가 관리자의 요구사항을 충족하지 못하면 사용자는 이메일에 접근할 수 없게 됩니다.

iOS 구성 프로파일은 다음을 포함하여 지정 가능한 다수의 설정을 포함합니다.

- 암호 정책
- 기기 기능에 대한 제한(예: 카메라 비활성화)
- Wi-Fi 설정
- VPN 설정
- Mail 서버 설정
- Exchange 설정
- LDAP 디렉토리 서비스 설정
- CalDAV 캘린더 서비스 설정
- 웹 클립
- 인증서 및 키
- 고급 셀룰러 네트워크 설정

현재 목록을 보려면(관리자용) 아래 사이트로 이동하십시오.
help.apple.com/deployment/ios/#/cad5370d089

현재 목록을 보려면(개발자용) 아래 사이트로 이동하십시오.
developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef

구성 프로파일은 그 출처를 인증하고 무결성을 확인하며 콘텐츠를 보호하기 위해 서명하고 암호화할 수 있습니다. 구성 프로파일은 3DES와 AES-128을 지원하는 CMS(RFC 3852)를 사용하여 암호화됩니다.

또한 구성 프로파일은 기기에 잠가 제거하지 못하도록 완전히 막을 수 있거나 암호를 사용해야만 제거할 수 있습니다. 많은 기업 사용자가 개인 iOS 기기를 소유하고 있기 때문에 기기를 MDM 솔루션에 바인딩하는 구성 프로파일은 제거가 가능하지만 제거하는 경우 관리되는 구성 정보, 데이터 및 앱이 모두 제거됩니다.

사용자는 Apple Configurator 2를 사용하여 직접 본인의 기기에 구성 프로파일을 설치할 수 있습니다. 또는, Safari를 통해 다운로드하거나 이메일을 통해 전송하거나 MDM 솔루션을 사용하여 무선으로 전송해 설치가 가능합니다. 사용자가 기기 등록 프로그램 또는 Apple School Manager에서 기기를 설정하면 해당 기기에서 MDM 등록 프로파일을 다운로드하고 설치합니다.

Mobile Device Management(MDM)

iOS의 MDM 지원으로 기업은 전사적으로 대규모의 iPhone, iPad, Apple TV 및 Mac 배포를 안전하게 설정하고 관리할 수 있습니다. MDM 기능은 구성 프로파일, 무선 등록 및 Apple 푸시 알림 서비스 같은 기존 iOS 기술을 기반으로 합니다. 예를 들어, APNS는 보안된 연결을 통해 직접 MDM 솔루션과 통신할 수 있도록 기기를 깨우는 데 사용됩니다. 기밀 또는 독점 정보는 APNS를 통해 전송되지 않습니다.

MDM을 사용하여 IT 부서는 기업 환경에 iOS 기기를 등록하고 무선으로 설정을 구성 및 업데이트하고 기업 정책 준수 여부를 감독하며 관리되는 기기를 원격으로 삭제하거나 잠글 수도 있습니다.

MDM에 대한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
www.apple.com/iphone/business/it/management.html

공유 iPad

공유 iPad는 교육용 iPad 배포에 사용되는 다중 사용자 모드입니다. 이를 통해 학생은 도큐멘트나 데이터를 공유할 필요 없이 iPad를 공유할 수 있습니다. 학생들에게 각자의 홈 디렉토리가 할당됩니다. 홈 디렉토리는 사용자의 인증서로 보호되는 APFS 볼륨으로 생성됩니다. 공유 iPad는 교육 기관에서 발급하고 소유하는 관리되는 Apple ID를 사용해야 합니다. 공유 iPad를 통하여 학생은 기관에서 소유하고 여러 학생이 사용하도록 구성한 모든 기기에 로그인할 수 있습니다.

학생의 데이터는 별도의 홈 디렉토리에 분할되어 있습니다. 각 홈 디렉토리는 각자의 데이터 보호 도메인에 있으며 UNIX 권한 및 샌드박스로 보호됩니다. 사용자가 로그인하면 관리되는 Apple ID는 SRP 프로토콜을 사용하여 Apple의 신원 서버를 통하여 인증됩니다. 인증에 성공하면 해당 기기에 특정한 임시 접근 토큰을 부여받습니다. 학생이 해당 기기를 사용한 적이 있는 경우, 로컬 사용자 계정을 이미 가지고 있으며 동일한 인증서를 사용하여 잠금 해제됩니다. 학생이 해당 기기를 사용한 적이 없다면 새로운 UNIX 사용자 ID, 사용자의 홈 디렉토리가 있는 APFS 볼륨, 올바른 키체인이 권한 설정됩니다. 기기가 인터넷에 연결되어있지 않으면(예: 학생이 현장 학습을 간 경우) 며칠의 정해진 기한 동안 로컬 계정이 인증될 수 있습니다. 이 경우 기존 로컬 계정을 가진 사용자만 로그인할 수 있으며, 로컬 계정이 있더라도 해당 기한이 지나면 온라인으로 인증해야 합니다.

학생의 로컬 계정이 잠금 해제되거나 새로 생성된 상태에서 원격으로 인증되었다면 Apple 서버에서 발급한 임시 토큰은 iCloud 로그인을 승인하는 iCloud 토큰으로 변환됩니다. 그런 다음, 학생의 설정이 복원되고 학생의 도큐멘트와 데이터가 iCloud로부터 동기화됩니다.

학생 세션이 진행 중이고 기기가 인터넷에 연결되어 있는 경우, 도큐멘트 및 데이터가 생성되거나 수정되면 iCloud에 저장됩니다. 추가로 학생이 로그아웃하는 경우, 백그라운드 동기화 메커니즘을 통해 변경사항을 iCloud로 푸시합니다. 해당 사용자의 백그라운드 동기화가 완료되면 APFS 볼륨은 마운트 해제되며 사용자의 인증서를 제공해야만 다시 마운트할 수 있습니다.

공유 iPad가 iOS 10.3 이전 버전에서 10.3 이상으로 업그레이드되면 1회성 파일 시스템 전환이 발생하여 HFS+ 데이터 파티션을 APFS 볼륨으로 전환합니다. 이때 시스템에 사용자 홈 디렉토리가 하나라도 있는 경우, 해당 홈 디렉토리는 개별 APFS 볼륨으로 전환되지 않고 메인 데이터 볼륨에 남습니다. 다른 학생이 추가로 로그인하면 그 학생의 홈 디렉토리도 메인 데이터 볼륨에 남습니다. 위에서 설명한 것과 같이 메인 데이터 볼륨의 모든 사용자 계정이 삭제되기 전까지는 새로운 사용자 계정이 각자의 APFS 볼륨으로 생성되지 않습니다. 그러므로 해당 사용자가 더 안전하게 보호받고 있으며 APFS로부터 할당 받았음을 확인하려면 해당 iPad를 지우고 재설치하는 방법을 통해 10.3 이상으로 업그레이드하거나, 사용자 삭제 MDM 명령으로 기기의 모든 사용자를 삭제해야 합니다.

Apple School Manager

Apple School Manager는 교육 기관을 위한 서비스로서 교육 기관에서 콘텐츠를 구입하거나 MDM 솔루션에 자동 기기 등록을 구성하거나 학생 및 교직원이 사용할 계정을 생성하거나 iTunes U 강의를 설정할 수 있습니다. Apple School Manager는 웹에서 접근이 가능하며 기술 관리자와 IT 관리자, 임직원 및 교사를 위해 디자인되었습니다.

Apple School Manager에 대한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
help.apple.com/schoolmanager

기기 등록

Apple School Manager 및 Apple Deployment Programs의 일부인 DEP(기기 등록 프로그램)는 Apple에서 직접 구입했거나 프로그램에 참여하는 Apple 공인 대리점 및 이동통신사를 통해 구입한 iOS 기기를 배포하는 데 빠르고 간소화된 방법을 제공합니다. 구입한 후 iOS 11 이상이 설치된 iOS 기기도 Apple Configurator 2를 사용하여 DEP에 추가될 수 있습니다.

조직은 직접 기기를 만지거나 기기를 먼저 준비할 필요 없이 자동으로 MDM에 기기를 등록할 수 있습니다. 프로그램에서 등록한 후, 관리자는 프로그램 웹 사이트에 로그인하여 해당 프로그램을 MDM 솔루션에 링크합니다. 그런 다음, 구입한 기기가 MDM을 통해 사용자에게 할당됩니다. 기기가 사용자에게 할당된 이후, MDM이 지정한 구성, 제한사항 또는 제어 설정이 자동으로 설치됩니다. 기기와 Apple 서버 간의 모든 통신은 전송 중에 HTTPS(SSL)를 통하여 암호화됩니다.

설정 지원에서 특정 단계를 제거하여 설정 과정을 더욱 단순하게 만들어 사용자가 기기를 더욱 빠르게 사용할 수도 있습니다. 관리자는 또한 사용자가 기기에서 MDM 프로파일을 제거할 수 있는지의 여부를 제어하고 최초 사용 시에 기기 제한이 설정되어 있는지 확인할 수도 있습니다. 기기가 활성화되면 기업의 MDM 솔루션에 등록되어 모든 관리 설정, 앱 및 책이 설치됩니다.

비즈니스 관련 자세한 정보를 보려면 아래 사이트로 이동하십시오.
help.apple.com/deployment/business

교육 기관 관련 자세한 정보를 보려면 아래 사이트로 이동하십시오.
help.apple.com/schoolmanager

참고: 기기 등록은 일부 국가나 지역에서 사용할 수 없습니다.

Apple Configurator 2

MDM뿐 아니라 macOS용 Apple Configurator 2를 사용하면 사용자에게 iOS 기기 및 Apple TV를 건네기 전에 손쉽게 기기를 설정하고 미리 구성할 수 있습니다. Apple Configurator 2로 기기에 앱, 데이터, 제한사항 및 설정 등을 빠르게 미리 구성할 수 있습니다.

Apple Configurator 2에서는 Apple School Manager(교육 기관용) 또는 기기 등록 프로그램(비즈니스용)을 사용하여 MDM 솔루션에 기기를 등록할 수 있습니다. 사용자가 설정 지원을 사용하여 기기를 등록하지 않아도 됩니다. 또한 Apple Configurator 2는 iOS 기기 및 Apple TV를 구입한 후 Apple School Manager 또는 기기 등록 프로그램에 추가할 수 있습니다.

Apple Configurator 2에 대한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
help.apple.com/configurator/mac

감독

기업은 기기를 설정하는 동안 기기가 감독을 받도록 설정할 수 있습니다. 감독을 통해 기기를 기업에서 소유하고 있음을 나타낼 수 있고, 설정 및 제한사항을 추가적으로 제어할 수 있습니다. 기기는 Apple School Manager, 기기 등록 프로그램 또는 Apple Configurator 2를 통해 설정 중에 감독을 받을 수 있습니다. 기기를 감독하려면 해당 기기를 지우고 운영 체제를 다시 설치해야 합니다.

MDM 또는 Apple Configurator 2를 사용하여 기기 설정 및 관리에 대한 자세한 정보를 보려면 아래 사이트로 이동하십시오.
help.apple.com/deployment/ios

제한사항

관리자가 제한사항을 활성화하여(일부 경우에는 비활성화 가능) 사용자가 기기의 특정 앱, 서비스 또는 기능을 이용하지 못하도록 제한할 수 있습니다. 제한사항은 구성 프로파일에 붙어있는 제한사항 페이로드에 있는 기기로 전송됩니다. 제한사항은 iOS, tvOS 및 macOS 기기에 적용될 수 있습니다. 관리되는 iPhone에서 특정 제한사항은 연결된 Apple Watch에도 반영됩니다.

현재 목록을 보려면(IT 관리자용) 아래 사이트로 이동하십시오.
help.apple.com/deployment/ios/#/apdbd6309354

원격 지우기

이 기능을 통해 관리자 또는 사용자가 iOS 기기를 원격으로 지울 수 있습니다. 삭제할 수 있는 저장 장치(Effaceable Storage)에서 블록 저장 공간 암호화 키를 안전하게 폐기하여 모든 데이터를 읽을 수 없도록 만드는 방법으로 원격 지우기가 이루어집니다. MDM, Exchange 또는 iCloud에서 이러한 원격 지우기 명령을 실행할 수 있습니다.

MDM이나 iCloud에서 원격 지우기 명령을 실행하면 기기는 명령을 확인하고 삭제를 수행합니다. Exchange를 통한 원격 지우기는 기기가 명령을 수행하기 전에 Exchange 서버를 통해 명령을 확인합니다.

사용자는 또한 설정 앱을 사용하여 가지고 있는 기기를 삭제할 수도 있습니다. 그리고 언급한 바와 같이 여러 번 암호 입력에 실패한 경우 기기가 자동으로 지워지도록 설정할 수 있습니다.

분실 모드

기기를 분실하거나 도난당한 경우, MDM 관리자가 iOS 9.3 이상이 설치된 감독 중인 기기에서 분실 모드를 원격으로 활성화할 수 있습니다. 분실 모드가 활성화되면 현재 사용자가 로그아웃되며 기기는 잠금 해제할 수 없게 됩니다. 기기를 발견하면 연락할 전화번호를 표시하는 것과 같이 관리자가 직접 입력한 메시지를 화면에 표시합니다. 기기가 분실 모드가 되면 관리자는 해당 기기의 현재 위치를 송신하도록 요청할 수 있습니다. 또한 원하는 경우 사운드도 재생할 수 있습니다. 분실 모드를 나갈 수 있는 유일한 방법으로는 관리자가 분실 모드를 끄는 것인데, 이 경우 해당 사용자는 분실 모드가 꺼진 것을 잠금 화면이나 홈 화면에 표시되는 알림으로 확인할 수 있습니다.

활성화 잠금

나의 iPhone 찾기가 켜져 있으면 소유자의 Apple ID 인증 정보 또는 기기의 이전 암호 없이는 기기를 활성화할 수 없습니다.

기업이 소유한 기기의 경우 개인 사용자가 각자의 Apple ID 인증 정보를 입력하여 기기를 재활성화하는 것보다 기기를 감독하여 활성화 잠금을 기업에서 직접 관리하는 것이 좋습니다.

감독 중인 기기의 경우, 호환되는 MDM 솔루션은 활성화 잠금이 활성화될 때 우회 코드를 저장합니다. 나중에 기기를 삭제하여 새로운 사용자에게 할당해야 할 때 이 코드를 사용하여 활성화 잠금을 자동으로 해제할 수 있습니다.

기본적으로 사용자가 나의 iPhone 찾기를 켜더라도 감독 중인 기기는 활성화 잠금이 활성화되지 않습니다. 하지만 MDM 솔루션이 우회 코드를 가져와 해당 기기에서 활성화 잠금이 활성화되도록 허용할 수 있습니다. 나의 iPhone 찾기가 켜져 있고 MDM 솔루션이 활성화 잠금을 활성화하는 경우 그 시점에 활성화 잠금이 활성화됩니다. 나의 iPhone 찾기가 꺼져 있고 MDM 서버가 활성화 잠금을 활성화하는 경우라면 다음에 사용자가 나의 iPhone 찾기를 활성화할 때 활성화 잠금이 활성화됩니다.

Apple School Manager를 통하여 생성한 '관리되는 Apple ID'를 이용하여 교육 환경에서 사용되는 기기의 경우, 활성화 잠금을 사용자 Apple ID 대신에 관리자 Apple ID에 연결하거나 기기의 우회 코드를 사용하여 비활성화시킬 수 있습니다.

개인 정보 보호 제어

Apple은 고객의 개인 정보를 중요하게 생각합니다. 다양한 제어 설정 및 옵션이 iOS에 내장되어 사용자가 앱이 이용할 정보의 내용뿐 아니라 해당 정보를 이용하는 방법과 시점을 결정할 수 있습니다.

위치 서비스

위치 서비스는 GPS, Bluetooth 및 크라우드 소스 Wi-Fi 핫스팟 및 통신탑의 위치를 이용하여 사용자의 대략적인 위치를 파악합니다. 위치 서비스를 설정에서 간단하게 끄거나 사용자가 위치 서비스를 사용하는 각 앱에 대한 접근을 승인할 수 있습니다. 앱을 사용하는 동안에만 위치 데이터를 받도록 요청하거나 계속 받도록 요청할 수도 있습니다. 또한 사용자는 위치 서비스에 대한 접근을 허용하지 않도록 선택할 수 있으며 설정에서 언제든지 변경 가능합니다. 설정에서 앱의 위치 사용 요청에 따라 접근 허용을 안 함, 사용하는 동안 또는 항상으로 설정할 수 있습니다. 또한 언제든지 위치를 사용할 수 있는 앱이 백그라운드 모드에 들어가게 되면 사용자에게 위치 승인에 대한 알림이 전송되고 사용자는 앱의 위치 서비스 접근을 변경할 수도 있습니다.

또한, 사용자는 시스템 서비스의 위치 정보 사용을 세부적으로 제어할 수 있습니다. 이는 iOS 및 서비스를 향상시키기 위해 Apple에서 사용한 분석 서비스를 통해 수집된 정보에서 위치 정보를 포함하는 설정을 끄는 것을 포함합니다. 또한 관련 서비스는 위치 기반의 Siri 정보, Siri 제안 검색을 위한 위치 기반 정보, 지역 교통 상황 및 과거에 방문한 주요 위치를 포함합니다.

개인 데이터 접근

iOS에서는 권한 없이 앱이 사용자의 개인 정보에 접근할 수 없습니다. 또한, 설정에서 사용자는 특정 정보에 대한 접근 권한이 있는 앱을 확인할 수 있으며 접근을 승인하거나 취소할 수도 있습니다. 접근 권한이 있는 정보는 다음과 같습니다.

- 연락처
- 캘린더
- 미리 알림
- 사진
- 동작 활동 및 피트니스
- 위치 서비스
- Apple Music
- 사용자의 음악 및 비디오 활동
- 트위터와 Facebook 같은 소셜 미디어 계정
- 마이크
- 카메라
- HomeKit
- 건강
- 음성 인식
- Bluetooth 공유
- 사용자의 미디어 보관함

사용자가 iCloud에 로그인하면 앱은 기본적으로 iCloud Drive에 접근할 수 있습니다. 사용자는 설정의 iCloud 패널에서 각 앱의 iCloud Drive에 대한 접근을 제어할 수 있습니다. 또한, iOS는 앱 간의 데이터 이동과 MDM 솔루션에서 설치한 계정 및 사용자가 설치한 계정 간에 데이터 이동을 제한할 수 있습니다.

개인정보 취급방침

Apple의 개인정보 취급방침을 보려면 아래 사이트로 이동하십시오.
www.apple.com/kr/legal/privacy

Apple 보안 포상금

Apple에서는 중대한 문제를 Apple에게 알리는 연구자에게 포상금을 제공합니다. Apple 보안 포상금을 받으려면 연구자는 정확한 보고서와 개념의 잠정적인 증명을 제공해야 합니다. 해당 취약점은 최신 iOS 및 관련된 최신 하드웨어에 영향을 미쳐야 합니다. 정확한 포상금의 액수는 Apple에서 심사 후에 결정합니다. 심사 기준에는 참신함, 노출 가능성 및 사용자 상호 작용 필요 정도가 포함됩니다.

문제가 적절히 공유되면 Apple에서는 확인된 문제를 최대한 빨리 해결할 수 있도록 우선적으로 노력합니다. 연구자가 기밀로 요청하지 않는 한 Apple에서는 적합한 경우 연구자의 실적을 공개합니다.

카테고리	최대 포상금(USD)
보안 부팅 펌웨어 구성요소	$200,000
보안 엔클레이브에서 보호하는 기밀 자료 추출	$100,000
커널 권한으로 임의 코드 실행	$50,000
Apple 서버에서 iCloud 계정 데이터에 대한 승인되지 않은 접근	$50,000
샌드박스 프로세스에서 샌드박스 밖에 있는 사용자 데이터에 접근	$25,000

결론

보안에 대한 노력

Apple은 고객을 보호하기 위해 최선을 다하고 있습니다. 개인 정보 보호 및 보안의 첨단 기술을 통해 개인 정보를 보호하고 포괄적인 방법으로 기업 환경에서 기업 데이터를 보호하기 위해 노력합니다.

보안은 iOS에 내장되어 있습니다. 플랫폼, 네트워크, 앱에 이르기까지, 기업이 원하는 모든 것이 iOS 플랫폼에 있습니다. 더불어 이러한 구성요소를 통해 iOS는 사용자 환경을 저해하지 않고 업계 최고의 보안 기술을 제공합니다.

Apple은 iOS와 iOS 앱 생태계 전반에 일관된 통합 보안 인프라를 사용합니다. 하드웨어 기반의 저장 공간 암호화는 기기가 분실된 경우에 원격 지우기 기능을 제공하여 기기를 판매하거나 다른 소유자에게 전달하는 경우 사용자가 모든 기업 및 개인 정보를 완전히 삭제할 수 있습니다. 진단 정보 또한 익명으로 수집됩니다.

Apple에서 제작한 iOS 앱은 보안 강화를 염두에 두고 개발되었습니다. 예를 들어 iMessage 및 FaceTime은 클라이언트간 암호화를 제공합니다.

타사 앱에는 코드 서명, 샌드박스 및 권한의 조합이 사용되어 바이러스, 악성 코드 및 기타 취약점으로부터 사용자를 업계 최고로 안전하게 보호합니다. App Store 제출 과정에서 모든 iOS 앱은 사용자에게 공개되기 전에 심사를 거치므로 이러한 위험으로부터 사용자를 더욱 보호합니다.

iOS에 내장된 광범위한 보안 기능을 최대한 활용하기 위해 기업이 IT 및 보안 정책을 검토하여 iOS 플랫폼에서 제공하는 보안 기술 계층을 충분히 활용하도록 권장합니다.

Apple은 모든 Apple 제품을 지원하기 위한 전용 보안 팀을 유지합니다. 보안 팀은 출시된 제품 및 개발 중인 제품에도 보안 감사와 테스트를 제공합니다. Apple 팀은 또한 보안 도구와 교육을 제공하고 새로운 보안 문제와 위험 리포트를 적극적으로 모니터링합니다. Apple은 FIRST(Forum of Incident Response and Security Teams)의 회원입니다.

Apple에 문제를 보고하고 보안 알림을 구독하는 것에 대해 알아보려면 아래 사이트로 이동하십시오.
www.apple.com/kr/support/security

용어집

ASLR(Address Space Layout Randomization)	소프트웨어 버그가 공격하기 어렵도록 만드는 iOS에 채용된 기술. 메모리 주소와 오프셋을 예측할 수 없게 하여 악성 코드가 이 값을 하드 코드할 수 없습니다. iOS 5 이상 버전에서 모든 시스템 앱과 라이브러리는 위치 독립 실행 파일로 컴파일된 모든 타사 앱과 함께 무작위로 위치가 지정됩니다.
APNS(Apple 푸시 알림 서비스)	푸시 알림을 iOS 기기에 전달하는 Apple이 제공하는 전 세계적인 서비스.
부트 ROM	기기가 처음 부팅할 때 기기의 프로세서가 실행하는 최초의 코드. 프로세서의 필수불가결한 부분으로서 Apple이나 공격자가 변경할 수 없습니다.
데이터 보호	iOS의 파일과 키체인 보호 방식. 앱이 파일과 키체인 항목을 보호하는 데 사용하는 API를 지칭하기도 합니다.
DFU(기기 펌웨어 업그레이드)	기기의 부트 ROM 코드가 USB를 통해 복구되도록 대기하는 모드. 화면이 검은색으로 표시되지만 iTunes를 실행 중인 컴퓨터에 연결하는 즉시 다음 메시지가 표시됩니다. 'iTunes가 복구 모드에 있는 iPad를 발견했습니다. iTunes에서 사용하기 전에 이 iPad를 복구해야 합니다.'
ECID	각 iOS 기기의 프로세서별로 고유한 64비트 식별자. 한 기기에서 전화를 받는 경우 가까이에 있는 iCloud로 연결된 기기의 벨소리가 Bluetooth LE 4.0 광고를 받아 바로 꺼집니다. 광고 바이트는 Handoff 광고와 같은 방식을 통하여 암호화됩니다. 개인화 프로세스의 일부로 사용되며 비밀 정보로 처리되지 않습니다.
삭제할 수 있는 저장 장치(Effaceable Storage)	NAND 저장 공간의 전용 영역, 암호화 키를 저장하는 데 사용되며 직접 주소를 지정하고 안전하게 삭제됩니다. 공격자가 기기를 물리적으로 소유하는 경우에 보호를 제공하지는 않지만 삭제할 수 있는 저장 장치(Effaceable Storage)에 보관된 키를 키 계층의 부분으로 사용하여 빠른 삭제를 진행하여 전방향 안전성을 제공합니다.
파일 시스템 키	클래스 키를 포함하여 각 파일의 메타데이터를 암호화하는 키. 삭제할 수 있는 저장 장치(Effaceable Storage)에 보관되어 기밀성 유지보다는 빠른 삭제를 진행할 수 있습니다.
GID(그룹 ID)	UID와 비슷하지만 한 클래스의 모든 프로세서에 공통입니다.
HSM(하드웨어 보안 모듈)	디지털 키 보호 및 관리에 전문화된 변경 방지 컴퓨터.
iBoot	LLB로 로드된 다음 보안 부팅 체인의 부분으로 XNU를 로드하는 코드.
IDS(ID 서비스)	키와 기기 주소를 찾는 데 사용되는 iMessage 공개 키, APNS 주소, 전화번호 및 이메일 주소의 Apple 디렉토리.
IC(집적 회로)	마이크로칩으로 알려짐.
JTAG(Joint Test Action Group)	프로그래머와 회로 개발자들이 사용하는 표준 하드웨어 디버깅 도구.
Keybag	클래스 키의 모음을 저장하는 데 사용되는 데이터 구조. 각 유형(User, Device, System, Backup, Escrow 및 iCloud Backup)에는 다음과 같은 동일한 포맷이 있습니다. • 헤더에 포함된 내용: – 버전(iOS 5에서 3으로 설정) – 유형(시스템, 백업, 에스크로 또는 iCloud 백업) – Keybag UUID – HMAC(Keybag이 서명된 경우) – 클래스 키를 래핑하는 데 사용되는 방법: 솔트 및 반복 횟수와 함께 UID 또는 PBKDF2를 사용하여 탱글링 • 클래스 키의 목록: – 키 UUID – 클래스(파일 또는 키체인 데이터 보호 클래스) – 래핑 유형(UID 파생 키 전용, UID 파생 키와 암호 파생 키) – 래핑된 클래스 키 – 비대칭 클래스의 공개 키

키체인	암호, 키 및 기타 민감한 자격 증명 정보를 저장하고 검색하기 위해 iOS와 타사 앱이 사용하는 인프라와 API 세트.
키 래핑	한 키를 다른 키로 암호화하는 방법. iOS는 RFC 3394와 같이 NIST AES 키 래핑을 사용합니다.
저레벨 부트로더(LLB)	부트 ROM이 호출하고 차례로 보안 부팅 체인의 부분으로 iBoot를 로드하는 코드.
파일별 키	파일 시스템에서 파일을 암호화하는 데 사용되는 AES 256비트 키. 파일별 키는 클래스 키로 래핑되고 파일의 메타데이터에 저장됩니다.
권한 설정 프로파일	iOS 기기에 앱을 설치하고 테스트하도록 허용하는 권한과 항목 세트를 포함하는 Apple이 서명한 plist. 개발 권한 설정 프로파일은 개발자가 임시 배포를 위해 선택한 기기의 목록을 표시합니다. 배포 권한 설정 프로파일은 기업이 개발한 앱의 앱 ID를 포함합니다.
융선 흐름 각도 매핑	지문의 일부에서 추출한 융선의 방향과 너비의 수학적 표현.
스마트 카드	보안 ID, 인증 및 데이터 저장 공간을 제공하는 내장된 집적 회로.
SoC(System on a chip)	여러 구성요소를 단일 칩으로 통합한 집적 회로(IC). 보안 엔클레이브는 Apple의 A7 또는 이상 버전의 중앙 프로세서 내에 있는 SoC입니다.
탱글링	사용자의 암호가 암호화 키로 전환되어 기기의 UID로 강화되는 과정. 무작위 대입 공격은 정해진 기기에서 수행되므로 속도가 제한되어 탱글링과 동시에 수행될 수 없습니다. 탱글링 알고리즘은 PBKDF2입니다. PBKDF2는 PRF(의사 난수 함수)로서 각 반복에 대해 기기 UID와 키로 연결된 AES를 사용합니다.
URI(Uniform Resource Identifier)	웹 기반의 리소스를 식별하는 문자 스트링.
UID(고유 ID)	제조 시 각 프로세서에 각인되는 256비트 AES 키. 펌웨어나 소프트웨어가 읽을 수 없고 프로세서의 하드웨어 AES 엔진만 사용할 수 있습니다. 실제 키를 받기 위해 공격자는 프로세서의 실리콘에 대해 매우 복잡하고 고가의 물리적 공격을 마운트해야 합니다. UID는 UDID 뿐만 아니라 기기의 모든 식별자와 관련이 없습니다.
XNU	iOS와 macOS 운영 체제의 핵심 커널. 신뢰받는 것으로 간주되어 코드 서명, 샌드박스, 권한 검사 및 ASLR 같은 보안책을 시행합니다.

도큐멘트 수정 내역

날짜	요약
2018년 1월	**iOS 11.2용으로 업데이트됨** • Apple Pay Cash **iOS 11.1용으로 업데이트됨** • 보안 인증서 및 프로그램 • Touch ID/Face ID • 공유 메모 • CloudKit 엔드 투 엔드 암호화 • TLS • Apple Pay, 웹에서 Apple Pay로 결제하기 • Siri 제안 • 공유 iPad • iOS 11의 보안 콘텐츠에 대한 자세한 정보를 보려면 다음 사이트 참조: support.apple.com/ko-kr/HT208112
2017년 7월	**iOS 10.3용으로 업데이트됨** • 시스템 엔클레이브 • 파일 데이터 보호 • Keybag • 보안 인증서 및 프로그램 • SiriKit • HealthKit • 네트워크 보안 • Bluetooth • 공유 iPad • 분실 모드 • 활성화 잠금 • 개인 정보 보호 제어 • iOS 10.3의 보안 콘텐츠에 대한 자세한 정보를 보려면 다음 사이트 참조: support.apple.com/ko-kr/HT207617
2017년 3월	**iOS 10용으로 업데이트됨** • 시스템 보안 • 데이터 보호 클래스 • 보안 인증서 및 프로그램 • HomeKit, ReplayKit, SiriKit • Apple Watch • Wi-Fi, VPN • 단일 로그인 • Apple Pay, 웹에서 Apple Pay로 결제하기 • 신용 카드, 직불 카드 및 선불 카드 권한 설정 • Safari 제안 • iOS 10의 보안 콘텐츠에 대한 자세한 정보를 보려면 다음 사이트 참조: support.apple.com/ko-kr/HT207143

날짜	요약
2016년 5월	**iOS 9.3용으로 업데이트됨** • 관리되는 Apple ID • Apple ID용 이중 인증 • Keybag • 보안 인증서 • 분실 모드, 활성화 잠금 • 보안 메모 • Apple School Manager, 공유 iPad • iOS 9.3의 보안 콘텐츠에 대한 자세한 정보를 보려면 다음 사이트 참조: support.apple.com/ko-kr/HT206166
2015년 9월	**iOS 9용으로 업데이트됨** • Apple Watch 활성화 잠금 • 암호 정책 • Touch ID API 지원 • A8의 데이터 보호가 AES-XTS 사용함 • 자동 소프트웨어 업데이트를 위한 Keybag • 인증서 업데이트 • 기업 앱 신뢰 모델 • Safari 책갈피를 위한 데이터 보호 • 앱 전송 보안 • VPN 사양 • HomeKit용 iCloud 원격 접근 • Apple Pay 적립 카드, Apple Pay 카드 발급처 앱 • Spotlight 기기 내 인덱스 • iOS 페어링 모델 • Apple Configurator 2 • 제한사항 • iOS 9의 보안 콘텐츠에 대한 자세한 정보를 보려면 다음 사이트 참조: support.apple.com/ko-kr/HT205212

2018년 1월