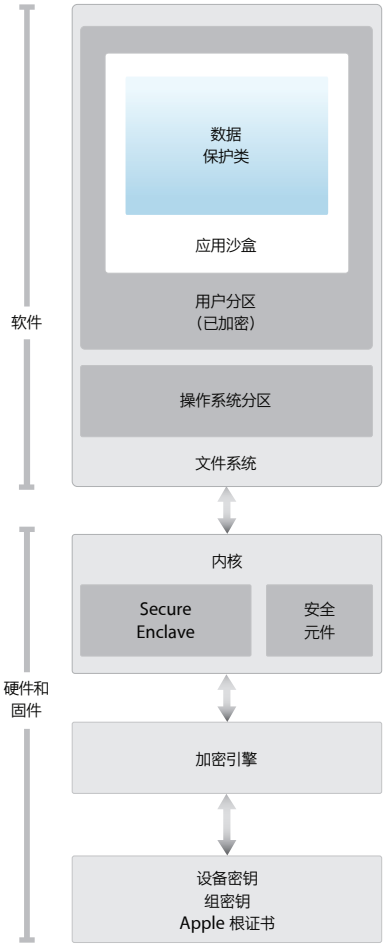
iOS 安全保护

适用于 iOS 9.0 或更高版本

2015 年 9 月

目录

介绍



iOS 安全性架构图以直观的图表形式概述了本文要探讨的各类技术。

Apple 设计的 iOS 平台以安全性为核心。着手于开发一流的移动平台时，我们充分利用了数十年积累的丰富经验，力求打造出全新的架构。在深入思考桌面环境中的诸多安全性隐患后，我们在 iOS 的设计中建立了一个全新的安全保护机制。我们开发并整合了一系列有助于增强移动环境安全性的创新功能，可在默认情况下为整个系统提供保护。这一切使得 iOS 在移动设备安全领域迈出了更深远的一步。

软件、硬件和服务在每台 iOS 设备上紧密联系、共同工作，旨在为用户提供最高的安全性和透明的体验。iOS 不仅保护设备和其中的静态数据，还保护整个生态系统，包括用户在本地、网络上以及使用互联网核心服务执行的所有操作。

iOS 和 iOS 设备不但提供先进的安全性功能，而且还易于使用。很多安全性功能在默认情况下均处于启用状态，因此 IT 部门无需执行大量的配置操作。而设备加密等关键的安全性功能是不可修改的，因此可以避免用户在无意中停用这些功能。Touch ID 等其他功能让设备安全性变得更简单直观，从而改善了用户体验。

本文详细介绍了安全性技术和功能如何在 iOS 平台中得以实现。在本文的帮助下，各个公司能够将 iOS 平台安全性技术和功能与自身的政策和规程结合在一起，从而满足公司的特定安全性需求。

本文主要分为以下几个主题：

- **系统安全性**：iPhone、iPad 和 iPod touch 上安全的一体化软硬件平台。
- **加密和数据保护**：当设备丢失或被盗，或有未授权人员尝试使用或修改设备时，对用户数据进行保护的架构和设计。
- **应用安全性**：确保应用安全运行，同时又不破坏平台完整性的系统。
- **网络安全性**：针对传输中的数据提供安全认证和加密的工业标准联网协议。
- **Apple Pay**：Apple 推行的安全支付方式。
- **互联网服务**：Apple 基于网络技术架构提供信息通信、同步和备份。
- **设备控制**：防止在未授权的情况下使用设备的方法，以及当设备丢失或被盗时能够进行远程擦除的方法。
- **隐私控制**：iOS 中可用于控制“定位服务”和用户数据访问权限的功能。

系统安全性

进入设备固件升级 (DFU) 模式
进入 DFU 模式后恢复设备，可使设备恢复到已知的正常状态，该状态下只存在未修改的 Apple 签名的代码。可通过以下方式手动进入 DFU 模式：首先，使用 USB 线缆将设备连接到电脑，然后同时按住主屏幕按钮和睡眠/唤醒按钮。8 秒钟之后，继续按住主屏幕按钮的同时，松开睡眠/唤醒按钮。注：设备处于 DFU 模式时，屏幕上不会显示任何内容。如果显示 Apple 标志，表示按住睡眠/唤醒按钮的时间过长。

系统安全性旨在确保每台 iOS 设备的所有核心组件都能为软件和硬件提供安全保护。这包括启动过程、软件更新和 Secure Enclave。此架构是 iOS 安全体系的核心，并且不会影响设备的正常使用。

iOS 设备的硬件和软件实现了紧密集成，可确保系统的每个组件均获得信任，并对系统进行整体验证。从初始启动到 iOS 软件更新，再到第三方应用，每个步骤都经过分析和审查，确保硬件和软件以最优化的方式协同工作，并以恰当的方式使用资源。

安全启动链

启动过程每个步骤包含的组件都经 Apple 加密签名以确保其完整性，只有在验证信任链后，每个步骤才能继续。这些组件包括引导加载程序、内核、内核扩展项和基带固件。

打开 iOS 设备后，其应用程序处理器会立即执行只读内存（称为 Boot ROM）中的代码。这些不可更改的代码（称为硬件的信任根）是在制造芯片时设好的，为隐式受信任代码。Boot ROM 代码包含 Apple 根 CA 公钥，该公钥用于验证底层引导加载程序 (LLB) 是否经过 Apple 签名，以决定是否允许其加载。这是信任链中的第一步，信任链中的每个步骤都确保下一步骤获得 Apple 的签名。当 LLB 完成其任务后，它会验证并运行下一阶段的引导加载程序 iBoot，后者又会验证并运行 iOS 内核。

此安全启动链有助于确保底层的软件未被篡改，并只允许 iOS 运行在经过验证的 Apple 设备上。

对于可接入蜂窝移动网络的设备，基带子系统也使用其类似的安全启动过程，包括已签名的软件以及由基带处理器验证的密钥。

对于搭载 A7 或更高版本 A 系列处理器的设备，Secure Enclave 协处理器还会使用安全启动过程，用以确保其单独的软件经过 Apple 验证和签名。

如果该启动过程中的某个步骤无法加载或验证下一过程，启动过程会停止，设备屏幕会显示“连接到 iTunes”。这就是所谓的恢复模式。如果 Boot ROM 无法加载或验证 LLB，它会进入 DFU（设备固件升级）模式。这两种情况下，设备都必须通过 USB 连接到 iTunes，并恢复为出厂默认设置。有关手动进入恢复模式的更多信息，请访问 support.apple.com/kb/HT1808?viewlocale=zh_CN。

系统软件授权

Apple 会定期发布软件更新以解决新出现的安全性问题，并提供全新功能；此类更新会同时提供给所有受支持的设备。用户会在设备上和 iTunes 中看到 iOS 更新通知，更新通过无线方式发送，旨在鼓励用户尽快应用最新的安全性修正。

上述启动过程有助于确保设备上只能安装 Apple 签名的代码。为避免设备降级为缺少最新安全性更新的早期版本，iOS 采用了名为**“系统软件授权”**的过程。如果可以将设备降级，攻击者一旦有了设备的控制权，便会安装早期版本的 iOS，并利用旧版本中未修复的漏洞来进行破坏。

对于搭载 A7 或更高版本 A 系列处理器的设备，Secure Enclave 协处理器还会利用“系统软件授权”来确保软件的完整性，并防止降级安装。请参阅下面的“Secure Enclave”。

iOS 软件更新可通过 iTunes 安装，也可采用无线 (OTA) 方式直接在设备上安装。如果通过 iTunes 安装更新，系统会下载并安装完整的 iOS 副本。如果采用 OTA 方式安装软件更新，系统将仅下载完成更新所需的组件，而不是下载整个操作系统，这样可有效提升网络效率。此外，软件更新还可以缓存在安装有 OS X Server 且运行缓存服务的本地网络服务器上，这样 iOS 设备无需访问 Apple 服务器，即可获取必要的更新数据。

在 iOS 升级过程中，iTunes（若采用 OTA 软件更新方式，则为设备本身）会连接到 Apple 安装授权服务器，并向其发送以下数据：要安装的安装包各部分（例如，LLB、iBoot、内核及操作系统映像）的加密测量值列表、一个随机的反重放值（随机数）以及设备的唯一 ID (ECID)。

授权服务器将提供的测量值列表与允许安装的版本进行比较，如果找到匹配项，就会将 ECID 添加到测量值并对结果进行签名。作为升级过程的一部分，服务器会将完整的一组已签名数据传递给设备。添加 ECID 可为请求设备“个性化”授权。通过仅对已知测量值授权和签名，服务器可确保更新的内容与 Apple 所提供的完全相同。

启动时信任链评估用于验证签名是否来自 Apple，并结合设备的 ECID 来确认从磁盘加载的项目测量值是否与该签名包含的内容相匹配。

这些步骤可确保针对特定设备进行授权，并且旧版 iOS 无法从一台设备拷贝到另一台设备。随机数可阻止攻击者存储服务器的响应和利用该响应来破坏设备或通过其他方式篡改系统软件。

Secure Enclave

Secure Enclave 是 Apple A7 或更高版本 A 系列处理器中集成的协处理器。它独立于应用处理器之外，并利用自己的安全启动和个性化软件更新。它为数据保护密钥管理提供所有加密操作，即使在内核遭到入侵的情况下，也可维护数据保护的完整性。

Secure Enclave 使用加密内存，并包含一个硬件随机数生成器。其微内核基于 L4 系列，并由 Apple 进行改进。Secure Enclave 与应用程序处理器之间的通信被隔离到一个中断驱动的信箱以及共享的内存数据缓冲区。

每个 Secure Enclave 在制造期间均预置了自身的 UID（唯一 ID），此 UID 无法由系统的其他部分访问，并且对 Apple 也是未知的。设备启动时会创建一个临时密钥，此密钥与 UID 配合使用，用于对设备内存空间的 Secure Enclave 部分进行加密。

此外，由 Secure Enclave 存储到文件系统的数据还会通过与 UID 配合使用的密钥以及反重放计数器进行加密。

Secure Enclave 负责处理来自 Touch ID 传感器的指纹数据，确定是否存在与注册的指纹相匹配的指纹数据，然后代表用户允许访问或购买。处理器和 Touch ID 传感器之间的通信通过串行外围接口总线实现。处理器将数据转发到 Secure Enclave，但处理器本身无法读取这些数据。数据通过会话密钥进行加密和认证，该密钥通过为 Touch ID 传感器和 Secure Enclave 预置的设备共享密钥进行协商。会话密钥交换针对双方使用 AES 密钥封装，并提供一个用于建立会话密钥并使用 AES-CCM 传输加密的随机密钥。

Touch ID

Touch ID 是指纹感应系统，有助于更快、更轻松地对设备进行安全的访问。此技术可从任意角度读取指纹数据，随着传感器每次使用时识别出更多重叠的节点而不断扩大指纹图，逐步提高对用户指纹识别的能力。

Touch ID 让使用更长、更复杂的密码变得更加实际，因为用户无需频繁地输入它。Touch ID 还克服了基于密码进行锁定的不便，它并不取代密码锁定机制，而是允许在精心设计的范围和时间限制内安全地访问设备。

Touch ID 和密码

要使用 Touch ID，用户必须对设备进行相应的设置，令设备需要密码来解锁。当 Touch ID 扫描并可识别已注册的指纹时，设备便会自动解锁，而无需用户输入设备密码。用户随时都可以使用密码来代替 Touch ID，并且在以下情况下必须使用密码：

- 设备刚刚开机或重新启动。
- 设备未解锁的时间超过 48 小时。
- 设备收到了远程锁定命令。
- 尝试五次后未能成功匹配指纹。
- 设置 Touch ID 或为其注册新指纹时。

启用 Touch ID 后，设备会在按下睡眠/唤醒按钮时立即锁定。在仅使用密码的安全机制下，许多用户会设置解锁宽限期，以免每次使用设备时都输入密码。使用 Touch ID 后，设备每次进入睡眠模式时都会锁定，而每次唤醒时都需要扫描指纹，或输入密码。

用户可以训练 Touch ID 识别多达五个不同的指纹。对于一个已注册的指纹，该指纹与他人指纹出现随机匹配的概率为五万分之一。但是，Touch ID 只允许五次不成功的指纹匹配尝试，然后用户必须输入密码才能获取访问权限。

Touch ID 的其他用途

用户还可以对 Touch ID 进行配置，以便批准从 iTunes Store、App Store 和 iBooks Store 购买项目，省去每次都要输入 Apple ID 密码的麻烦。当他们选择对某项购买进行授权时，设备和商店之间会交换认证令牌。令牌和加密随机数都存储在 Secure Enclave 中。随机数通过由所有设备和 iTunes Store 共享的 Secure Enclave 密钥进行签名。

Touch ID 还可配合 Apple 推行的安全支付方式 Apple Pay 一起使用。有关更多信息，请参阅本文中的 Apple Pay 部分。

此外，第三方应用可以使用系统提供的 API，要求用户使用 Touch ID 或密码进行认证。应用只会收到认证是否成功的通知，而无法访问 Touch ID 或与已注册指纹相关的数据。

钥匙串项也可使用 Touch ID 进行保护，而只有通过指纹匹配或设备密码，Secure Enclave 才会将其释出。应用开发者也可以通过 API 来确定密码由用户所设，由此允许使用 Touch ID 认证或解锁钥匙串项。

借助 iOS 9，开发者可以要求 Touch ID API 操作不后退到应用程序密码或者设备密码。同时由于能够取回表示录入指纹的手指状态信息，这允许在注重安全性的应用中将 Touch ID 用作第二重身份。

Touch ID 安全性

只有当主屏幕按钮周围的电容金属环检测到手指触摸时，指纹传感器才会启动，从而触发先进的成像阵列扫描手指，并将扫描结果发送至 Secure Enclave。

光栅扫描结果会临时储存在 Secure Enclave 的加密内存中，同时系统会对其进行向量化处理以便分析，然后将删除相关数据。此分析采用皮下纹路走向角度映射，这是一种有损过程，会在分析完成后删除用于重建用户实际指纹的详细数据。最终生成的节点图以一种只能由 Secure Enclave 读取的加密格式进行储存，不包含任何身份信息并且绝不会发送给 Apple 或备份至 iCloud 或 iTunes。

Touch ID 如何解锁 iOS 设备

如果 Touch ID 已关闭，当设备锁定时，保存在 Secure Enclave 中“全面保护”类的数据保护密钥将被丢弃。除非用户输入密码来解锁设备，否则不允许访问该类中的文件和钥匙串项。

如果 Touch ID 已开启，当设备锁定时，这些密钥不会被丢弃，而是通过提供给 Secure Enclave 中的 Touch ID 子系统的密钥进行封装。当用户尝试解锁设备时，如果 Touch ID 可以识别用户的指纹，它将提供用于解封数据保护密钥的密钥，从而使设备得到解锁。此过程要求数据保护和 Touch ID 子系统相互配合以解锁设备，因此提供了额外的保护。

使用 Touch ID 解锁设备时所需的密钥在设备重新启动后会丢失，而且 48 小时或五次 Touch ID 识别尝试失败后，该密钥也会被 Secure Enclave 丢弃。

加密和数据保护

安全启动链、代码签名和运行时进程安全性都有助于确保只有受信任的代码及应用可以在设备上运行。iOS 还有更多加密和数据保护功能来保护用户数据的安全，即使安全性基础架构的其他部分遭到入侵（例如，在设备上进行未授权的修改）。这对于用户和 IT 管理员都大有助益，它可始终保护个人和企业信息，并提供了设备被盗或丢失时将它立即彻底远程擦除的方法。

硬件安全性功能

在移动设备上，速度和节能至关重要。加密操作非常复杂，如果在设计和实施时未考虑这两个重要因素，可能会带来一些性能或电池续航方面的问题。

每台 iOS 设备都配备了专用的 AES 256 加密引擎，它内置于闪存与主系统内存之间的 DMA 路径中，可以实现高效的文件加密。

设备的唯一 ID (UID) 和设备组 ID (GID) 是 AES 256 位密钥，密钥已在制造过程中被固化 (UID) 或编译 (GID) 在应用程序处理器和 Secure Enclave 中。任何软件或固件都无法直接读取这些 ID；而只能查看由植入于硅片中的专用 AES 引擎将 UID 或 GID 用作密钥所执行的加密或解密操作结果。此外，只有专用于 Secure Enclave 的 AES 引擎才能使用 Secure Enclave 的 UID 和 GID。每台设备的 UID 都是唯一的，Apple 或其任何其供应商都没有记录在案。GID 对于同一类设备（例如，使用 Apple A8 处理器的所有设备）的所有处理器是通用的，它可用于非高安全性的任务，例如在安装和恢复过程中交付系统软件。将这些密钥集成到硅片中可防止篡改或绕过它们，或在 AES 引擎外部对其进行访问。UID 和 GID 也不可以通过 JTAG 或其他调试接口使用。

利用 UID 可以采用加密方式将数据与特定设备捆绑起来。例如，用于保护文件系统的密钥层次结构包括 UID，因此如果将内存芯片从一台设备整个移至另一台设备，文件将不可访问。UID 与设备上的任何其他标识符都无关。

抹掉所有内容和设置

“设置”中的“抹掉所有内容和设置”选项可以清除可擦除存储器中的所有密钥，通过加密方式使设备上的所有用户数据不可访问。因此，它是在将设备送给别人或送修时确保从设备中移除所有个人信息的理想方式。重要事项：除非设备已备份，否则请勿使用“抹掉所有内容和设置”选项，因为抹掉的数据无法恢复。

除了 UID 和 GID，所有其他加密密钥都由系统的随机数生成器 (RNG) 使用基于 CTR_DRBG 的算法创建。系统熵是在启动期间从时间变化以及设备启动后从中断计时中生成的。在 Secure Enclave 内部生成的密钥使用真正的硬件随机数生成器，通过基于 CTR_DRBG 处理的多个环形振荡器创建而成。

安全抹掉存储的密钥与生成它们一样重要。在闪存上执行这项操作尤其具有挑战性，因为损耗均衡 (wear-leveling) 可能意味着需要抹掉多份数据副本。为了解决该问题，iOS 设备加入了一种专用于安全擦除数据的功能，称为可擦除存储器。此功能利用基础存储技术（例如 NAND）直接进行非常低级别的寻址并抹掉少量数据块。

文件数据保护

除了 iOS 设备内置的硬件加密功能，Apple 还使用称为数据保护的技术，进一步保护储存于设备闪存中的数据。数据保护使设备不但可以响应来电等常见事件，还可以针对用户数据实现高级加密。诸如“信息”、“邮件”、“日历”、“通讯录”、“照片”和“健康”数据值等主要系统应用在默认情况下使用数据保护，而 iOS 7 或更高版本上安装的第三方应用则可以自动获得此保护。

数据保护是通过构建和管理密钥层次结构来实现的，并建立在每台 iOS 设备的硬件加密技术基础上。它通过将某个类分配给每个文件来实现对文件的逐个控制；可访问性取决于该类密钥是否已解锁。

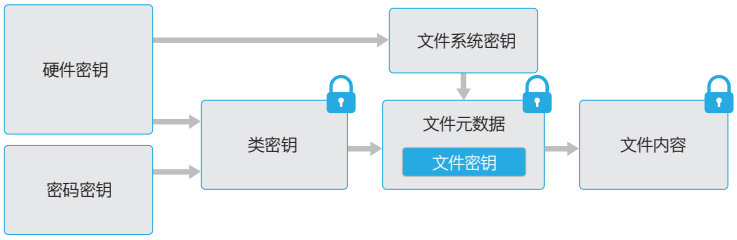
架构概览

每次在数据分区中创建文件时，数据保护都会创建一个新的 256 位密钥（“文件独有”密钥），并将其提供给硬件 AES 引擎，此引擎会使用该密钥并采用 AES CBC 模式对写入闪存的文件进行加密。（在配备 A8 处理器的设备上，使用了 AES-XTS。）初始化向量 (IV) 使用文件块偏移量进行计算，它使用文件独有密钥的 SHA-1 哈希值进行加密。

根据每个文件的可访问性，文件独有密钥使用其中一个类密钥进行封装。像所有其他封装一样，这是使用 NIST AES 密钥封装根据 RFC 3394 来执行的。封装的文件独有密钥储存在文件的元数据中。

当打开一个文件时，系统会使用文件系统密钥解密文件的元数据，以显露出封装的文件独有密钥和表示它受哪个类保护的记号。文件独有密钥使用类密钥解封，然后提供给硬件 AES 引擎，该引擎会在从闪存中读取文件时对文件进行解密。所有封装文件密钥的处理发生在 Secure Enclave 中；文件密钥绝不会直接透露给应用程序处理器。启动时，Secure Enclave 与 AES 引擎协商得到一个临时密钥。当 Secure Enclave 解开文件密钥时，它们又通过该临时密钥再次封装，然后发送回应用程序处理器。

文件系统中所有文件的元数据都使用随机密钥进行加密，该密钥在首次安装 iOS 或用户擦除设备时创建。文件系统密钥储存在可擦除存储器中。因为该密钥储存在设备上，因此它不是用于保持数据的机密性，而是可以根据需要快速抹掉（由用户使用“抹掉所有内容和设置”选项来抹掉，或者由用户或管理员通过从移动设备管理 (MDM) 服务器、Exchange ActiveSync 或 iCloud 发出远程擦除命令来抹掉）。以这种方式抹掉密钥将通过加密方式使设备上的所有文件不可访问。



文件的内容使用文件独有密钥进行加密，该密钥使用类密钥封装并储存在文件的元数据中，文件元数据又使用文件系统密钥进行加密。类密钥通过硬件 UID 获得保护，而某些类则通过用户密码获得保护。此层次结构既可提供灵活性，又可保证性能。例如，更改文件的类只需重新封装其文件独有密钥，更改密码只需重新封装类密钥。

密码注意事项

如果输入较长的纯数字密码，锁定屏幕上会显示数字小键盘，而非全键盘。与较短的字母数字密码相比，较长的数字密码更易于输入，而且可以提供类似的安全性。

密码

通过设置一个设备密码，用户便自动启用了数据保护。iOS 支持六位数、四位数和任意长度的字母数字密码。除了用于给设备解锁，密码还为特定的加密密钥提供熵。这意味着攻击者即使拿到设备，在没有密码的情况下也无法访问某些特定保护类的数据。

密码与设备的 UID 配合使用，因此暴力尝试只能是在受到攻击的设备上进行的。为此，iOS 系统使用较大的迭代次数来延缓每次尝试。迭代次数已经校准过以使每次尝试约耗时 80 毫秒。这意味着尝试 6 位字符的字母（小写）数字密码的全部组合将耗时 5 年半时间。

用户密码的强度越大，加密密钥强度就越高。Touch ID 可用于增强这样的因果关系，因为它可以让用户创建一个要比实用密码安全性高很多的密码。这增加了对用于数据保护的加密密钥进行保护的密码强度，而且不会对一天中多次解锁 iOS 设备的用户体验产生负面影响。

密码尝试间的延迟

尝试	执行延迟
1-4	无
5	1 分钟
6	5 分钟
7-8	15 分钟
9	1 小时

为了进一步阻止暴力破解，在锁定屏幕上输入无效密码后的延迟时间会逐步增加。如果“设置”>“Touch ID 与密码”>“抹掉数据”已打开，则当连续 10 次尝试输入错误的密码后，设备会自动擦除。此设置还可作为管理策略通过移动设备管理 (MDM) 和 Exchange ActiveSync 提供，而且可设置为较低的阈值。

在配备 A7 或更高版本 A 系列处理器的设备上，延迟由 Secure Enclave 执行。如果设备在定时延迟期间重新启动，延迟仍然执行，且定时器从当期重新计时。

数据保护类

在 iOS 设备上创建新文件时，创建它的应用会为其分配一个类。每个类使用不同的策略来确定数据何时可被访问。基本类和策略会在以下部分进行描述。

全面保护

(`NSFileProtectionComplete`)：该类密钥通过从用户密码和设备 UID 派生的密钥得到保护。用户锁定设备不久后（如果“需要密码”设置为“立即”，则为 10 秒钟后），已解密的类密钥会被丢弃，使得此类中的所有数据只有在用户再次输入密码或使用 Touch ID 解锁设备时才可被访问。

未打开文件的保护

(`NSFileProtectionCompleteUnlessOpen`)：某些文件可能需要在设备锁定时写入。如邮件附件在后台下载。此行为是通过使用非对称椭圆曲线加密技术（基于 Curve25519 的 ECDH）实现的。普通的文件独有密钥是通过使用一次性迪菲－赫尔曼密钥交换协议（One-Pass Diffie-Hellman Key Agreement，如 NIST SP 800-56A 中所述）派生的密钥进行保护。

该协议的临时公钥与封装的文件独有密钥一起储存。KDF 是串联密钥导出函数 (Approved Alternative 1)，如 NIST SP 800-56A 中 5.8.1 所述。AlgorithmID 已忽略。PartyUInfo 和 PartyVInfo 是独立的临时静态公钥。SHA-256 被用作哈希函数。一旦文件关闭，文件独有密钥会从内存中擦除。要再次打开该文件，系统会使用“未打开文件的保护”类的私钥和文件的临时公钥重新创建共享密钥；其哈希值被用来解开文件独有密钥的封装，然后用文件独有密钥来解密文件。

首次用户认证前保护

(`NSFileProtectionCompleteUntilFirstUserAuthentication`)：此类和“全面保护”类的行为方式相同，只不过锁定设备时已解密的类密钥不会从内存中删除。此类中的保护与桌面电脑全宗卷加密有类似的属性，可防止数据受到涉及重新启动的攻击。这是未分配给数据保护类的所有第三方应用数据的默认类。

无保护

(`NSFileProtectionNone`)：此类密钥仅受 UID 的保护，并且存储在可擦除存储器中。由于解密该类中的文件所需的所有密钥都储存在设备上，因此采用该类加密的唯一好处就是可以进行快速远程擦除。即使未向文件分配数据保护类，此文件仍会以加密形式储存（就像 iOS 设备上的所有数据那样）。

钥匙串数据保护

许多应用都需要处理密码和其他一些简短但比较敏感的数据，如密钥和登录令牌。iOS 钥匙串提供了储存这些项的安全方式。

钥匙串以储存在文件系统中的 SQLite 数据库的形式实现，且数据库只有一个；securityd 监控程序决定哪些钥匙串项可被每个进程或应用所访问。钥匙串访问 API 将生成对监控程序的调用，从而查询应用的“keychain-access-groups”、“application-identifier”和“application-group”权限。访问组允许在应用之间共享钥匙串项，而非将访问权限限制于单个进程。

钥匙串项只能在来自同一开发者的应用之间共享。管理它的方法是要求第三方应用使用访问组，并采用经由应用程序组通过 iOS Developer Program（iOS 开发者计划）为其分配的前缀。对前缀的要求和应用程序组唯一性通过代码签名、预置描述文件和 iOS Developer Program（iOS 开发者计划）强制实施。

系统用于保护钥匙串数据的类结构与文件数据保护中使用的类结构相似。这些类具有与文件数据保护类相同的行为，但使用的密钥不同，所属 API 的名称也不同。

可用性	文件数据保护	钥匙串数据保护
未锁定状态下	NSFileProtectionComplete	kSecAttrAccessibleWhenUnlocked
锁定状态下	NSFileProtectionCompleteUnlessOpen	暂无
首次解锁后	NSFileProtectionCompleteUntilFirstUserAuthentication	kSecAttrAccessibleAfterFirstUnlock
始终	NSFileProtectionNone	kSecAttrAccessibleAlways
密码启用状态下	暂无	kSecAttrAccessible-WhenPasscodeSetThisDeviceOnly

利用后台刷新服务的应用可将 `kSecAttrAccessibleAfterFirstUnlock` 用于后台更新过程中需要访问的钥匙串项。

类 `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly` 与 kSecAttrAccessibleWhenUnlocked 行为方式相同，但它仅当设备配置了密码时可用。此类只在系统密钥包中存在；它们不会同步到 iCloud 钥匙串；不会进行备份；也不会包括在托管密钥包中。如果密码被移除或重设，类密钥便会丢弃，这些项目也变得无法使用。

其他钥匙串类都有对应的“仅限本设备”项目，后者在备份期间从设备拷贝时始终受 UID 保护，因此如果恢复至其他设备将无法使用。

Apple 根据所保护信息的类型和 iOS 需要这些信息的时间来选择钥匙串类，妥善平衡了安全性和可用性。例如，VPN 证书必须始终可用，这样设备才能保持持续的连接，但它归类为“不可迁移”，因此不能将其移至另一台设备。

钥匙串项的组件

除了访问组，每个钥匙串项还包含管理元数据（如“创建时间”和“上次更新时间”时间戳）。

它还包含一些属性的 SHA-1 哈希值，用来查询某些项目（例如帐户和服务器名称）以允许无需解密每个项目即可进行查找。最后，它还包含加密数据，其中包括：

- 版本号
- 访问控制列表 (ACL) 数据
- 指明项目所属保护类的值
- 使用保护类密钥封装的项目独有密钥
- 描述项目的属性字典（传递到 SecItemAdd），编码为二进制 plist 并使用项目独有密钥加密

加密为 GCM（Galois/Counter Mode，伽罗瓦/计数器模式）模式下的 AES 128；访问组包含在属性中，受加密过程中计算的 GMAC 标签保护。

对于由 iOS 创建的钥匙串项，将强制实施以下类保护：

项目	可访问
无线局域网密码	首次解锁后
邮件帐户	首次解锁后
Exchange 帐户	首次解锁后
VPN 密码	首次解锁后
LDAP、CalDAV、CardDAV	首次解锁后
社交网络帐户令牌	首次解锁后
Handoff 广播加密密钥	首次解锁后
iCloud 令牌	首次解锁后
家庭共享密码	未锁定状态下
“查找我的 iPhone”令牌	始终
iTunes 备份	未锁定状态下，不可迁移
Safari 密码	未锁定状态下
Safari 书签	未锁定状态下
VPN 证书	始终，不可迁移
Bluetooth® 密钥	始终，不可迁移
Apple 推送通知服务令牌	始终，不可迁移
iCloud 证书和私钥	始终，不可迁移
iMessage 密钥	始终，不可迁移
由配置描述文件安装的证书和私钥	始终，不可迁移
SIM PIN 码	始终，不可迁移

钥匙串访问控制

钥匙串可以使用访问控制列表 (ACL) 以设定可访问性和认证要求的策略。钥匙串项可以设立条件，要求用户指定使用 Touch ID 或输入设备密码进行认证，否则不能访问。ACL 在 Secure Enclave 内部进行评估，只有符合其指定的限制条件时，才会释放到内核。

访问 Safari 已存储的密码

iOS 应用可以与 Safari 存储的钥匙串项交互，通过使用以下两个 API 进行密码自动填充：

- `SecRequestSharedWebCredential`
- `SecAddSharedWebCredential`

只有应用开发者和网站管理员同时批准且用户同意后，才会授予钥匙串访问权限。应用开发者通过在其应用中加入授权，让系统获知他们需要访问 Safari 已存储的密码。授权书中列出了相关网站的完全限定域名。网站必须将文件放在其服务器上，并在其中列出已批准应用的唯一应用标识符。安装了带有 com.apple.developer.associated-domains 授权的应用后，iOS 向列表中的每个网站发出 TLS 请求来请求文件 /apple-app-site-association。如果文件列出被安装应用的应用标识符，则 iOS 会将网站和应用标记为具有信任关系。只有具有信任关系才会调用这两个 API 并向用户发出提示，用户同意之后，密码才能发放给应用，或者被更新或删除。

密钥包

文件和钥匙串数据保护类的密钥收集在密钥包中，通过密钥包管理。iOS 使用以下四种密钥包：系统、备份、托管和 iCloud 云备份。

系统密钥包是设备常规操作中使用的封装类密钥的储存位置。例如，输入密码后，`NSFileProtectionComplete` 密钥会从系统密钥包中加载并解封。它是储存在“无保护”类中的二进制 plist，但其内容使用可擦除存储器中储存的密钥进行加密。为了给密钥包提供更高的安全性，用户每次更改密码时，系统都会擦除并重新生成此密钥。AppleKeyStore 内核扩展管理系统密钥包，并且可用于查询设备的锁定状态。仅当系统密钥包中的所有类密钥均可访问且成功解封时，它才会报告设备已解锁。

备份密钥包在 iTunes 进行加密备份时创建，它储存在设备进行备份的电脑中。新密钥包是通过一组新密钥创建的，备份的数据会使用这些新密钥重新加密。如前所述，不可迁移钥匙串项仍使用 UID 派生密钥封装，以使其可以恢复到最初备份它们的设备，但在其他设备上不可访问。

密钥包通过在 iTunes 中设置且运行了 10,000 次 PBKDF2 迭代的密码进行保护。虽然迭代次数非常多，但是密钥包并未捆绑特定设备，因此，理论上可尝试在多台电脑上对备份密钥包并行展开暴力破解。而安全性足够高的密码可以减小这一威胁。

如果用户选择不加密 iTunes 备份，那么不管备份文件属于哪一种数据保护类，备份文件都不加密，但钥匙串仍使用 UID 派生密钥获得保护。这就是只有设置备份密码才能将钥匙串项迁移到新设备的原因。

托管密钥包用于 iTunes 同步和 MDM。此密钥包允许 iTunes 执行备份和同步，而无需用户输入密码，它还允许 MDM 服务器远程清除用户密码。它储存在用于与 iTunes 进行同步的电脑，或者管理设备的 MDM 服务器上。

托管密钥包改善设备同步过程中的用户体验，期间可能需要访问所有类别的数据。当使用密码锁定的设备首次连接到 iTunes 时，会提示用户输入密码。然后设备创建托管密钥包，其中包含的类密钥与设备上使用的完全相同，该密钥包由新生成的密钥进行保护。托管密钥包及用于保护它的密钥划分到设备和主机或服务器上，其数据以“首次用户认证前保护”类储存在设备上。这就是重新启动后，用户首次使用 iTunes 进行备份之前必须输入设备密码的原因。

在 OTA 软件更新的情况下，开始更新时系统会提示用户输入密码。这被用来安全地创建一个“一次性解锁令牌”，该令牌在更新后解锁系统密钥包。此令牌只有在输入用户密码后才能生成，且如果更改了用户密码，则任何此前生成的令牌均无效。

“一次性解锁令牌”用于有人值守式或无人值守式的软件更新安装。Secure Enclave 中单调计数器的当前值、密钥包的 UUID 和 Secure Enclave 的 UID 会生成一个密钥，用来加密“一次性解锁令牌”。

SEP 中“一次性解锁令牌”计数器的增值会使任何现有令牌无效。计数器会在以下情况下增值：令牌使用时、重启设备首次解锁后、（用户或系统）取消软件更新时或令牌的策略计时器过期时。

对于有人值守式的软件更新，“一次性解锁令牌”会在 20 分钟后过期。此令牌从 Secure Enclave 导出，并被写入可擦除存储器。如果设备未在 20 分钟内重新启动，策略定时器将会使计数器增值。

对于无人值守式软件更新（设定方法：用户在收到更新通知时，选取“稍后安装”），应用程序处理器可保持“一次性解锁令牌”在 Secure Enclave 中有效长达 8 小时。之后，策略定时器会使计数器增值。

iCloud 云备份密钥包与备份密钥包类似。该密钥包中的所有类密钥都是非对称的（与“未打开文件的保护”数据保护类一样，使用 Curve25519），因此可以在后台执行 iCloud 云备份。对于除“无保护”类之外的所有数据保护类，加密的数据从设备中读取并发送到 iCloud。对应的类密钥通过 iCloud 密钥进行保护。钥匙串类密钥使用 UID 派生密钥进行封装，方式与未加密的 iTunes 备份相同。非对称密钥包还可用于“iCloud 钥匙串”钥匙串恢复中的备份。

安全性认证和计划

加密验证 (FIPS 140-2)

对于自 iOS 6 起发布的每个版本，iOS 中的加密模块均已进行符合美国联邦信息处理标准 (FIPS) 140-2 级别 1 的验证。iOS 9 中的加密模块与 iOS 8 中完全相同，但对于发布的每个版本，Apple 都会提交模块进行重新验证。这一计划证明，在正确使用 iOS 加密服务的 Apple 应用和第三方应用中，加密操作具有完整性。

通用标准认证 (ISO 15408)

Apple 已经开始进行通用标准认证 (CCC) 计划下的 iOS 认证。当前正在进行《Mobile Device Fundamental Protection Profile v2.0》(MDFPP2) 和《VPN IPSecPP1.4 Client Protection Profile》(VPNIPSecPP1.4) 这两项认证。在国际技术社区 (ITC) 中，Apple 积极参与开发当前还不可用的聚焦于关键移动安全性技术评估的保护描述文件 (PPs)。且 Apple 继续评估和开展针对当前可用的 PP 新版本和更新版本的认证。

涉密项目商业解决方案 (CSfC)

适用情况下，Apple 还提交了 iOS 平台和各种服务以包括到涉密项目商业解决方案 (CSfC) 计划组件列表中。具体来说，即针对移动平台的 iOS 和针对 IPSec VPN 客户端的 IKEv2 客户端（仅限 IKEv2“始终打开 VPN”）。由于 Apple 平台和服务正在进行通用标准认证，它们也将提交以包括到 CSfC 计划组件列表中。

安全性配置指南

Apple 一直与全世界各地政府协作开发相应指南，为维护更加安全的环境（也被称为“设备强化”）提供指导和建议。针对如何配置和利用 iOS 功能来加强保护，这些指南提供了经过核实的明确信息。

有关 iOS 安全性认证、验证和指导的信息，请参阅 support.apple.com/kb/HT202739?viewlocale=zh_CN。

应用安全性

应用是现代移动安全架构最关键的要素之一。尽管应用可显著提高用户的工作效率，但如果处理不当，也可能对系统安全性、稳定性和用户数据产生负面影响。

有鉴于此，iOS 提供了多重保护来确保应用经过签名和验证且经过沙盒化处理，从而始终保护用户数据。这些要素为应用提供了安全稳定的平台，使成千上万的开发者能够在 iOS 上提供数十万款应用，而不会影响系统完整性。用户可以在其 iOS 设备上访问这些应用，而不必过分担心病毒、恶意软件或未经授权的攻击。

应用代码签名

iOS 内核启动后，它将控制哪些用户进程和应用可以运行。为确保所有应用均来自批准的已知来源并且未被篡改，iOS 要求所有可执行代码均使用 Apple 颁发的证书进行签名。设备附带的应用（如“邮件”和 Safari）由 Apple 签名。第三方应用也必须使用 Apple 颁发的证书进行验证和签名。强制性代码签名将信任链的概念从操作系统扩展至应用，可防止第三方应用加载未签名的代码资源，或使用自修改代码。

要在 iOS 设备上开发并安装应用，开发者必须向 Apple 注册并加入 iOS Developer Program（iOS 开发者计划）。Apple 首先验证每个开发者（无论是个人还是企业）的真实身份，然后再颁发证书。开发者可使用该证书对应用进行签名，并将其提交至 App Store 进行分发。因此，App Store 中的所有应用都是由身份可识别的个人或组织提交的，由此防止恶意应用的创建。此外，这些应用都经过 Apple 的严格审核，以确保它们可以按照所述的方式运行，并且没有明显的错误或其他问题。除了已经讨论过的技术，这一精选过程还会让顾客对所购应用的品质更加放心。

iOS 允许开发者将框架嵌入应用中，使它可被应用本身使用，也可被应用中嵌入的扩展项使用。为保护系统并防止其他应用在其地址空间中加载第三方代码，系统将为启动时所有链接到该进程的动态资源库执行代码签名验证。此验证过程通过团队标识符 (Team ID) 完成。团队标识符来自 Apple 颁发的证书，是由10 个字符组成的字母数字串，例如 1A2B3C4D5F。程序可通过链接到随系统发布的任何资源库平台或其代码签名中具有同一团队标识符的资源库平台来成为主可执行程序。因为作为系统一部分发布的可执行程序不具有团队标识符，所以它们只能链接到随系统本身发布的资源库。

企业也可以编写供组织内部使用的企业内部应用，并将其分发给员工。企业和组织可以使用 D-U-N-S 编号申请加入 Apple Developer Enterprise Program（ADEP，Apple 开发者企业计划）。Apple 会在验证申请者的身份和资格后批准其请求。一旦组织成为 ADEP 的成员，便可以注册以获得一个预置描述文件，该描述文件允许企业内部应用在其授权的设备上运行。用户必须安装预置描述文件才能运行企业内部应用。这可以确保只有组织的目标用户能够将应用加载到其 iOS 设备上。通过 MDM 安装的应用为隐式受信任应用，因为组织与设备之间的关系已经确立。否则，用户必须在“设置”中批准应用的预置描述文件。组织可以限制用户批准来自未知开发者的应用。任何企业级应用首次启动时，设备必须收到 Apple 的肯定询证，表明允许该应用运行。

与其他移动平台不同，iOS 不允许用户安装来自网站的潜在恶意未签名应用或者运行不受信任的代码。运行时，会在加载所有可执行内存页后对其进行代码签名检查，以确保应用自安装或上次更新之后未被修改过。

运行时进程安全性

确认应用来自批准的来源后，iOS 会强制实施相应的安全措施，以防止其危害其他应用或系统的其余部分。

所有第三方应用均已经过“沙盒化”，因此在访问其他应用储存的文件或对设备进行更改时会受到限制。这样可以防止应用收集或修改其他应用储存的信息。每个应用还拥有唯一的主目录来存放其文件，主目录是在安装应用时随机分配的。如果第三方应用需要访问除自身信息以外的其他信息，只能通过 iOS 明确提供的服务来实现。

系统文件和资源也会与用户的应用保持隔离。与所有第三方应用一样，iOS 的绝大部分应用以非特权用户“mobile”的身份运行。整个操作系统分区以只读方式装载。不必要的工具（如远程登录服务）未包含在系统软件中，并且 API 不允许应用提升自己的特权来修改其他应用或 iOS 本身。

iOS 使用声明的授权来控制第三方应用对用户信息及功能（如 iCloud 和扩展功能）的访问。授权是签名到应用的密钥值对，允许对运行时因素之外的内容（如 unix 用户 ID）进行认证。授权已经过数码签名，因此无法更改。系统应用和监控程序广泛应用授权来进行特定特权操作，如果不使用授权，则需要以 root 用户身份运行进程。这极大降低了遭入侵的系统应用程序或监控程序提升特权的可能性。

此外，应用只能通过系统提供的 API 来执行后台处理。这就使应用能够继续运行，而不会降低性能或显著影响电池续航能力。

地址空间布局随机化 (ASLR) 可防止对内存损坏缺陷的利用。内置应用使用 ASLR 确保启动时随机安排所有内存区域。通过随机安排可执行代码、系统库和相关编程结构的内存地址，降低了遭到许多复杂攻击的可能性。例如，“return-to-libc”攻击试图通过操纵堆栈和系统库的内存地址来诱使设备执行恶意代码。随机安排内存地址大大增加了执行攻击的难度，尤其是对多个设备的攻击。作为 iOS 开发环境，Xcode 可自动编译启用了 ASLR 支持的第三方程序。

iOS 使用 ARM 的 Execute Never (XN) 功能提供进一步的保护，该功能会将内存页标记为不可执行。只有处于严格的控制条件下，应用才能使用标记为可写入和可执行的内存页：内核会检查 Apple 专有的动态代码签名授权是否存在。即使如此，也只有单个 mmap 调用能用于请求一个可执行且可写入的内存页，系统为可执行且可写入的内存页分配了随机地址。Safari 对其 JavaScript JIT 编译器使用了此功能。

扩展项

iOS 允许应用提供扩展项来增加其他应用的功能。扩展项是具有特定用途的已签名可执行二进制代码，封装在应用中。系统会在安装时自动检测扩展项，并允许与扩展项系统匹配的其他应用使用这些扩展项。

支持扩展项的系统区域称为扩展点。每个扩展点都提供 API，并为该区域强制执行策略。系统基于扩展点特定的匹配规则来决定哪些扩展项可用。系统自动按需启动扩展进程，并管理它们的生命周期。通过使用授权来限制特定系统应用程序的扩展可用性。例如，“今天”视图 Widget 只显示在“通知中心”中，而共享扩展项只在“共享”面板中可用。扩展点有“今天”视图 Widget、共享、自定操作、照片编辑、文稿提供程序和自定键盘。

扩展项在其自己的地址空间运行。应用与其激活的扩展项之间的通信采用由系统框架协调的进程间通信。它们无权访问彼此的文件或内存空间。扩展项的设计旨在将它们彼此隔离、与其包含的应用隔离，并且与使用它们的应用隔离。与其他第三方应用类似，它们也经过沙盒化并且拥有的容器与含有应用的容器隔开。但是扩展项与其容器应用对隐私控制具有相同的访问权限。因此，如果用户给应用授予“通讯录”的访问权限，该应用中嵌入的扩展项也会获得此权限，但由应用激活的扩展项不具有该权限。

自定键盘是一种特殊的扩展项类型，因为它由用户启用并适用于整个系统。一旦启用，该扩展项会用于所有的文本栏，除了密码输入栏和任何安全文本视图。出于隐私保护的原因，默认情况下自定键盘运行在一个非常受限的沙盒中，该沙盒阻止网络访问、阻止代表进程执行网络操作的服务，并阻止可允许扩展项偷偷键入数据的 API。自定键盘的开发者可以要求其扩展项拥有“开放存取”权限，使系统在得到用户的同意后在默认的沙盒中运行扩展项。

对于在移动设备管理中注册的设备，文稿和键盘扩展项将遵循“被管理的打开方式”规则。例如，MDM 服务器可阻止用户将被管理的应用中的文稿导出到未被管理的文稿提供程序，或阻止他们在被管理的应用中使用未被管理的键盘。另外，应用开发者可阻止在其应用中使用第三方键盘扩展项。

应用组

指定开发者帐户拥有的应用和扩展项在配置为应用组的一部分后可共享内容。由开发者决定是否在 Apple Developer Portal（Apple 开发者门户）上创建合适的群组并包括所需的一套应用和扩展项。应用配置为应用组的一部分后，可访问以下内容：

- 磁盘上共享的存储容器，只要有应用组内的一个应用被安装，它就会一直保留在设备上
- 共享的偏好设置
- 共享的钥匙串项

Apple Developer Portal（Apple 开发者门户）保证了应用组 ID 在整个应用生态系统的唯一性。

应用中的数据保护

iOS 软件开发套件 (SDK) 提供全套 API，使第三方和企业内部开发者能够轻松地利用数据保护类，帮助确保在应用中实现最高级别的保护。数据保护适用于文件和数据库 API，包括 NSFileManager、CoreData、NSData 和 SQLite。

“邮件”应用（包括附件）、被管理的图书、Safari 书签、应用启动图像和位置数据也将加密储存，加密密钥通过用户设备上的密码进行保护。“日历”（不包括附件）、“通讯录”、“提醒事项”、“备忘录”、“信息”和“照片”采用“首次用户认证前保护”。

没有选择加入某个特定数据保护类且由用户安装的应用默认接受“首次用户认证前保护”。

配件

“Made for iPhone, iPod touch, and iPad (MFi)”许可计划允许已审查的配件制造商对 iPod 配件协议 (iAP) 和必要的支持硬件组件进行访问。

当 MFi 配件使用 Lightning 接口或通过蓝牙与 iOS 设备通信时，设备要求配件使用 Apple 提供的证书（设备对此证书进行验证）进行回应，以证明配件经过 Apple 授权。然后，设备发送一个质询，配件必须使用已签名的证书来响应。这个过程完全由定制集成电路来处理，而且对于配件本身是透明的。执行处理操作的定制集成电路由 Apple 提供给许可配件制造商。

配件可以请求访问不同的传输方法和功能；例如，通过 Lightning 线缆访问数字音频流或通过蓝牙提供位置信息。认证集成电路确保只有经过批准的设备才能获得对设备的完全访问权限。如果配件不提供认证，其访问仅限于模拟音频和一小部分串行 (UART) 音频播放控件。

AirPlay 还利用认证集成电路来验证接收器已由 Apple 批准。AirPlay 音频流和 CarPlay 视频流使用 MFi-SAP（安全关联协议），此协议使用 AES-128 在 CTR 模式下对配件和设备之间的通信进行加密。临时密钥使用 ECDH 密钥交换 (Curve25519) 进行交换，并使用认证电路的 1024 位 RSA 密钥进行签名以作为端到端 (STS) 协议的一部分。

HomeKit

HomeKit 奠定了家庭自动化的基础，通过利用 iCloud 和 iOS 安全性来保护和同步专用数据，而无需将这些数据透露给 Apple。

HomeKit 身份标识

HomeKit 身份标识和安全性基于 Ed25519 公-私密钥对。iOS 设备上会为每位用户针对 HomeKit 生成 Ed25519 密钥对，即用户的 HomeKit 身份标识。该密钥对被用来认证 iOS 设备之间以及 iOS 设备和配件之间的通信。

密钥储存于钥匙串内并只包括在加密的钥匙串备份中，而且会在使用 iCloud 钥匙串的设备间同步。

与 HomeKit 配件通信

HomeKit 配件会生成自己的 Ed25519 密钥对，用来与 iOS 设备进行通信。如果将配件恢复为出厂设置，则会生成新的密钥对。

为了在 iOS 设备和 HomeKit 配件之间建立联系，会使用安全远程密码（3072 位）协议来交换密钥：用户在 iOS 设备上输入由配件生产商提供的 8 位数代码，然后使用 HKDF-SHA-512 派生密钥按照 ChaCha20-Poly1305 AEAD 进行加密。在设置过程中，还会对配件的 MFi 认证进行验证。

当 iOS 设备和 HomeKit 配件在使用过程中进行通信时，它们采用上述过程中交换的密钥相互进行认证。每个会话均使用端到端协议建立，并基于每会话 Curve25519 密钥通过 HKDF-SHA-512 派生密钥进行加密。这同时适用于基于 IP 的配件和低功耗蓝牙配件。

本地数据储存

HomeKit 将有关家庭、配件、场景和用户的数据储存在用户的 iOS 设备上。储存的数据会使用派生自用户 HomeKit 身份标识密钥的密钥和随机数进行加密。此外，HomeKit 数据还会使用“首次用户认证前保护”数据保护类进行储存。HomeKit 数据仅备份在加密的备份中，因此诸如未加密的 iTunes 备份就不包含 HomeKit 数据。

设备和用户间的数据同步

HomeKit 数据可在用户使用 iCloud 和 iCloud 钥匙串的 iOS 设备间同步。同步期间，HomeKit 数据会使用派生自用户 HomeKit 身份的密钥和随机数进行加密同步时，此数据会作为不透明 blob 处理。最近处理的 blob 会储存在 iCloud 中以启用同步，并不会用作其他目的。因为 HomeKit 数据加密所使用的密钥仅在用户的 iOS 设备上可用，因此在传输和 iCloud 储存过程中无法对其内容进行访问。

HomeKit 数据还可在同一个家庭的多个用户间进行同步。这一过程所采用的认证和加密方法与 iOS 设备和 HomeKit 配件之间所使用的相同。当用户加入家庭时，设备间会交换 Ed25519 公钥进行认证。新用户加入家庭后，会使用端到端协议和每会话密钥来认证和加密任何进一步的通信。

只有最初在 HomeKit 创建家庭的用户能添加新用户。其设备会使用新用户的公钥来配置配件，这样配件就能认证并接受新用户的命令。配置 Apple TV 以配合 HomeKit 使用的这个过程采用与添加其他用户时相同的认证和加密方法，但如果创建家庭的用户在 Apple TV 上登录了 iCloud 且 Apple TV 位于家庭中，则会自动对 Apple TV 进行配置。

如果用户没有多台设备，且未批准其他用户访问家庭，那么没有 HomeKit 数据会同步到 iCloud。

家庭数据和应用

用户可通过“隐私”设置来控制应用对家庭数据的访问。当应用请求访问家庭数据（与请求访问“通讯录”、“照片”和其他 iOS 数据源类似）时，会要求用户授予访问权限。如果用户批准，应用可以访问房间名称、配件名称、每个配件所处的房间以及在 HomeKit 开发者文稿中详述的其他信息。

Siri

Siri 可用来询问和控制配件以及激活各种场景。Siri 只会匿名获得有关家庭配置的极少量信息（请参阅本白皮书的 Siri 部分）。所提供有关房间名称、配件和场景的信息为命令识别所需。

针对 HomeKit 配件的 iCloud 远程访问

HomeKit 配件可以直接连接 iCloud，使 iOS 设备能够在蓝牙或无线局域网通信不可用时控制配件。

iCloud 远程访问经过精心设计，使配件可被控制并发送通知，同时不向 Apple 透露是什么配件，或发送的是什么命令和通知。HomeKit 不会通过 iCloud 远程访问发送关于家庭的信息。

当用户使用 iCloud 远程访问发送命令时，配件和 iOS 设备相互认证，且数据使用针对本地连接的同样步骤进行加密。通信的内容经过加密，对 Apple 不可见。通过 iCloud 的寻址基于在设置过程中所注册的 iCloud 标识符。

支持 iCloud 远程访问的配件在设置配件的过程中预置。预置过程从用户登录到 iCloud 开始。接着，iOS 设备提示配件使用内建于所有“Built for HomeKit”配件的 Apple 认证协处理器来给质询签名。配件还会生成 prime256v1 椭圆曲线密钥，且公钥与签名的质询和认证协处理器的 X.509 证书一起被发送到 iOS 设备。这些用于为配件从 iCloud 预置服务器请求一个证书。该证书由配件储存，但不包含关于配件的任何识别信息（除了它已被授予 HomeKit iCloud 远程访问的权限）。正在进行预置的 iOS 设备还会向配件发送一个包，其中包含连接到 iCloud 远程访问服务器所需的 URL 及其他信息。此信息不指向任何用户或配件。

每个配件在 iCloud 远程访问服务器中会注册一个被许可用户列表。给家庭添加配件的人已授予这些用户控制配件的权限。iCloud 服务器授予用户一个标识符，且用户可以映射到 iCloud 帐户，以传送来自配件的通知信息和响应。同样，配件也有 iCloud 颁发的标识符，但是这些标识符经过了模糊化处理，不会透露关于配件本身的任何信息。

配件连接到 HomeKit iCloud 远程访问服务器时，会显示其证书和凭证。凭证是从不同的 iCloud 服务器获得，且每个配件的凭证并非唯一。当配件请求凭证时，请求中将包括其制造商、型号和固件版本。此请求中不会发送任何用户识别信息或家庭识别信息。为了帮助保护隐私，与凭证服务器的连接没有进行认证。

配件通过 HTTP/2 连接到 iCloud 远程访问服务器，使用 TLS 1.2 配合 AES-128-GCM 和 SHA-256 保障安全。配件将其与 iCloud 远程访问服务器的连接保持开放，这样配件就可接收传入信息并向 iOS 设备发送回应和通知。

HealthKit

HealthKit 框架提供了一个通用的数据库，用户可授权应用使用该数据库来储存和访问健身与健康数据。HealthKit 还直接与健康和健身设备打交道，例如兼容的低功耗蓝牙心率监视器，以及内建于许多 iOS 设备中的运动协处理器。

健康数据

HealthKit 使用数据库来储存用户的健康数据，例如身高、体重、步行距离、血压等信息。此数据库使用“全面保护”数据保护类储存，这意味着只有在用户输入密码或使用 Touch ID 解锁设备时才能进行访问。

另一个数据库储存操作数据，例如应用的访问表格、连接到 HealthKit 的设备名称，以及新数据可用时，用来开启应用的计划信息。此数据库使用“首次用户认证前保护”数据保护类储存。

临时日志文件储存设备锁定时（例如用户锻炼时）所生成的健康记录。这些文件使用“未打开文件的保护”数据保护类储存。设备解锁后，这些临时日志文件会导入进主要健康数据库，并会在合并完成后被删除。

健康数据不会通过 iCloud 共享，也不会在设备间同步。健康数据库包含在备份到 iCloud 或 iTunes 的加密设备备份中。健康数据不会包含在未加密的 iTunes 备份中。

数据完整性

储存在数据库中的数据包括用于追踪每条数据记录起源的元数据。该元数据包括应用程序标识符，用以识别储存了该记录的应用。此外，可选元数据项还可能包含记录的数码签名副本，从而保持受信设备生成记录的数据完整性。数码签名采用在 IETF RFC 5652 中描述的“密码讯息语法”(CMS) 格式。

第三方应用访问

应用只有通过授权才能访问 HealthKit API，且必须遵守数据使用方式的访问限制。例如，不允许应用使用健康数据进行广告。应用还需要向用户提供隐私政策，详细说明其如何使用健康数据。

用户可通过“隐私”设置来控制应用对健康数据的访问。当应用请求访问健康数据（与请求访问“通讯录”、“照片”和其他 iOS 数据源类似）时，会要求用户授予访问权限。但对于健康数据，应用将获得读取和写入数据的单独访问权限，以及对每种健康数据类型的单独访问权限。用户可以在“健康”应用的“数据来源”标签中，查看和撤销所授予的访问健康数据的权限。

如果获得写入数据的权限，应用还可以读取其写入的数据。如果获得读取数据的权限，则可以读取所有来源写入的数据。但应用不能决定其他应用所获得的访问权限。此外，应用不能完全确定其是否获得了读取健康数据的权限。如果应用没有读取权限，所有查询都不会返回数据，如同查询空数据库。这可以阻止应用通过学习用户所跟踪的数据类型来推断用户的健康状况。

医疗急救卡

“健康”应用可让用户在医疗急救卡表单中填写急救时可能至关重要的信息。这些信息可由用户输入或手动更新，且不会与健康数据库中的信息同步。

在锁定屏幕上轻点“紧急情况”按钮可查看医疗急救卡信息。该信息使用“无保护”数据保护类储存于设备上，因此无需输入设备密码就可进行访问。“医疗急救卡”作为一项可选功能，可让用户权衡安全性和隐私二者之间的关系。

Apple Watch

Apple Watch 使用 iOS 内置的安全性功能和技术来帮助保护设备上的数据，以及与所配对 iPhone 和互联网的通信。这包括使用诸如数据保护和钥匙串访问控制等技术。用户的密码还会与设备 UID 配合使用，从而创建加密密钥。

Apple Watch 和 iPhone 配对时，通过带外 (OOB) 处理交换公钥和 BTLE 链接共享密钥进行保护。Apple Watch 显示一幅动画图案供 iPhone 摄像头捕捉。该图案包含加密的密钥，用于 BTLE 4.1 带外配对。如果需要，Apple Watch 会使用标准 BTLE 万能钥匙进入模式作为备用配对方法。

一旦建立了 BTLE 会话，Apple Watch 和 iPhone 就会通过改进自 IDS 的流程（请参阅本白皮书的 iMessage 部分）交换密钥。密钥交换完成后，蓝牙会话密钥被丢弃，Apple Watch 和 iPhone 间的所有通信使用 IDS 加密，同时，加密的 BTLE 和无线局域网链接提供第二层加密。为避免流量被截获，每隔 15 分钟将使用滚动式密钥来限制曝光窗口。

为支持需要流化数据的应用，加密采用了本白皮书中 FaceTime 部分所描述的方法，即利用配对 iPhone 提供的 IDS 服务。

Apple Watch 对文件和钥匙串项采用硬件加密储存以及基于类的保护（请参阅本白皮书的“数据保护”部分）。同时对钥匙串项还使用了访问控制密钥包。对于手表和 iPhone 间通信使用的密钥，也采用了基于类的保护进行加密。

当 Apple Watch 不在蓝牙通信范围内时，可以转而使用无线局域网。配对 iPhone 自动向手表提供已知网络的列表，如果加入网络的凭证不在配对 iPhone 上，Apple Watch 便不会加入无线局域网络。

长按侧边按钮可手动锁定 Apple Watch。此外，如果将手表从手腕摘除，会立即使用运动启发技术来尝试自动锁定该设备。锁定后，Apple Pay 将无法使用。如果在设置中关闭手腕检测提供的自动锁定功能，Apple Pay 会被停用。用户可以在 iPhone 上的 Apple Watch 应用中关闭手腕检测，也可以使用移动设备管理来强制实施此设置。

佩戴着手表时，也可以使用配对 iPhone 来解锁手表。先通过配对期间生成的密钥来认证二者间建立的连接，然后手表再使用 iPhone 发送的密钥来解锁其数据保护密钥，从而实现解锁。手表密码不为 iPhone 所知，也不会被传输。此功能可以在 iPhone 上的 Apple Watch 应用中关闭。

Apple Watch 一次只能与一部 iPhone 配对。与新 iPhone 配对会自动抹掉 Apple Watch 中的所有内容和数据。

在配对的 iPhone 上启用“查找我的 iPhone”也会启用 Apple Watch 上的激活锁。激活锁使任何人都难以使用或出售丢失或被盗的 Apple Watch。激活锁需要用户的 Apple ID 和密码来取消配对、抹掉或重新激活 Apple Watch。

网络安全性

除了 Apple 用于保护 iOS 设备上所储存数据的内置安全保护，也有许多网络安全措施可供企业组织采用并确保信息在来往于 iOS 设备时安全无虞。

移动用户必须能在全球任何地方访问公司网络，因此很重要的一点是确保他们得到授权并且其数据在传输期间受到保护。iOS 使用标准联网协议并使开发者能够访问这些协议，以进行经过认证的已授权加密通信。为了实现这些安全目标，iOS 集成了经证实的技术和最新标准来进行无线局域网和蜂窝移动数据网络连接。

在其他平台上，需要用防火墙软件保护开放通信端口以防止入侵。由于 iOS 通过限制监听端口以及移除不必要的网络工具（如 telnet、shell 或 Web 服务器），使受攻击的范围减小，因此在 iOS 设备上不需要额外的防火墙软件。

TLS

iOS 支持传输层安全协议（TLS v1.0、TLS v1.1、TLS v1.2）和数据包传输层安全协议。Safari、“日历”、“邮件”和其他互联网应用自动使用这些机制在设备与网络服务之间建立一条加密的通信通道。上层 API（如 CFNetwork）使开发者可以轻松在其应用中采用 TLS，而底层 API (SecureTransport) 则提供精细控制。默认情况下，CFNetwork 不接受 SSLv3，且使用 WebKit（例如 Safari）的应用禁止建立 SSLv3 连接。

应用传输安全性

应用传输安全性提供默认的连接要求，这样应用在使用 NSURLConnection、CFURL 或 NSURLSession API 时会遵守安全连接的最佳实践。

服务器必须至少支持前向保密 TLS 1.2，且证书必须有效并使用 SHA-256 或更佳加密算法签名，且包含至少 2048 位 RSA 密钥或 256 位椭圆曲线密钥。

不满足这些要求的网络连接将会失败，除非应用重写了应用传输安全协议。无效的证书始终导致硬故障和无连接。应用传输安全性自动应用到针对 iOS 9 编译的应用。

VPN

类似于虚拟专用网的安全网络服务通常只需简单的设置和配置，便可配合 iOS 设备使用。与 iOS 设备配合使用的 VPN 服务器支持以下协议和认证方式：

- IKEv2/IPSec，此协议通过共享密钥、RSA 证书、ECDSA 证书、EAP-MSCHAPv2 或 EAP-TLS 认证。
- Pulse Secure、Cisco、Aruba Networks、SonicWALL、Check Point、Palo Alto Networks、Open VPN、AirWatch、MobileIron、NetMotion Wireless 以及 F5 Networks SSL-VPN，它们在 App Store 中提供相应的客户端应用。
- Cisco IPSec，此协议通过密码、RSA SecurID 或 CRYPTOCard 进行用户认证，并通过共享密钥和证书进行机器认证。
- L2TP/IPSec，此协议通过 MS-CHAPV2 密码、RSA SecurID 或 CRYPTOCard 进行用户认证，并通过共享密钥进行机器认证。
- PPTP，此协议通过 MS-CHAPV2 密码和 RSA SecurID 进行用户认证，支持但不建议通过 CRYPTOCard 进行用户认证。

对于使用基于证书认证的网络，iOS 支持“请求 VPN 域”。IT 策略通过使用配置描述文件来指定哪些域需要 VPN 连接。

iOS 还支持为应用单独设置 VPN 支持，帮助更精确地建立 VPN 连接。移动设备管理 (MDM) 可为每个被管理的应用和/或 Safari 中特定的域指定连接。这有助于确保进出公司网络的数据始终是安全的，而用户的个人数据不会进出公司网络。

iOS 支持“始终打开 VPN”，通过 MDM 管理的设备、使用 Apple Configurator 或 Device Enrollment Program（设备注册计划）监督的设备可进行该配置。这使得用户在连接到蜂窝移动网络和无线局域网络时不需要手动打开 VPN 以启用保护。“始终打开 VPN”通过将所有 IP 流量回传至组织，使得组织拥有设备流量的完整控制权。默认隧道协议 IKEv2 通过数据加密保护流量传输安全。现在，组织可以监控并过滤传输到和传输自其设备的流量、保护组织网络内的数据安全并限制设备访问互联网。

无线局域网

iOS 支持工业标准的无线局域网协议，包括“WPA2 企业级”，可针对公司无线网络提供访问认证服务。“WPA2 企业级”使用 128 位 AES 加密，可为用户提供最高级别的安全保障：在通过无线局域网络连接发送和接收通信时，确保用户的数据始终受到保护。由于支持 802.1X，iOS 设备可集成到各种 RADIUS 认证环境中。iPhone 和 iPad 上支持的 802.1X 无线认证方式包括 EAP-TLS、EAP-TTLS、EAP-FAST、EAP-SIM、PEAPv0、PEAPv1 和 LEAP。

当设备没有关联到无线局域网络且设备处理器处于睡眠状态时，iOS 会在引导 PNO (Preferred Network Offload) 扫描时使用随机的介质访问控制 (MAC) 地址。屏幕关闭后，设备的处理器会立即进入睡眠状态。运行 PNO 扫描是为了确定用户可否连接到首选无线局域网络，进行诸如与 iTunes 无线同步等活动。

当设备没有关联到无线局域网络或设备处理器处于睡眠状态时，iOS 还会在引导 ePNO (enhanced Preferred Network Offload) 扫描时使用随机 MAC 地址。如果设备中要使用地理围栏的应用（例如基于位置的提醒事项，确定设备是否接近某个特定位置）在使用“定位服务”，则会运行 ePNO 扫描。

现在由于设备未接入无线局域网络时其 MAC 地址会更改，因此即使设备连接到了蜂窝移动网络，无线局域网流量的被动观察程序不能使用该地址一直跟踪设备。

我们一直同无线局域网制造商保持合作，告诉他们后台扫描时使用随机 MAC 地址，且 Apple 和制造商都无法预测这些随机的 MAC 地址。

iPhone 4s 不支持无线局域网 MAC 地址随机化。

蓝牙

iOS 的蓝牙支持旨在提供实用的功能，而不会增加对专用数据不必要的访问。iOS 设备支持 Encryption Mode 3、Security Mode 4 和 Service Level 1 连接。iOS 支持以下蓝牙描述文件：

- 免提描述文件 (HFP 1.5)
- 电话簿访问描述文件 (PBAP)
- 高级音频分发描述文件 (A2DP)
- 音频/视频远程控制描述文件 (AVRCP)
- 个人区域网络描述文件 (PAN)
- 人机接口设备描述文件 (HID)

对这些描述文件的支持因设备而异。有关更多信息，请访问 support.apple.com/kb/ht3647?viewlocale=zh_CN。

单点登录

iOS 支持通过单点登录 (SSO) 对企业网络进行认证。SSO 与基于 Kerberos 的网络配合使用，针对用户有权访问的服务对用户进行认证。SSO 可用于各种网络活动，从安全的 Safari 会话到第三方应用。

iOS SSO 利用 SPNEGO 令牌和 HTTP Negotiate 协议，与基于 Kerberos 的认证网关和支持 Kerberos 票据的“集成 Windows 身份验证”系统 (Windows Integrated Authentication system) 系统配合使用，同时还支持基于证书的认证。SSO 支持基于开源 Heimdal 项目。

支持以下加密类型：

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

Safari 支持 SSO，而且使用标准 iOS 联网 API 的第三方应用可也可进行配置来使用它。为了配置 SSO，iOS 支持配置描述文件有效负载，允许 MDM 服务器向下推送必要的设置。其中包括：设置用户主体名称（即 Active Directory 用户帐户）和 Kerberos 领域设置，以及配置应允许哪些应用和/或 Safari Web URL 使用 SSO。

AirDrop 安全性

支持 AirDrop 的 iOS 设备使用低功耗蓝牙 (BLE) 和 Apple 创建的点对点无线局域网技术来向附近的设备发送文件和信息，包括具有 AirDrop 功能并运行 OS X Yosemite 或更高版本的 Mac 电脑。无线局域网信号用来在设备之间进行直接通信，无需使用任何互联网连接或无线局域网接入点。

用户启用 AirDrop 后，设备上就会储存一个 2048 位 RSA 身份标识。此外，设备还会根据与用户 Apple ID 相关联的电子邮件地址和电话号码，创建一个 AirDrop 身份标识哈希值。

当用户选择使用 AirDrop 共享项目时，设备会通过低功耗蓝牙发出 AirDrop 信号。附近处于唤醒状态且启用了 AirDrop 的其他设备检测到这一信号后，会使用其所有者的身份标识哈希值的精简版本进行响应。

默认情况下，AirDrop 的共享对象设置为“仅限联系人”。用户还可以选择是否希望使用 AirDrop 与所有人进行共享，或者完全关闭这一功能。在“仅限联系人”模式下，接收到的身份标识哈希值会与发起人“通讯录”应用中联系人的哈希值进行对比。如果发现匹配，发送设备会创建一个点对点无线局域网络并使用 Bonjour 告知已建立 AirDrop 连接。接收设备会使用这一连接将其完整身份标识哈希值发送给发起人。如果完整哈希值仍与“通讯录”匹配，接收者的名字和照片（如果“通讯录”中有）会显示在 AirDrop 共享表单中。

使用 AirDrop 时，由发送方用户选择要与其共享内容的对象。发送设备会与接收设备建立一个加密的 (TLS) 连接，此连接会交换他们的 iCloud 身份证书。身份证书中的身份标识会针对每位用户的“通讯录”应用进行验证。然后会请求接收方用户接收从经过验证的人或设备传输的内容。如果选择了多个接收者，将针对每个目标重复此过程。

在“所有人”模式中会采用同样的过程，但如果未能在“通讯录”中找到匹配项，接收设备会显示在 AirDrop 发送表单中，并带有一个小图标及设备名称，该名称可在“设置”>“通用”>“关于本机”>“名称”中找到。

针对通过移动设备管理解决方案管理的设备或应用，组织可以限制 AirDrop 的使用。

Apple Pay

通过 Apple Pay，用户可以使用受支持的 iOS 设备和 Apple Watch 以方便、安全和保密的方式进行付款。Apple Pay 操作简单，且在硬件和软件中都采用了集成安全技术。

Apple Pay 的设计还可以保护用户的个人信息。它不会收集可绑定到用户的任何交易信息。付款交易只在用户、商户和发卡机构之间发生。

Apple Pay 组件

安全元件 (Secure Element)：安全元件是业内公认、运行 Java Card 平台的认证芯片，它符合金融行业对于电子支付的要求。

NFC 控制器：NFC 控制器处理“近距离无线通信”协议，并发送应用程序处理器和安全元件之间以及安全元件和销售点终端之间的通信。

Wallet：Wallet 应用被用来添加和管理信用卡、借记卡、回馈卡和商店卡，并通过 Apple Pay 进行支付。用户可以在 Wallet 应用中查看其付款卡以及关于发卡机构的其他信息、发卡机构的隐私政策、最近的交易等内容。还可以在“设置助理”和“设置”中将付款卡添加到 Apple Pay。

Secure Enclave：在 iPhone 和 iPad 上，Secure Enclave 负责管理认证过程并让支付交易得以继续。它储存 Touch ID 的指纹数据。

在 Apple Watch 上，设备必须解锁且用户必须连按侧边按钮。检测到的连按操作会直接发送到安全元件，而不经过应用程序处理器。

Apple Pay 服务器：Apple Pay 服务器负责管理 Wallet 中信用卡和借记卡的状态，以及储存在安全元件中的“设备帐号”。它们同时与设备和支付网络服务器通信。Apple Pay 服务器还负责再次加密在应用内进行支付时的支付凭证。

Apple Pay 如何使用安全元件

安全元件包含专门设计用来管理 Apple Pay 的小程序，还包括由支付网络认证的支付小程序。加密的信用卡或借记卡数据会从支付网络或发卡机构发送到这些支付小程序，期间会使用仅为支付网络和支付小程序的安全域所知的密钥。此数据储存在这些支付小程序内，并使用安全元件的安全性功能进行保护。交易期间，终端使用专门的硬件总线通过“近距离无线通信”(NFC) 控制器直接与安全元件进行通信。

Apple Pay 如何使用 NFC 控制器

作为安全元件的入口，NFC 控制器确保所有非接触式支付交易都通过处于设备近距离范围内的销售点终端进行。NFC 控制器只会将来自场内终端的支付请求标记为非接触式交易。

一旦持卡人使用 Touch ID 或密码授权支付，或者在解锁的 Apple Watch 上通过连按侧边按钮来授权支付，控制器会将安全元件内支付小程序准备的非接触式响应专门发送给 NFC 场。因此，非接触式交易的支付授权详细信息会包含在本地 NFC 场中，绝不会透露给应用程序处理器。相比之下，在应用内进行支付时，支付授权详细信息会被发送到应用程序处理器，但只有在安全元件加密后才会发送给 Apple Pay 服务器。

信用卡和借记卡预置

当用户将信用卡或借记卡（包括商店卡）添加到 Apple Pay 时，Apple 会安全地将付款卡信息以及关于用户帐户和设备的其他信息，发送给发卡机构。发卡机构将使用此信息，决定是否批准将付款卡添加到 Apple Pay。

Apple Pay 使用三个服务器端调用命令来发送和接收与发卡机构或网络间的通信，以作为付款卡预置过程的一部分：**必填字段**、**核对付款卡**以及**链接和预置**。发卡机构或网络使用这些调用命令来验证、批准付款卡并将其添加到 Apple Pay。这些客户端服务器会话使用 SSL 加密。

完整的付款卡号码不会储存在设备或 Apple 服务器上。相反，会创建唯一的“设备帐号”、进行加密，然后储存在安全元件中。此唯一的“设备帐号”采用 Apple 无法访问的方式加密。“设备帐号”是唯一的，与通常的信用卡或借记卡号码不同。发卡机构可以阻止在磁条卡、电话或网站上使用“设备帐号”。安全元件中的“设备帐号”与 iOS 和 WatchOS 是分开的，永不会储存在 Apple Pay 服务器上或备份到 iCloud。

在 iPhone 上的 Apple Watch 应用中，用户可以为 Apple Pay 预置配合 Apple Watch 使用的付款卡。为 Apple Watch 预置付款卡时，要求手表位于蓝牙通信范围内。配合 Apple Watch 使用的付款卡会进行特别注册，拥有自己的设备帐号且储存在 Apple Watch 上的安全元件内。

可通过以下两种方式将信用卡或借记卡预置到 Apple Pay 中：

- 手动将信用卡或借记卡添加到 Apple Pay
- 将存档的信用卡或借记卡从 iTunes Store 帐户添加到 Apple Pay

手动将信用卡或借记卡添加到 Apple Pay

要手动添加付款卡（包括商店卡），需要使用姓名、信用卡号码、过期日期和 CVV 码来辅助预置过程。用户可以在“设置”、Wallet 应用或 Apple Watch 应用中键入或使用 iSight 摄像头来输入该信息。摄像头捕获到付款卡信息后，Apple 会尝试填充姓名、卡号和过期日期。所拍摄的照片永不会存储到设备或储存在照片图库中。在填写好所有栏位后，“核对付款卡”流程会验证 CVV 码以外的栏位。这些信息会通过加密方式发送到 Apple Pay 服务器。

如果“核对付款卡”流程返回条款与条件 ID，Apple 会下载发卡机构的条款与条件并向用户显示。如果用户接受该条款与条件，Apple 会将所接受条款的 ID 以及 CVV 码发送到“链接和预置”流程。此外，作为“链接和预置”流程的一部分，Apple 会与发卡机构或网络共享设备中的信息，比如有关您 iTunes Store 和 App Store 帐户活动的信息（例如，在 iTunes 中是否有长期的交易历史记录），有关您设备的信息（例如，电话号码、姓名、设备型号以及设置 Apple Pay 所需的任何配套 iOS 设备），以及添加付款卡时您大致的位置（如果启用了“定位服务”）。发卡机构将使用此信息，决定是否批准将付款卡添加到 Apple Pay。

“链接和预置”流程会执行以下两项操作：

- 设备开始下载代表信用卡或借记卡的 Wallet 凭证文件。
- 设备开始将付款卡与安全元件绑定。

凭证文件包含用来下载付款卡插图的 URL，有关付款卡的元数据，例如联系信息、相关的发卡机构应用以及支持的功能。它还包括凭证状态：例如安全元件是否完成了个性化设置、付款卡当前是否被发卡机构暂停使用或者在付款卡能够使用 Apple Pay 进行支付前是否需要进行额外验证。

将信用卡或借记卡从 iTunes Store 帐户添加到 Apple Pay

对于 iTunes 存档的信用卡或借记卡，可能需要用户重新输入 Apple ID 密码。然后从 iTunes 取回卡号，并启动“核对付款卡”流程。如果付款卡符合 Apple Pay 的条件，设备将会下载并显示条款与条件，然后将其与条款 ID 和付款卡安全码一起发送到“链接和预置”流程。对于存档的 iTunes 帐户付款卡，可能会需要进行额外验证。

从发卡机构的应用添加信用卡或借记卡

当应用注册以使用 Apple Pay 时，将为应用和商户的服务器建立密钥。这些密钥用于加密发送到商户的付款卡信息，从而阻止信息被 iOS 设备读取。预置流程与上述手动添加付款卡时类似，只有一点不同，即用一次性密码代替 CVV 码。

额外验证

发卡机构可以决定是否需要对信用卡或借记卡进行额外验证。根据发卡机构提供的功能，用户可能有以下选择进行额外验证：短信提醒、电子邮件通知、客服电话通知或者通过认证的第三方应用来完成验证。用户可以选择发卡机构存档的联系信息来获取短信或电子邮件通知，并在 Wallet、“设置”或 Apple Watch 应用中输入收到的代码。对于客服通知或使用应用验证的方式，发卡机构有其自己的通信流程。

支付授权

安全元件只有在接收到来自 Secure Enclave 的授权，确认用户已使用 Touch ID 或设备密码认证后，才会允许进行支付。如果可用，Touch ID 即为默认的支付方式；但是用户可随时使用密码来代替 Touch ID。如果尝试通过 Touch ID 匹配指纹三次不成功，会自动提供密码输入选项；五次尝试不成功，则会要求输入密码。如果 Touch ID 尚未配置或没有为 Apple Pay 启用，用户也需要输入密码。

Secure Enclave 和安全元件之间通过串行接口通信：安全元件连接到 NFC 控制器，NFC 控制器连接到应用程序处理器。虽然并非直接相连，但 Secure Enclave 和安全元件可以使用共享的配对密钥进行安全通信，该密钥已在生产过程中预置。加密和认证的通信基于 AES，且都会使用加密随机数来防止重放攻击。配对密钥会使用 Secure Enclave 的 UID 密钥和安全元件的唯一标识符在 Secure Enclave 内部生成。之后配对密钥会在工厂中从 Secure Enclave 安全地传输到硬件安全模块 (HSM) 中，拥有所需密钥材料的该模块会将配对密钥注入安全元件中。

用户授权交易后，Secure Enclave 会向安全元件发送绑定认证随机 (AR) 值的认证类型签名数据以及交易类型的详细信息（非接触式或应用中）。用户在首次预置信用卡时，会在 Secure Enclave 中生成 AR 值，只要 Apple Pay 启用，该值便会一直存在，且会受到 Secure Enclave 的加密和防回滚机制的保护。AR 值会通过配对密钥安全地传送到安全元件中。在收到新的 AR 值后，安全元件会将之前添加的付款卡标记为删除。

只有安全元件经过了授权（使用与添加付款卡时相同的配对密钥和 AR 值），才能使用添加到安全元件的信用卡和借记卡。在以下情形中，这可让 iOS 告知 Secure Enclave 通过将 AR 副本标记为无效来停用付款卡：

密码已停用时。

- 用户注销了 iCloud。
- 用户选择“抹掉所有内容和设置”。
- 设备从恢复模式进行恢复。

对于 Apple Watch，发生以下情况时付款卡会被标记为无效：

- 停用了手表的密码。
- 解除了手表与 iPhone 的配对。
- 关闭了手腕检测功能。

安全元件会使用配对密钥和当前 AR 值的副本来验证接收自 Secure Enclave 的授权，然后启用支付小程序进行非接触式支付。在应用内发生交易时，也会在从支付小程序取回加密的支付数据时应用此过程。

交易专用动态安全码

源自支付小程序的所有支付交易均包括交易专用动态安全码和“设备帐号”。此一次性代码使用计数器和密钥计算，计数器值随每次新交易的产生而递增，密钥则在个性化过程中预置在支付小程序中，为支付网络和/或发卡机构所知。根据支付方案的不同，计算这些代码的过程中还可能使用其他数据，包括：

- 支付小程序生成的随机码
- 如果是 NFC 交易，由终端生成的另一个随机码

  或者

- 如果是应用内交易，由服务器生成的另一个随机码

这些安全码会提供给支付网络和发卡机构，允许其验证每笔交易。完成交易的类型不同，这些安全码的长度也可能有所不同。

使用 Apple Pay 进行非接触式支付

如果 iPhone 已开机且检测到了 NFC 场，它会向用户显示相关的信用卡或借记卡，或者默认的付款卡（可以在“设置”中进行管理）。用户还可以前往 Wallet 应用并选取一张信用卡或借记卡，或在设备锁定时连按主屏幕按钮。

接着，用户必须使用 Touch ID 或密码来认证，之后才会传输支付信息。Apple Watch 解锁后，连按侧边按钮激活默认付款卡进行支付。如果用户不认证，则不会发送支付信息。

用户认证后，在处理支付时会使用“设备帐号”和交易专用动态安全码。Apple 和用户的设备都不会将实际信用卡或借记卡的完整号码发送给商户。Apple 可能会接收诸如交易的大概时间和位置等匿名交易信息，来帮助改进 Apple Pay 和 Apple 的其他产品和服务。

使用 Apple Pay 进行应用内支付

Apple Pay 还可用来在 iOS 应用内进行支付。用户使用 Apple Pay 在应用内支付时，Apple 会收到加密的交易信息，并且会使用商户特定的密钥对其进行重新加密，然后才发送给商户。Apple Pay 会保留诸如大概的购买金额等匿名交易信息。该信息不会绑定到用户，也永不会包括用户购物的内容。

应用发起 Apple Pay 支付交易后，Apple Pay 服务器会先于商户收到来自设备的加密交易。Apple Pay 服务器然后使用商户特定的密钥对其进行重新加密，然后将该交易转给商户。

应用请求支付时，会调用 API 来确定设备是否支持 Apple Pay 以及用户所拥有的信用卡或借记卡能否在商户接受的支付网络中进行支付。应用会请求用来处理和实现交易所需的任意信息，例如收单和收货地址以及联系信息。然后，应用会请 iOS 显示 Apple Pay 表单，表单会请求应用的信息以及其他必要信息，例如要使用的付款卡。

此时应用会收到省市以及邮编信息来计算最终的运费。除非用户使用 Touch ID 或设备密码进行授权支付，否则所请求的整套信息绝不会提供给应用。授权支付后，Apple Pay 表单中显示的信息会传输给商户。

用户授权支付后，会调用 Apple Pay 服务器来获取加密随机数，该随机数与实体店交易中 NFC 终端返回的值类似。接着会将该随机数和其他交易数据一起发送到安全元件以生成使用 Apple 密钥加密的支付凭证。由安全元件生成的加密支付凭证会发送到 Apple Pay 服务器，服务器会解密该凭证，将凭证中的随机数与安全元件发送的随机数进行核对，然后使用与商户 ID 相关联的商户密钥对支付凭证重新加密。紧接着会将该凭证返回给设备，由设备通过 API 交还给应用。应用会将凭证传递给商户系统进行处理。商户然后使用其私钥解密支付凭证来进行处理。该凭证和来自 Apple 服务器的签名可让商户验证该交易是否针对此特定商户。

API 会请求指定受支持商户 ID 的授权。应用还可以包括用以发送到安全元件进行签名的其他数据，例如顺序号或客户身份，以确保交易不会转到其他客户手中。这由应用开发者来实现。应用开发者可以在 PKPaymentRequest 上指定 applicationData。此数据的哈希值会包括在加密的支付数据中。商户负责验证其 applicationData 哈希值与包含在支付数据中的哈希值是否匹配。

回馈卡

自 iOS 9 起，Apple Pay 支持增值服务 (VAS) 协议，用于向兼容的 NFC 终端传输商户回馈卡。VAS 协议可在商户终端上实施，并使用 NFC 来与受支持的 Apple 设备通信。VAS 协议在较短的距离内使用，用于提供补充服务，例如传输回馈卡信息，作为 Apple Pay 交易的一部分。

NFC 终端通过发送付款卡请求开始接收付款卡信息。如果用户的卡片具有商店标识符，将提示用户授权使用该卡。如果商户支持加密，则付款卡信息、时间戳和一次性使用的随机 ECDH P-256 密钥将连同商户的公钥一起用来生成一个加密密钥，以对发送到终端的付款卡数据进行加密。如果商户不支持加密，在发送回馈卡信息前，将提示用户再次向终端出示设备。

暂停使用、移除和抹掉付款卡

通过使用“查找我的 iPhone”将设备置于“丢失模式”，用户可以暂停使用 iPhone 和 iPad 上的 Apple Pay。还可以使用“查找我的 iPhone”、“iCloud”设置或直接在设备上使用 Wallet，从 Apple Pay 移除和抹掉其付款卡。在 Apple Watch 上，用户可以使用 iCloud 设置、iPhone 上的 Apple Watch 应用或者直接从手表上移除付款卡。即使设备离线且未接入蜂窝移动网络或无线局域网络，发卡机构或者各自的支付网络也可停用或移除设备上 Apple Pay 付款卡的支付功能。用户也可以致电发卡机构停用或移除 Apple Pay 中的付款卡。

此外，当用户使用“抹掉所有内容和设置”与“查找我的 iPhone”来抹掉整个设备，或者使用恢复模式来恢复设备时，iOS 会告知安全元件将所有付款卡标记为删除。这等同于立即停用付款卡，直到能够联系 Apple Pay 服务器，从安全元件中完全抹掉付款卡。Secure Enclave 还会单独将 AR 标记为无效，因此将无法使用之前注册的付款卡进行进一步的支付授权。设备在线后，它会尝试联系 Apple Pay 服务器，确保安全元件中的所有付款卡都已被抹掉。

互联网服务

创建 Apple ID 强密码

Apple ID 可用来连接到一系列服务，包括 iCloud、FaceTime 和 iMessage。为帮助用户创建强密码，所有新帐户都必须包含以下密码属性：

- 至少有八个字符
- 至少有一个字母
- 至少有一个大写字母
- 至少有一个数字
- 连续的相同字符不得超过三个
- 不能与帐户名称相同

Apple 构建了一系列强大的服务来帮助用户更充分地使用设备并提高工作效率，其中包括 iMessage、FaceTime、Siri、iCloud、“iCloud 云备份”和“iCloud 钥匙串”。

这些互联网服务在设计上继承了 iOS 在整个平台中推行的安全目标。这些目标包括：安全处理数据，无论数据存储在设备上还是在通过无线网络进行传输；保护用户的个人信息；防范威胁，阻止对信息和服务进行恶意或未经授权的访问。每一种服务都采用自己强大的安全架构，丝毫不会影响 iOS 的整体易用性。

Apple ID

Apple ID 由用户名和密码组成，用来登录 Apple 服务，例如 iCloud、iMessage、FaceTime、iTunes Store、iBooks Store 和 App Store 等等。对于用户而言，安全地保护其 Apple ID 以防止未经授权访问用户的帐户十分重要。为了达成这一目标，Apple 要求使用长度至少为 8 个字符的强密码，同时包含字母和数字，连续的相同字符不得超过三个，且不能为常用的密码。在此规则的基础上，用户可以通过添加更多的字符和标点符号，让密码变得更加强大。在帐户发生重大更改时，Apple 还会向用户发送电子邮件和推送通知。例如，密码或账单信息发生更改，或者在新设备上使用 Apple ID 登录。如有异常发生，Apple 会提示用户立即更改其 Apple ID 密码。

Apple 还提供了 Apple ID 两步式验证，为用户帐户提供了第二重安全性保护。两步式验证启用后，用户必须通过发送至其中一台受信设备的临时代码来验证其身份，然后才允许更改 Apple ID 帐户信息，登录 iCloud、iMessage、FaceTime 和 Game Center，以及在新设备的 iTunes Store、iBooks Store 或 App Store 中进行购物。即使他人知道密码，也可以阻止其访问用户的帐户。如果用户忘记了密码或者无法访问其受信设备，还可以使用储存在安全地方、由 14 位字符组成的“恢复密钥”。

有关 Apple ID 两步式验证的更多信息，请访问 support.apple.com/kb/ht5570?viewlocale=zh_CN。

iMessage

Apple 推出的 iMessage 是一项适用于 iOS 设备和 Mac 电脑的信息收发服务。iMessage 支持文本和附件，例如照片、联系人信息和位置信息。信息会显示在用户所有注册的设备上，这样用户就可以在其他设备上继续对话。iMessage 充分利用了 Apple 推送通知服务 (APNs)。Apple 不记录信息或附件，同时其内容受端到端的加密服务保护，因此只有发送者和接收者可以访问它们。Apple 不能解密这些数据。

当用户在设备上打开 iMessage 后，设备会生成以下两对密钥供这一服务使用：用于加密的 RSA 1280 位密钥和 NIST P-256 曲线上用于签名的 ECDSA 256 位密钥。两组密钥对的私钥存储在设备的钥匙串中，公钥则与设备的 APNs 地址一起发送至 Apple 的目录服务 (IDS)，在目录服务中，公钥会与用户的电话号码或电子邮件地址关联在一起。

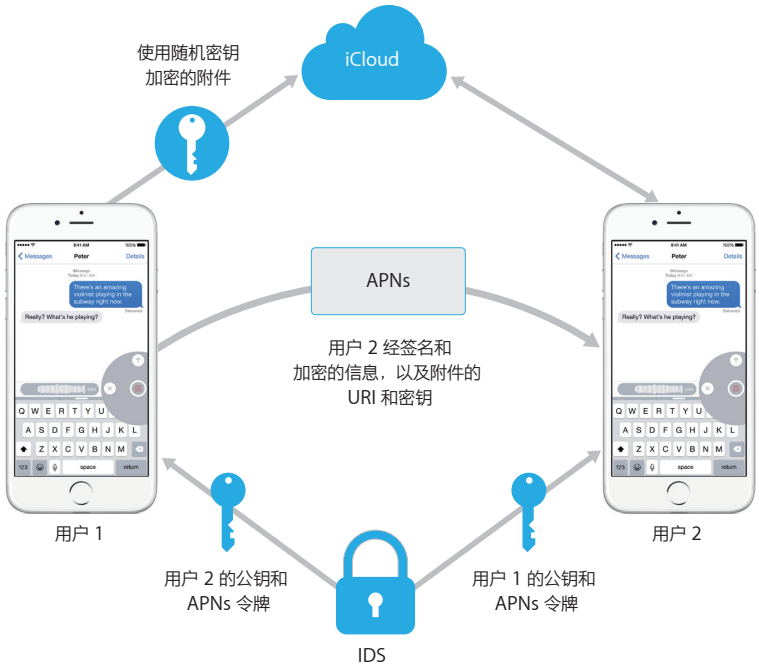
在用户启用其他设备来使用 iMessage 时，它们的加密和签名公钥、APNs 地址以及所关联的电话号码都会添加至目录服务中。用户还可以添加更多电子邮件地址，这些电子邮件地址会通过发送确认链接进行验证。电话号码通过运营商网络和 SIM 卡进行验证。而且，当有新设备、电话号码或电子邮件地址添加进来时，用户所有已注册的设备都会显示一条警告消息。

iMessage 如何发送和接收信息

用户通过输入一个地址或姓名来开始一次 iMessage 对话。如果他们输入一个电话号码或电子邮件地址，设备就会与 IDS 进行联系，来提取与该地址相关联的所有设备的公钥和 APNs 地址。如果用户输入的是一个名字，设备首先会使用用户的“通讯录”应用来收集与该名字相关联的电话号码和电子邮件地址，然后再从 IDS 中获取公钥和 APNs 地址。

对于每个接收者的设备，用户发出的信息都会单独进行加密。接收设备的公共 RSA 加密密钥取自 IDS。对于每台接收设备，发送设备将利用自身所生成的随机 128 位密钥并使用 AES 在 CTR 模式下对信息进行加密。此信息独有的 AES 密钥采用接收设备上用于加密公钥的 RSA-OAEP（算法）进行加密。之后使用 SHA-1 对加密的信息文本和加密的信息密钥进行混编，该哈希值会使用发送设备的专用签名密钥通过 ECDSA 签名。针对每部接收设备所生成的每条信息包含加密的信息文本、加密的信息密钥和发送者的数码签名。信息然后会分派至 APNs 以进行发送。时间戳和 APNs 路由信息等元数据则不加密。与 APNs 的通信使用前向保密 TLS 频道加密。

APNs 最多只能转发大小为 4 KB 或 16 KB 的信息，具体取决于 iOS 的版本。如果信息文本过长，或者附件中有照片等文件，那么附件会使用 AES 在 CTR 模式下通过随机生成的 256 位密钥进行加密并上传至 iCloud。附件的 AES 密钥、其 URI（统一资源标识符）以及加密形式的 SHA-1 哈希值会作为 iMessage 信息的内容发送给收件人。常用的 iMessage 加密会保护以上内容的机密性和完整性，具体如下所述。



对于小组对话，每一位接收者及其设备之间都会重复此过程。

在接收方，每台设备接收到的是 APNs 发来的信息的副本，而且如有需要，设备会从 iCloud 提取附件。如果发送人的电话号码或电子邮件地址与接收者的通讯录相匹配，则会显示一个名字。

与所有推送通知一样，信息在发出之后就会从 APNs 中删除。然而与其他 APNs 通知不同的是，如果设备不在线，iMessage 信息会列入队列等待发送。信息当前会储存长达 30 天。

FaceTime

FaceTime 是 Apple 的视频和音频通话服务。与 iMessage 类似，FaceTime 通话使用 Apple 推送通知服务，与用户已注册过的设备建立初始连接。FaceTime 通话的音频/视频内容由端到端的加密进行保护，因此只有发送者和接收者可以访问它们。Apple 不能解密这些数据。

FaceTime 使用“交互式连接建立”(ICE) 在设备之间建立点对点连接。通过“会话发起协议”(SIP) 消息，设备验证其身份证书并为每个会话建立共享密钥。每台设备提供的加密随机值合并在每个媒体通道的盐密钥 (salt keys) 中，通过使用 AES-256 加密的“安全实时协议”(SRTP) 进行传输。

iCloud

iCloud 可以储存用户的通讯录、日历、照片、文稿和其他内容，并让这些信息在其设备间自动保持最新。它也可以供第三方应用使用，来储存和同步文稿以及由开发者所定义的应用关键数据。用户可以通过登录 Apple ID 来设置 iCloud 并选取想要使用的服务，而 IT 管理员可以通过配置描述文件来停用 iCloud 的部分功能，包括“我的照片流”、iCloud Drive 和“iCloud 云备份”。该服务无法获悉正在储存的内容，并以字节集合的方式对所有文件进行处理。

每个文件被分为区块，并由 iCloud 使用 AES-128 以及从利用 SHA-256 的每个区块内容派生的密钥进行加密。密钥和文件的元数据由 Apple 储存在用户的 iCloud 帐户中。文件的加密区块通过第三方的存储服务（例如 Amazon S3 和 Windows Azure）进行储存，不带任何用户识别信息。

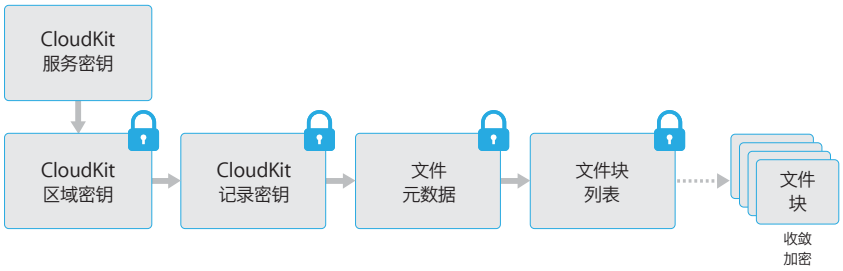
iCloud Drive

iCloud Drive 会添加基于帐户的密钥来保护储存在 iCloud 中的文稿。和现有的 iCloud 服务一样，它会将文件内容分块并进行加密，然后使用第三方服务来储存加密块。但文件内容密钥由记录密钥所封装，与 iCloud Drive 元数据储存在一起。而这些记录密钥则由用户的 iCloud Drive 服务密钥所保护，储存在用户的 iCloud 帐户中。用户可以通过 iCloud 认证来访问 iCloud 文稿元数据，但还必须拥有 iCloud Drive 服务密钥才能显示 iCloud Drive 储存中受保护的部分。

CloudKit

CloudKit 允许应用开发者将键-值数据、结构性数据和资源储存在 iCloud 中。对 CloudKit 的访问通过应用授权进行控制。CloudKit 支持公共数据库和专用数据库。公共数据库可被应用的所有副本使用（通常用作一般性资源），且不加密。专用数据库储存用户的数据。

与 iCloud Drive 一样，CloudKit 使用基于帐户的密钥来保护储存在用户专用数据库中的信息，且与其他 iCloud 服务类似，会使用第三方服务对文件进行分块、加密和储存。CloudKit 使用密钥层级，与数据保护类似。文件独有密钥由 CloudKit 记录密钥封装。而记录密钥则会受到区域范围内密钥的保护，而区域范围内的密钥则受到用户的 CloudKit 服务密钥的保护。CloudKit 服务密钥储存在用户的 iCloud 帐户中，只有在用户使用 iCloud 认证后才可使用。



iCloud 云备份

iCloud 每天通过无线局域网将信息（包括设备设置、应用数据、“相机胶卷”中的照片和视频、“信息”应用中的对话）备份。iCloud 会在您通过互联网发送内容时对其进行加密、以加密格式储存内容并使用安全令牌进行认证，来保障内容的安全。仅当设备处于锁定状态、连接到电源且可通过无线局域网访问互联网时，“iCloud 云备份”才会工作。由于 iOS 采用独特的加密技术，因此系统经过专门设计，既可保护数据安全，又能进行无人值守式增量备份和还原。

以下是 iCloud 云备份的内容：

- 关于已购买的音乐、影片、电视节目、应用和图书的信息，但不包括已购买的内容本身
- “相机胶卷”中的照片和视频
- 通讯录、日历事件、提醒事项和备忘录
- 设备的设置信息
- 应用数据
- 添加到 iBooks 但未购买的 PDF 和图书
- 通话记录
- 主屏幕与应用排列方式
- iMessage 信息、短信 (SMS) 和彩信 (MMS)
- 电话铃声
- HomeKit 数据
- HealthKit 数据

当文件从设备锁定时无法访问的数据保护类中创建时，其文件独有密钥通过“iCloud 云备份”密钥包中的类密钥进行加密。文件以其原始的加密状态备份至 iCloud。处于“无保护”数据保护类中的文件在传输期间进行加密。

“iCloud 云备份”密钥包包含每个数据保护类的非对称 (Curve25519) 密钥，这些密钥用于加密文件独有密钥。有关备份密钥包和“iCloud 云备份”密钥包内容的更多信息，请参阅“加密和数据保护”部分中的“钥匙串数据保护”。

备份集储存于用户的 iCloud 帐户中，包括用户的文件副本和“iCloud 云备份”密钥包。而随备份集一同储存的随机密钥会保护“iCloud 云备份”密钥包。（用户的 iCloud 密码不用于加密，因此更改 iCloud 密码不会使现有备份失效。）

当用户的钥匙串数据库备份至 iCloud 时，它仍然受到与 UID 绑定的密钥保护。这样使得钥匙串仅能恢复至生成它的同一台设备，意味着任何人（包括 Apple）均无法读取用户的钥匙串项。

恢复后，备份的文件、“iCloud 云备份”密钥包和密钥包的密钥将从用户的 iCloud 帐户取回。“iCloud 云备份”密钥包通过其密钥进行解密，然后密钥包中的文件独有密钥用于解密备份集中的文件，这些文件作为新文件写入文件系统中，从而根据其数据保护类对其重新加密。

Safari 与 iCloud 钥匙串整合

Safari 可以自动为网站密码生成强加密随机字符串，然后将其储存在钥匙串中并与其他设备同步。钥匙串项通过 Apple 服务器在不同的设备之间传输，但会严格进行加密，Apple 和其他设备无法读取其内容。

iCloud 钥匙串

iCloud 钥匙串可帮助用户在 iOS 设备和 Mac 电脑之间安全地同步密码，而不会将此信息泄露给 Apple。除了强大的隐私保护和安全性，易用性和恢复钥匙串的功能对 iCloud 钥匙串的设计和架构也具有重要影响。iCloud 钥匙串包含两项服务：钥匙串同步和钥匙串恢复。

Apple 设计的 iCloud 钥匙串和钥匙串恢复可确保用户的密码在下列情况下仍然受到保护：

- 用户的 iCloud 帐户被盗。
- iCloud 遭到外部攻击者或员工的入侵。
- 第三方访问用户帐户。

钥匙串同步

当用户首次启用 iCloud 钥匙串时，设备将建立信任圈并为自己创建同步身份。同步身份包括私钥和公钥。同步身份的公钥放置在信任圈中，该信任圈已签名两次：第一次由同步身份的私钥签署，第二次由来自用户 iCloud 帐户密码的非对称椭圆密钥（使用 P256）签署。随信任圈一起储存的还有参数（随机盐密钥和迭代次数），用于创建基于用户 iCloud 密码的密钥。

已签名的同步信任圈放置在用户的 iCloud 密钥值存储区域。如果不知道用户的 iCloud 密码，就无法对其进行读取；如果没有信任圈成员同步身份的私钥，就无法对其进行有效地修改。

当用户在其他设备上启用 iCloud 钥匙串时，新设备将在 iCloud 中通知用户该设备不是之前已建立的同步信任圈的成员之一。该设备将创建其同步身份密钥对，然后创建应用程序申请单以请求加入该信任圈。该申请单包括设备的同步身份公钥，系统将要求用户使用其 iCloud 密码进行认证。椭圆密钥生成参数通过 iCloud 取回，并生成用于签署应用程序申请单的密钥。最终，应用程序申请单将放置在 iCloud 中。

当第一台设备接收到应用程序申请单时，它会显示一则通知，让用户确认新设备正在请求加入同步信任圈。该用户输入其 iCloud 密码，应用程序申请单通过匹配的私钥签名进行验证。这样即确认发出请求加入信任圈的人在发出请求时输入了用户的 iCloud 密码。

用户批准将新设备添加至信任圈后，第一台设备将新成员的公钥添加至同步信任圈，使用其同步身份和来自用户 iCloud 密码的密钥再次签名。新的同步信任圈放置在 iCloud 中，该信任圈的新成员同样进行了签名。

假设签名信任圈有两个成员，并且每个成员拥有与其配对的公钥。它们现在开始通过 iCloud 键值存储交换各个钥匙串项。如果两个信任圈成员拥有相同的项目，系统将同步修改日期最近的项目。如果另一个成员拥有该项目并且修改日期相同，这些项目将被跳过。每个同步的项目专门针对目的设备进行加密。其他设备或 Apple 均无法解密。另外，加密的项目仅在 iCloud 中短暂存储；它会被同步的每个新项目所覆盖。

当新设备加入同步信任圈时，将会重复该过程。例如当第三台设备加入时，用户的另外两台设备上均会出现确认消息。用户可以从其中任意一台设备上批准新成员。随着新的同级设备加入，每台同级设备均与新设备进行同步，以确保所有成员拥有相同的钥匙串项。

但是，整个钥匙串不会进行同步。某些项目仅限于特定的设备（例如 VPN 身份），它们不会离开设备。仅具有 `kSecAttrSynchronizable` 属性的项目会被同步。Apple 已经为 Safari 用户数据（包括用户名、密码和信用卡号）、无线局域网密码以及 HomeKit 加密密钥设置了该属性。

另外，在默认情况下，第三方应用添加的钥匙串项不会进行同步。将项目添加至钥匙串时，开发者必须设置 `kSecAttrSynchronizable`。

钥匙串恢复

钥匙串恢复功能使用户可以将其钥匙串交与 Apple 托管，但不允许 Apple 读取密码和钥匙串包含的其他数据。即使用户只有一台设备，钥匙串恢复也可以提供安全的网络来防止数据丢失。当 Safari 用于为 Web 帐户生成随机强密码时，这尤其重要，因为这些密码的唯一记录在钥匙串中。

钥匙串恢复包含两大基本要素：二次身份认证和安全托管服务，后者是 Apple 专为支持此功能而创建的服务。用户的钥匙串通过强密码进行加密，只有满足一系列严格的条件，托管服务才会提供钥匙串副本。

当 iCloud 钥匙串打开时，系统要求用户创建 iCloud 安全码。恢复托管的钥匙串需要此安全码。默认情况下，系统要求用户提供简单的 4 位安全码数值。但用户也可以自行指定较长的代码或允许其设备创建随机密码，他们可以自行记录和保存。

然后，iOS 设备会导出用户的钥匙串副本，将其与密钥加密封装于非对称钥匙包中，并放置在用户的 iCloud 键值存储区域。密钥包被用户的 iCloud 安全码和储存托管记录的 HSM（硬件安全模块）集群公钥所封装。它会变成用户的 iCloud 托管记录。

如果用户决定接受随机加密的安全码而不自行指定或使用 4 位数值，则不再需要托管记录。而 iCloud 安全码用于直接封装随机密钥。

除了建立安全码，用户必须注册电话号码。这用于在钥匙串恢复期间提供二级身份认证。用户将收到一条短信，必须回复这条短信才能继续恢复过程。

托管安全性

iCloud 为钥匙串托管提供了安全的基础架构，可确保只有经过授权的用户和设备才能执行恢复。iCloud 背后部署的是硬件安全模块 (HSM) 集群。这些集群为托管记录提供保护。集群的每位成员都有一个密钥，用于根据以上所述对其监管下的托管记录进行加密。

要恢复钥匙串，用户必须使用其 iCloud 帐户和密码进行身份验证，并对发送至其注册电话号码的短信进行回复。回复完成后，用户必须输入其 iCloud 安全码。HSM 集群利用安全远程密码协议 (SRP) 验证用户是否知道其 iCloud 安全码；密码本身不会发送给 Apple。集群的每个成员单独验证用户是否未超过检索记录所允许的最大尝试次数，如下所述。如果多数成员同意，集群会打开托管记录并将其发送至用户的设备。

下一步，设备使用 iCloud 安全码打开用于加密用户的钥匙串的随机密钥。利用该密钥，可以解密从 iCloud 键值存储取回的钥匙串并将其恢复到设备中。最多允许对托管记录认证和检索 10 次。多次尝试失败后，记录将被锁定，用户必须联系 Apple 技术支持部门才能进行更多尝试。第 10 次尝试失败后，HSM 集群将销毁托管记录，钥匙串将永久丢失。这种方式以牺牲钥匙串数据为代价，防止强行访问检索记录。

这些策略已编入 HSM 固件中。允许更改固件的管理访问卡已经被销毁。任何尝试更改固件或访问私钥的操作，都会导致 HSM 集群删除私钥。如果发生这种情况，受集群保护的所有钥匙串的所有者将会收到消息，告知其托管记录已经丢失。然后他们可以选择重新注册。

Siri

用户只需自然地开口讲话，即可借助 Siri 来发送消息、安排会议、拨打电话以及执行其他操作。Siri 采用语音识别、文本至语音以及“客户端-服务器”模式，可响应多种请求。Siri 支持的任务经过专门设计，可确保只使用尽可能少的个人信息，并对这些信息提供全面保护。

Siri 开启后，设备将创建随机标识符，用于语音识别和 Siri 服务器。这些标识符仅用于 Siri 内部，用于改善服务。如果 Siri 随后被关闭，设备将生成新的随机标识符以便在 Siri 重新打开时使用。

为了改进 Siri 功能，设备中的某些信息会被发送至服务器。这些信息包括：音乐资料库（歌曲名称、表演者和播放列表）、“提醒事项”列表名称以及“通讯录”中定义的姓名和关系。设备与服务器进行的所有通信均通过 HTTPS 来完成。

启动 Siri 会话后，系统会将用户的名字和姓氏（来自“通讯录”）以及大致的地理位置发送至服务器。这样，Siri 便可以使用姓名回应或回答那些只需要大致位置的问题，比如天气信息。

如果需要更精确的位置，例如确定附近电影院的位置，服务器将要求设备提供更精确的位置。以上示例说明了默认情况下信息如何发送至服务器（仅在为了处理用户请求而非常有必要发送的情况下）。任何情况下，只要 10 分钟不活动，会话信息就会被丢弃。

如果在 Apple Watch 上使用 Siri，手表会创建自己的唯一随机标识符（如上所述）。但其请求还会发送配对 iPhone 的 Siri 标识符来为该信息提供参考，而非再次发送用户信息。

用户的讲话录音会被发送至 Apple 的语音识别服务器。如果任务仅涉及听写，识别出的文本将被发回到设备中。否则，Siri 会对文本进行分析，必要时将其与设备相关描述文件的信息相结合。例如，如果请求是“给妈妈发信息”，系统将会利用从“通讯录”上传的关系和姓名。然后已确认操作的命令将被发回至要执行命令的设备。

许多 Siri 功能是由设备按照服务器的指令来完成的。例如，当用户要求 Siri 读出收到的消息时，服务器就会告诉设备读出其未读消息的内容。消息的内容和发送者不会发送至服务器。

用户讲话录音将被存储 6 个月，以便识别系统能够利用它们更好地理解用户的讲话。6 个月后将存储另一份不带标识符的副本，以供 Apple 持续改进和开发 Siri，存储时间不超过两年。另外，某些引用音乐、体育团队和队员以及商业或兴趣点的录音同样得以存储，用于改善 Siri。

无需动手，用户通过语音也可以激活 Siri。语音触发检测会在本地设备上进行。在这种模式下，只有当传入的音频模式与指定触发短语的原声高度匹配时，Siri 才会激活。检测到语音触发后，对应的音频（包括后续的 Siri 命令）会发送到 Apple 的语音识别服务器作进一步处理，这一过程遵循与通过 Siri 执行其他用户录音相同的规则。

连续互通

连续互通充分利用了诸如 iCloud、蓝牙和无线局域网等技术，让用户在另一台设备上继续从事在前一台设备上进行的活动、拨打和接听电话、发送和接收文本信息以及共享蜂窝移动互联网连接。

Handoff

当用户的 Mac 和 iOS 设备彼此接近时，用户可以使用 Handoff 功能，自动将正在处理的内容从一台设备传送到另一台设备。用户可以使用 Handoff 功能来切换设备并立即继续工作。

当用户在第二台支持 Handoff 功能的设备上登录 iCloud 时，两台设备使用 Apple 推送通知服务 (APNs) 建立频段外的低功耗蓝牙 4.0 配对。单个信息采用与 iMessage 相似的加密方式。设备配对后，每台设备都会生成对称的 256 位 AES 密钥，并储存在设备的钥匙串中。此密钥用于加密和认证低功耗蓝牙广播。低功耗蓝牙广播会在 GCM 模式下使用 AES-256 并采用重放保护措施，将设备的当前活动传递给其他已配对的 iCloud 设备。设备首次接收到来自新密钥的广播时，它会建立与发起设备之间的低功耗蓝牙连接并交换广播加密密钥。该连接使用标准的低功耗蓝牙 4.0 加密进行保护，和单个信息的加密相同（与 iMessage 的加密方式类似）。在某些情况下，这些信息会使用 Apple 推送通知服务，而非低功耗蓝牙。活动负载采用与 iMessage 相同的方式进行保护和传输。

在本地应用和网站之间使用 Handoff 功能

Handoff 功能可允许本地 iOS 应用继续访问由应用开发者合法控制域中的网页。它还允许本地应用的用户活动在 Web 浏览器中继续进行。

为阻止本地应用要求继续访问不是由开发者控制的网站，应用必须证明对其要继续访问的 Web 域具有合法控制权。对网站域的控制是通过共享的 Web 凭证所使用的机制来建立。有关详细信息，请参阅“加密和数据保护”部分中的“访问 Safari 已存储的密码”。在允许应用接受使用 Handoff 功能的用户操作前，系统会验证应用的域名控制。

使用 Handoff 功能传送的网页来源可以是任何采用了 Handoff API 的浏览器。当用户浏览网页时，系统会使用加密的 Handoff 广播字节来广播网页的域名。只有用户的其他设备能够解密该广播字节（之前在上述部分中有所描述）。

在接收设备上，系统会检测到安装的本地应用接受了来自已经广播域名的 Handoff，并将该本地应用图标显示为 Handoff 选项。开启后，本地应用会接收完整的 URL 以及网页标题。浏览器中的其他信息不会被传送到本地应用。

相反，如果 Handoff 接收设备未安装相同的本地应用，本地应用可能会指定回退 URL。如果出现这种情况，系统会将用户的默认浏览器显示为 Handoff 应用选项（如果该浏览器采用了 Handoff API）。请求使用 Handoff 时，系统会启动浏览器并使用来源应用提供的回退 URL。回退 URL 并不一定要限制为由本地应用开发者控制的域名。

使用 Handoff 传送较大的数据

除了 Handoff 的基本功能外，一些应用可能会选择使用支持发送大量数据（通过 Apple 开创的点对点无线局域网技术，与 AirDrop 类似）的 API。例如，“邮件”应用使用这些 API，通过 Handoff 功能来传送可能包含较大附件的邮件草稿。

应用使用此功能时，两台设备间开始交换，如同使用 Handoff 传送一样（请参阅上述部分）。但在使用低功耗蓝牙收到初始负载后，接收设备会通过无线局域网发起新的连接。此连接会使用 TLS 加密（交换其 iCloud 身份证书）。身份证书中的身份标识会针对每位用户的身份进行验证。其他负载数据会通过此加密的连接进行发送，直到传输完成。

iPhone 蜂窝移动网络通话中继

当您的 Mac、iPad 或者 iPod 连接到与 iPhone 相同的无线局域网络时，可以通过 iPhone 的蜂窝移动网络连接来拨打和接听电话。配置设备时，要求使用相同的 Apple ID 帐户，同时登录到 iCloud 和 FaceTime。

来电时，会通过 Apple 推送通知服务 (APNs) 来通知所有已配置的设备，每个通知都会使用与 iMessage 所用相同的端到端加密技术。连接到相同网络的设备上会显示来电通知用户界面。接通电话时，会使用安全的点对点连接技术在两台设备间无缝传输 iPhone 的音频。

呼出的通话也将通过 Apple 推送通知服务中继到 iPhone，并通过安全的点对点链接在设备间传输音频。

用户可以在 FaceTime 设置中关闭“iPhone 蜂窝移动网络通话”来停用设备的电话中继功能。

iPhone 短信转发

“短信转发”会自动将 iPhone 上接收的短信发送到用户注册的 iPad、iPod touch 或 Mac 上。每台设备均须使用相同的 Apple ID 帐户登录 iMessage 服务。“短信转发”打开后，会要求在每台设备上输入由 iPhone 随机生成的 6 位数字代码来验证注册。

设备链接后，iPhone 会使用本文 iMessage 部分中所描述的方法，将发来的短信进行加密并转发给每台设备。并使用相同的方法将回复发送回 iPhone，之后 iPhone 使用运营商的短信传输机制将回复以短信形式发送。用户可以在“信息”设置中打开或关闭“短信转发”。

Instant Hotspot

支持 Instant Hotspot 的 iOS 设备使用低功耗蓝牙来发现设备并与之通信，前提是设备需要使用相同的 iCloud 帐户进行登录。兼容 Instant Hotspot 且运行 OS X Yosemite 或更高版本的 Mac 电脑，使用相同的技术来发现支持 Instant Hotspot 的 iOS 设备，并与之通信。

用户在进入 iOS 设备上的无线局域网设置时，设备会发出低功耗蓝牙信号，该信号包含可被所有登录到相同 iCloud 帐户设备接受的标识符。该标识符由绑定到 iCloud 帐户的 DSID（目的地发讯识别器）生成，并会定期更新。当其他登录到相同 iCloud 帐户的设备彼此接近且支持个人热点时，它们会检测到信号并作出响应，以表示它们处于可通信状态。

用户选择支持个人热点的设备时，会向该设备发送打开“个人热点”的请求。而该请求会通过加密的链接（使用标准低功耗蓝牙加密方法）进行发送；请求的加密方式与 iMessage 的加密方式类似。之后，设备会使用包含个人热点连接信息的相同信息独有加密方式，通过同一低功耗蓝牙链接作出响应。

设备控制

iOS 支持一系列灵活的安全性策略和配置，易于强制执行和管理。这使得公司能保护公司信息并确保员工符合企业要求，即便员工使用的是自带设备也无妨，例如，在参与“自带设备办公”(BYOD)计划的过程中。

公司可以使用密码保护、配置描述文件、远程擦除和第三方 MDM 解决方案等资源来管理设备群，并确保公司数据的安全，甚至员工在自己的个人 iOS 设备上访问这些数据时也能确保安全。

密码保护

默认情况下，用户的密码可以定义为一个数字 PIN 码。在配备 Touch ID 的设备上，最小的密码长度为六位数。在其他设备上，最小长度为四位数。用户可以指定更长的字母数字密码，方法是在“设置”>“密码”的“密码选项”中选择“自定字母数字密码”。建议企业使用较长、较复杂的密码，因为这样的密码很难被猜中或遭到攻击。

管理员可利用 MDM 或 Exchange ActiveSync 或要求用户手动安装配置描述文件，强制实施复杂密码的要求和其他策略。以下密码策略可供使用：

- 允许简单值
- 要求字母数字值
- 最短的密码长度
- 最少的复杂字符数
- 密码的最长有效期
- 密码历史记录
- 自动锁定超时
- 设备锁定宽限期
- 最多可允许的尝试失败次数
- 允许 Touch ID

有关各项策略的详细信息，请参阅《Configuration Profile Key Reference》（配置描述文件键参考）文稿，网址是 developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/。

iOS 配对模型

iOS 使用配对模型从主电脑来控制对设备的访问。配对会在设备及其连接的主机之间通过公钥交换来建立信任关系。iOS 使用这种信任关系来启用与连接的主机之间的附加功能，例如数据同步。在 iOS 9 中，要求配对的服务只有在用户解锁设备后才能开始。

配对过程要求用户解锁设备并在主机上接受配对请求。用户执行此操作后，主机和设备会交换并存储 2048 位 RSA 公钥，然后主机获得了 256 位密钥，可解锁储存在设备上的托管密钥包（请参阅“密钥包”部分中的“托管密钥包”）。设备在将受保护的数据发送给主机或启动服务（例如 iTunes 同步、文件传输或 Xcode 开发等）前，会要求使用交换的密钥来启动加密的 SSL 会话。设备要求主机通过无线局域网进行连接，以将此加密的会话用于所有通信，因此之前必须通过 USB 进行配对。配对还会启用多项诊断功能。在 iOS 9 中，如果某个配对记录超过六个月未被使用，该记录将过期。有关更多信息，请访问 support.apple.com/kb/HT6331?viewlocale=zh_CN。

某些服务限制为通过 USB 工作，例如 com.apple.pcapd。此外，com.apple.file_relay 服务要求安装 Apple 签名的配置描述文件。

用户可以清除受信任的主机列表，方法是使用“还原网络设置”或者“还原位置与隐私”选项。有关更多信息，请参阅 support.apple.com/kb/HT5868?viewlocale=zh_CN。

配置执行

配置描述文件是一个 XML 文件，管理员可通过它向 iOS 设备分发配置信息。用户无法更改由已安装的配置描述文件定义的设置。如果用户删除配置描述文件，由该描述文件定义的所有设置也将随之删除。这样，管理员可以通过将策略与访问结合在一起来实施设置。例如，提供电子邮件配置的配置描述文件还可以指定设备密码策略。除非密码符合管理员的要求，否则用户将无法访问邮件。

iOS 配置描述文件包含多项可以指定的设置：

- 密码策略
- 针对设备功能的访问限制（例如停用相机）
- 无线局域网设置
- VPN 设置
- 邮件服务器设置
- Exchange 设置
- LDAP 目录服务设置
- CalDAV 日历服务设置
- Web Clip
- 凭证和密钥
- 高级蜂窝移动网络设置

可以对配置描述文件进行签名和加密来验证其来源、确保其完整性并保护其内容。配置描述文件采用支持 3DES 和 AES-128 的 CMS (RFC 3852) 进行加密。

配置描述文件还可以锁定到设备，以便彻底禁止删除它们，或只允许使用密码将其删除。由于许多企业用户使用的是自己的 iOS 设备，因此可以删除将设备绑定到 MDM 服务器的配置描述文件，但这样做也会删除所有被管理的配置信息、数据和应用。

用户可以使用 Apple Configurator 直接在其设备上安装配置描述文件，或者通过 Safari 下载配置描述文件、通过邮件信息发送或者使用 MDM 服务器以无线方式发送。

移动设备管理 (MDM)

iOS 支持 MDM，可让企业在其组织内部安全地配置和管理规模化的 iPhone 和 iPad 部署方案。MDM 功能建立在现有的 iOS 技术的基础之上，如配置描述文件、无线注册和 Apple 推送通知服务 (APNs)。例如，APNs 用于唤醒设备，使设备可以通过安全的连接与其 MDM 服务器直接通信。在此过程中，不会通过 APNs 传输任何机密或专属信息。

利用 MDM，IT 部门可在企业环境中注册 iOS 设备、无线配置和更新设置、监控公司政策的贯彻情况，甚至可以远程擦除或锁定被管理的设备。有关移动设备管理的更多信息，请参阅 www.apple.com/iphone/business/it/management.html。

Device Enrollment Program

Device Enrollment Program（DEP，设备注册计划）可让组织快速便捷地部署直接从 Apple 或参与的 Apple 授权经销商和运营商购买的 iOS 设备。在用户获得设备前，组织无需实际操作或者准备设备，而是在 MDM 中自动注册设备。通过在“设置助理”中移除特定的步骤，可进一步简化用户的设置过程，方便用户快速使用。管理员还可以控制用户能否将 MDM 描述文件从设备移除，并确保设备访问限制从一开始就准备到位。例如，用户可以直接从 Apple 订购设备，配置所有管理设置，然后将设备直接邮寄到其家庭地址。开箱并激活后，设备会在组织的 MDM 中注册。用户可以直接使用所有管理设置、应用和图书。

过程非常简单：注册计划后，管理员登录到该计划网站，将该计划链接到 MDM 服务器，“声明” iOS 设备购买自 Apple。然后设备就可以通过 MDM 分配给用户了。分配用户后，所有 MDM 指定的配置、访问限制或控制都会自动安装。有关更多信息，请访问 deploy.apple.com。

注：Device Enrollment Program（设备注册计划）并非在所有国家或地区都可用。

Apple Configurator

除了 MDM，OS X 版 Apple Configurator 也可以让所有人都能够轻松部署 iOS 设备。Apple Configurator 可用于为大量的设备快速配置应用、数据、访问限制和设置。

监督

在设备设置阶段，组织可以将设备配置为被监督。监督意味着设备由机构拥有，从而对设备的配置和访问限制提供额外控制。设备可以在设置阶段通过 Device Enrollment Program（设备注册计划）或 Apple Configurator 配置为被监督。

有关使用 MDM 或 Apple Configurator 来配置和管理设备的更多信息，请参阅《iOS 部署参考》，网址为 help.apple.com/deployment/ios。

有关针对被监督设备的额外控制的信息，请参阅《配置描述文件参考》，网址为：developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/iPhoneConfigurationProfileRef.pdf。

设备访问限制

管理员可以安装配置描述文件来限制设备功能。一些可用的访问限制包括：

- 允许应用安装
- 允许信任企业级应用
- 允许使用相机
- 允许 FaceTime
- 允许屏幕快照
- 锁定时允许语音拨号
- 允许在漫游时自动同步
- 允许 App 内购买
- 允许同步最近的“邮件”信息
- 强制用户为所有购买输入商店密码
- 设备锁定时允许使用 Siri
- 允许未被管理的目的位置中包含来自被管理的来源中的文稿
- 允许被管理的目的位置中包含来自未被管理的来源中的文稿
- 允许 iCloud 钥匙串同步
- 允许以无线方式更新证书信任数据库
- 允许在锁定屏幕上显示通知
- 强制 AirPlay 连接使用配对密码
- 允许 Spotlight 显示用户通过互联网生成的内容
- 允许使用 Handoff
- 将 AirDrop 视为未被管理的目的位置
- 允许备份企业级图书
- 允许在用户的设备间同步企业级图书中的笔记和书签
- 允许使用 Safari
- 启用 Safari 自动填充
- 启用 JavaScript
- 限制 Safari 中广告跟踪
- 阻止弹出式窗口
- 接受 Cookie
- 允许 iCloud 云备份
- 允许 iCloud 文稿和键值同步
- 允许 iCloud 照片共享
- 允许将诊断信息发送给 Apple
- 允许用户接受不被信任的 TLS 证书
- 强制执行加密备份
- 允许 Touch ID
- 允许在锁定屏幕使用控制中心
- 允许在锁定屏幕上显示“今天”视图
- 要求 Apple Watch 进行手腕检测

仅限被监督设备的访问限制

- 允许 iMessage
- 允许移除应用
- 允许手动安装配置描述文件
- HTTP 的全球网络代理
- 允许配对到电脑进行内容同步
- 使用白名单和可选连接密码限制 AirPlay 连接
- 允许 AirDrop
- 允许“查找我的朋友”修改
- 对于一些被管理的应用，允许自发进入“单应用模式”
- 允许帐户修改
- 允许蜂窝移动网络数据修改
- 允许主机配对 (iTunes)
- 允许激活锁
- 防止抹掉所有内容和设置
- 防止启用访问限制
- 第三方内容过滤器
- 单应用模式
- 始终打开 VPN
- 允许修改密码
- 允许 Apple Watch 配对
- 允许自动下载应用
- 允许键盘预测、自动改正、拼写检查和快捷键

有关访问限制的更多信息，请参阅 developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/iPhoneConfigurationProfileRef.pdf

远程擦除

管理员或用户可以远程擦除 iOS 设备。通过安全地丢弃可擦除存储器中的块存储加密密钥，使所有数据均不可读，实现即时远程擦除。用户可以通过 MDM、Exchange 或 iCloud 发起远程擦除命令。

通过 MDM 或 iCloud 触发远程擦除命令后，设备会发送确认并执行擦除操作。如果通过 Exchange 进行远程擦除，在执行擦除前，设备会签入 Exchange 服务器。

用户还可以使用“设置”应用来擦除自己设备上的数据。如上所述，可以将设备设置为在连续多次输入密码失败后自动擦除。

查找我的 iPhone 和激活锁

设备丢失或被盗后，取消激活和抹掉设备很重要。在 iOS 7 或更高版本中，如果启用了“查找我的 iPhone”，必须输入所有者的 Apple ID 凭证，才能重新激活设备。组织最好监督其设备或实施政策让用户停用该功能，这样“查找我的 iPhone”不会阻止组织将设备分配给其他用户。

在 iOS 7.1 或更高版本中，当用户打开“查找我的 iPhone”后，可使用兼容的 MDM 解决方案来启用被监督设备上的激活锁。MDM 管理员可使用 Apple Configurator 或者 Device Enrollment Program（设备注册计划）来监督设备，从而管理“查找我的 iPhone”激活锁。MDM 解决方案会在激活锁启用后储存忽略码，之后在需要抹掉设备并分配给新用户时，使用该代码来自动清除激活锁。请参阅 MDM 解决方案文稿以了解详细信息。

重要事项：默认情况下，即使用户打开了“查找我的 iPhone”，被监督的设备也绝不会启用激活锁。但 MDM 服务器仍有可能取回忽略码并允许启用设备上的激活锁。如果 MDM 服务器在启用激活锁时，“查找我的 iPhone”已打开，激活锁会在此时启用。如果 MDM 服务器启用激活锁时，“查找我的 iPhone”已经关闭，那么激活锁会在下次用户激活“查找我的 iPhone”时启用。

隐私控制

Apple 十分重视客户隐私，在 iOS 中内建了一系列控制和选项，允许 iOS 用户决定应用如何使用其信息，何时使用其信息以及使用何种信息。

定位服务

定位服务使用 GPS、蓝牙、众包的无线局域网热点以及信号发射塔位置来确定用户的大概位置。使用“设置”中的一个开关就可以关闭“定位服务”，或者为每个使用该服务的应用批准该项服务。仅当应用正在使用，或任何时候允许其访问定位服务时，它才会请求接收位置数据。用户可以选择不允许定位服务，或者在“设置”中随时更改选择。用户可以在“设置”中，将访问权限设定为永不允许、使用时允许或者始终允许，具体取决于应用所请求的定位用途。而且，如果被允许随时访问地址服务的应用在后台模式中运行时，系统会提醒用户地址服务已获批准，且可以关闭该服务。

此外，用户可以对位置信息的系统服务用途进行细微控制，这可让他们关闭提供以下信息的服务：Apple 用于改进 iOS 的诊断和用量服务所收集信息中的定位信息、基于位置的 Siri 信息、当地交通状况和用来估计行程时间的常去地点信息。

访问个人数据

iOS 可帮助阻止应用在未获得许可的情况下访问用户的个人信息。此外，用户可以在“设置”中查看已经批准了哪些应用访问特定信息，也可以批准或撤销未来的访问权限。这包括对以下内容的访问权限：

- 通讯录
- 日历
- 提醒事项
- 照片
- iPhone 5s 或后续机型上的运动记录
- 社交媒体帐户
- 麦克风
- 相机
- HomeKit
- HealthKit
- 蓝牙共享

如果用户登录 iCloud，则会默认授予应用访问 iCloud Drive 的权限。用户可以在“设置”的 iCloud 选项中控制每个应用的访问权限。此外，iOS 提供的访问限制可阻止数据在 MDM 所安装和用户所安装的应用和帐户之间移动。

隐私政策

有关 Apple 的隐私政策，请在线访问 www.apple.com/cn/legal/privacy。

结束语

安全性承诺

Apple 致力于使用领先的隐私和安全性技术来帮助客户保护其个人信息，以及采用全面的方法来保护商业环境中企业的数据。

安全性植根于 iOS 中。从平台到网络再到应用，公司需要的一切在 iOS 平台上应有尽有。这些元素一起构成了 iOS 业界领先的安全性和良好的用户体验。

Apple 在整个 iOS 和 iOS 应用生态系统中使用统一的集成安全基础架构。基于硬件的存储加密可在设备丢失时提供远程擦除功能，并让用户在将设备出售或转让给其他用户时能够彻底清除设备中的所有公司和个人信息。此外，诊断信息也以匿名方式收集。

由 Apple 设计的 iOS 应用将增强安全性放在重要位置。Safari 支持在线证书状态协议 (OCSP)、EV 证书以及证书验证警告，为用户提供了安全的浏览体验。“邮件”通过 S/MIME 对邮件进行认证和加密，且支持对单封邮件的操作，这样 S/MIME 用户可以选择默认对所有邮件进行签名和加密，也可以选择性地保护单封邮件。iMessage 和 FaceTime 也提供客户端到客户端的加密。

对于第三方应用，必需的代码签名、沙盒化以及授权相结合，为用户提供了强有力的保护，以防受到病毒、恶意软件以及破坏其他平台安全的其他攻击的侵害。App Store 提交流程通过对每个 iOS 应用进行严格审核然后才允许在 App Store 上出售，进一步保护用户免遭上述风险的侵害。

为充分利用 iOS 内置的大量安全性功能，我们鼓励企业审视自身的 IT 和安全策略，以确保充分利用该平台提供的多重安全技术。

Apple 拥有一支专门的安全团队，负责为所有 Apple 产品提供支持。该团队为开发中和已发布的产品提供安全审核和测试。Apple 团队还提供安全工具和培训，并积极监控对新增安全问题和威胁的报告。Apple 是事件响应与安全组织论坛 (FIRST) 的成员。要进一步了解如何向 Apple 报告问题以及如何订阅安全通知，请访问 apple.com/cn/support/security。

术语表

地址空间布局随机化 (ASLR)	iOS 所采用的一项技术，旨在让恶意利用软件错误的成功机率降至最低。通过确保内存地址和偏移量不可预测，使攻击代码无法对这些值进行硬编码。在 iOS 5 和更高版本中，所有系统应用和资源库的位置都是随机安排的，所有第三方应用均编译为与位置无关的可执行程序。
Apple 推送通知服务 (APNs)	一项由 Apple 提供的全球服务，用于向 iOS 设备推送通知。
Boot ROM	设备的处理器在首次启动时所执行的第一个代码。作为处理器不可分割的一部分，Apple 或攻击者均无法对其进行更改。
数据保护	iOS 的文件和钥匙串保护机制。它也可以指应用用来保护文件和钥匙串项的 API。
设备固件升级 (DFU)	设备的 Boot ROM 代码在等待通过 USB 进行恢复时所处的模式。处于 DFU 模式时，设备为黑屏。但在连接到运行 iTunes 的电脑时，会出现以下提示：“iTunes 检测到一个处于恢复模式的 iPad。您必须先用备份来恢复该 iPad 然后才能将它与 iTunes 配合使用。”
ECID	每台 iOS 设备上的处理器所独有的一个 64 位标识符。作为个性化流程的一部分，此标识符不被视为机密。
可擦除存储器	NAND 存储器中一个用于储存加密密钥的专用区域，可被直接寻址和安全擦除。尽管当攻击者实际占有设备时，可擦除存储器无法提供保护，但其中存储的密钥可用作密钥层次结构的一部分，用于实现快速擦除和提高安全性。
文件系统密钥	用于加密每个文件的元数据的密钥，包括其类密钥。存储在可擦除存储器中，用于实现快速擦除，而不被视为机密。
设备组 ID (GID)	类似于 UID，但同一类中的每个处理器的 GID 都相同。
硬件安全模块 (HSM)	专门的防篡改计算机，保障数据密钥的安全并对其进行管理。
iBoot	由 LLB 加载的代码，并随后加载 XNU，作为安全启动链的一部分。
身份识别服务 (IDS)	Apple 的 iMessage 公钥、APNs 地址和电话号码及电子邮件地址目录，用于查找密钥和设备地址。
集成电路 (IC)	也被称为微芯片。
联合测试行动小组 (JTAG)	程序员和电路开发者所采用的标准硬件调试工具。
密钥包	一种用于储存一组类密钥的数据结构。每种类型（系统、备份、托管或 iCloud 云备份）的格式都相同： • 包含以下内容的标头： – 版本（在 iOS 5 中设置为 3） – 类型（系统、备份、托管或 iCloud 云备份） – 密钥包 UUID – 密钥包签名后的 HMAC – 用于封装类密钥的方法：与 UID 或 PBKDF2 以及盐密钥和迭代次数配合使用 • 类密钥列表： – 密钥 UUID – 类（所属的文件或钥匙串数据保护类） – 封装类型（仅 UID 派生密钥；UID 派生密钥和密码派生密钥） – 封装的类密钥 – 非对称类的公钥
钥匙串	一种基础架构和一组 API，iOS 和第三方应用用来储存和检索密码、密钥及其他敏感凭证。
密钥封装	使用一个密钥来加密另一个密钥。iOS 按照 RFC 3394 使用 NIST AES 密钥封装。
底层引导加载程序 (LLB)	Boot ROM 调用的代码，之后会加载 iBoot，成为安全启动链的一环。

文件独有密钥	用于加密文件系统中文件的 AES 256 位密钥。文件独有密钥使用类密钥封装，储存在文件的元数据中。
预置描述文件	Apple 签名的 plist，其中列明允许在 iOS 设备上安装和测试应用的实体和授权。开发预置描述文件列出开发人员选择用于点对点分配的设备，分配预置描述文件中包含企业开发的应用的应用 ID。
纹路走向角度映射	一种提取自指纹的一部分、描述纹路走向和宽度的数学表达式。
智能卡	内置的集成电路，用于提供安全识别、认证和数据储存。
系统芯片 (SoC)	一种将多种组件整合到单个芯片上的集成电路 (IC)。Secure Enclave 是 Apple 的 A7 中央处理器或新款中央处理器中的 SoC。
Tangling	用户密码转换为密钥并使用设备的 UID 加强的过程。此举可确保暴力攻击只能在特定设备上执行，因此发生攻击的概率降低且可避免多部设备同时遭到攻击。Tangling 算法是 PBKDF2。这种算法为每次迭代使用 AES 加密的设备 UID 作为伪随机函数 (PRF)。
统一资源标识符 (URI)	可识别基于 Web 的资源的字符串。
唯一 ID (UID)	一个 256 位的 AES 密钥，在生产中刻录在每个处理器上。这种密钥无法由硬件或软件读取，只能由处理器的硬件 AES 引擎使用。若要获取实际密钥，攻击者必须对处理器的芯片发起极为复杂且代价高昂的物理攻击。UID 与设备上的任何其他标识符均无关，包括但不限于 UDID。
XNU	iOS 和 OS X 操作系统中央的内核。默认为受信任状态，并强制执行代码签名、沙盒化、授权核对和 ASLR 等安全措施。

文稿修订历史

日期	摘要
2015 年 9 月	**已更新适用于 iOS 9** • Apple Watch 激活锁 • 密码策略 • Touch ID API 支持 • A8 上数据保护使用 AES-XTS • 适用于无人值守式软件更新的密钥包 • 认证更新 • 企业级应用信任模型 • 对于 Safari 书签的数据保护 • 应用传输安全性 • VPN 规格 • 针对 HomeKit 的 iCloud 远程访问 • Apple Pay 回馈卡 • Apple Pay 发卡机构的应用 • Spotlight 设备上索引 • iOS 配对模型 • Apple Configurator • 访问限制 • 有关 iOS 9 安全性内容的更多信息，请参阅： support.apple.com/zh-cn/HT205212