**iOS & Android security**

**Android security**

**Security**

Android incorporates industry-leading security features and works with developers and device implementers to keep the Android platform and ecosystem safe. A robust security model is essential to enable a vigorous ecosystem of applications and devices built on and around the Android platform and supported by cloud services. As a result, through its entire development lifecycle, Android has been subject to a rigorous security program.

* **Android is designed:** 
  + To Be Open: Android was designed with multi-layered security that is flexible enough to support an open platform while still protecting all users of the platform.
* For Developers: Developers less familiar with security will be protected by safe defaults. In addition to providing a stable platform to build upon, Android gives additional support to developers in a number of ways.
* For Users: Users are provided visibility into permissions requested by each application and control over those permissions. This design includes the expectation that attackers would attempt to perform common attacks

## Linux Security

The foundation of the Android platform is the Linux kernel. The Linux kernel has been in widespread use for years, and is used in millions of security-sensitive environments. Through its history of constantly being researched, attacked, and fixed by thousands of developers, Linux has become a stable and secure kernel trusted by many corporations and security professionals.

As the base for a mobile computing environment, the Linux kernel provides Android with several key security features, including:

* A user-based permissions model
* Process isolation
* Extensible mechanism for secure IPC
* The ability to remove unnecessary and potentially insecure parts of the kernel

As a multiuser operating system, a fundamental security objective of the Linux kernel is to isolate user resources from one another. The Linux security philosophy is to protect user resources from one another. Thus, Linux:

* Prevents user A from reading user B's files
* Ensures that user A does not exhaust user B's memory
* Ensures that user A does not exhaust user B's CPU resources
* Ensures that user A does not exhaust user B's devices (e.g. telephony, GPS, Bluetooth)
* **Security tips**

## The Application Sandbox

Android’s application security is enforced by the application sandbox, which isolates apps from each other and protects apps and the system from malicious apps

## Filesystem Permissions

In a UNIX-style environment, filesystem permissions ensure that one user cannot alter or read another user's files. In the case of Android, each application runs as its own user. Unless the developer explicitly shares files with other applications, files created by one application cannot be read or altered by another application.

## Cryptography

* is associated with the process of converting ordinary plain text into unintelligible text and vice-versa.
* It is a method of storing and transmitting data in a particular form .
* not only protects data from theft or alteration, but can also be used for user authentication

## Interprocess Communication

Processes can communicate using any of the traditional UNIX-type mechanisms. Examples include the filesystem, local sockets, or signals. However, the Linux permissions still apply.

* In software systems it is often the case that different processes need to communicate with one another in order to cooperate on some task.
* The Android IPC mechanisms are designed to operate at the level of components rather than processes.
* In other words a component in one app may use IPC to communicate with a component in another app, or with another component in the same app**.**

**Difference between encryption and cryptology?**

* Earlier cryptography was effectively synonymous with encryption but nowadays cryptography is mainly based on mathematical theory and computer science practice.
* **Encryption**
* So, firstly we are now going to check what are the advantages of having encrypted Android phone.
  + It is very easy to do
  + Do not have to install any extra app
  + App data also get encrypted
  + Data Security Increases
  + Secure your phone from being retrieved

**Cons:**

* + Not available on every Android phone
  + It cannot be done using any app
  + It’s a very lengthy taking process
  + It cannot be revered
  + Decreases your phone performance

**System Partition and Safe Mode:**

* The system partition contains Android's kernel as well as the operating system libraries, application runtime, application framework, and applications. This partition is set to read-only. When a user boots the device into Safe Mode, third-party applications may be launched manually by the device owner but are not launched by default.

**Apps security :**

## Cost-Sensitive APIs

A cost sensitive API is any function that might generate a cost for the user or the network. The Android platform has placed cost sensitive APIs in the list of protected APIs controlled by the OS. The user will have to grant explicit permission to third-party applications requesting use of cost sensitive APIs. These APIs include:

* Telephony
* SMS/MMS
* Network/Data
* In-App Billing
* NFC Access

Android 4.2 adds further control on the use of SMS. Android will provide a notification if an application attempts to send SMS to a short code that uses premium services which might cause additional charges. The user can choose whether to allow the application to send the message or block it.

## SIM Card Access

Low level access to the SIM card is not available to third-party apps. The OS handles all communications with the SIM card including access to personal information (contacts) on the SIM card memory. Applications also cannot access AT commands, as these are managed exclusively by the Radio Interface Layer (RIL). The RIL provides no high level APIs for these commands.

## Personal Information

Android has placed APIs that provide access to user data into the set of protected APIs. With normal usage, Android devices will also accumulate user data within third-party applications installed by users. Applications that choose to share this information can use Android OS permission checks to protect the data from third-party applications.

## Sensitive Data Input Devices

Android devices frequently provide sensitive data input devices that allow applications to interact with the surrounding environment, such as camera, microphone or GPS. For a third-party application to access these devices, it must first be explicitly provided access by the user through the use of Android OS Permissions. Upon installation, the installer will prompt the user requesting permission to the sensor by name.

If an application wants to know the user's location, the application requires a permission to access the user's location. Upon installation, the installer will prompt the user asking if the application can access the user's location. At any time, if the user does not want any application to access their location, then the user can run the "Settings" application, go to "Location & Security", and uncheck the "Use wireless networks" and "Enable GPS satellites". This will disable location based services for all applications on the user's device.

## Device Metadata

Android also strives to restrict access to data that is not intrinsically sensitive, but may indirectly reveal characteristics about the user, user preferences, and the manner in which they use a device.

By default applications do not have access to operating system logs, browser history, phone number, or hardware / network identification information. If an application requests access to this information at install time, the installer will prompt the user asking if the application can access the information. If the user does not grant access, the application will not be installed.

**Attacks:**

Attacks can be classified into two types :

1 – Passive (For Reconnaissance)

2 – Active (For Exploitation)

**The difference between active and passive attacks:**

in case of active attack involve the updating upon the data means the active attack access the data and then perform alteration upon the data and then data transmit on the network but in case of passive attack the attacker just access the message and the contents of the message without any alteration upon the data means just type attack just involve to read and access the message the content mean just observe the message.

The active attack easy to detect but not easy to protect but passive attack difficult to detect easy to prevent

**Examples :**

1 – Remote Access Trojans

* Enables remote administrative control for Reconnaissance and attack.
* Monitor user behavior through key loggers and spywares.
* Activate webcam and record audio.
* Format drives
* Obtain access / login information of the victim.

**2 – Network attack**

* Occurs when an android device is connected to wifi or a malicious hotspot

**3 – Malicious file attack**

* Is a piece of code that is execute behind the scenes as an “Exploit” to some vulnerabilities in the victim’s system.
* This allows attackers to perform Remote code execution
* Such as : patches , fake links , encrypted text files and Fake images.

**4 – Root Permissions**

* on Android only the kernel and a small subset of the core applications run with root permissions. Android does not prevent a user or application with root permissions from modifying the operating system, kernel, or any other application.
* In general, root has full access to all applications and all application data.
* Users that change the permissions on an Android device to grant root access to applications increase the security exposure to malicious applications and potential application flaws

**iOS Security**

**Introduction**

Apple designed the iOS platform with security at its core. When we set out to create the best possible mobile platform, we drew from decades of experience to build an entirely new architecture. We thought about the security hazards of the desktop environment, and established a new approach to security in the design of iOS. We developed and incorporated innovative features that tighten mobile security and protect the entire system by default. As a result, iOS is a major leap forward in security for mobile devices.

Every iOS device combines software, hardware, and services designed to work together for maximum security and a transparent user experience. iOS protects not only the device and its data at rest, but the entire ecosystem, including everything users do locally, on networks, and with key internet services.

**System Security**

System security is designed so that both software and hardware are secure across all core components of every iOS device. This includes the boot-up process, software updates, and Secure Enclave. This architecture is central to security in iOS, and never gets in the way of device usability.

**-Secure boot chain**

Each step of the startup process contains components that are cryptographically signed by Apple to ensure integrity and that proceed only after verifying the chain of trust. This includes the bootloaders, kernel, kernel extensions, and baseband firmware. This secure boot chain helps ensure that the lowest levels of software aren’t tampered with.

When an iOS device is turned on, its application processor immediately executes code from read-only memory known as Boot ROM. This immutable code, known as the hardware root of trust, is laid down during chip fabrication, and is implicitly trusted. The Boot ROM code contains the Apple Root CA public key, which is used to verify that the iBoot bootloader is signed by Apple before allowing it to load. This is the first step in the chain of trust where each step ensures that the next is signed by Apple. When the iBoot finishes its tasks, it verifies and runs the iOS kernel. For devices with an A9 or earlier A-series processor, an additional Low-Level Bootloader (LLB) stage is loaded and verified by the Boot ROM and in turn loads and verifies iBoot.

A failure of the Boot ROM to load LLB (on older devices) or iBoot (on newer devices) results in the device entering DFU mode. In the case of a failure in LLB or iBoot to load or verify the next step, startup is halted and the device displays the connect to iTunes screen. This is known as recovery mode. In either case, the device must be connected to iTunes through USB and restored to factory default settings. The Boot Progress Register (BPR) is used by the Secure Enclave to limit access to user data in different modes and is updated before entering the following modes:

• Recovery Mode: Set by iBoot on devices with Apple A10, S2, and newer system on chip (SoCs)

• DFU Mode: Set by Boot ROM on devices with an A12 SoC

**-System Software Authorization**

Apple regularly releases software updates to address emerging security concerns and also provide new features; these updates are provided for all supported devices simultaneously. Users receive iOS update notifications on the device and through iTunes, and updates are delivered wirelessly, encouraging rapid adoption of the latest security fixes.

On a device with Secure Enclave, the Secure Enclave coprocessor also utilizes System Software Authorization to ensure the integrity of its software and prevent downgrade installations.

iOS software updates can be installed using iTunes or over the air (OTA) on the device. With iTunes, a full copy of iOS is downloaded and installed. OTA software updates download only the components required to complete an update, improving network efficiency, rather than downloading the entire OS. Additionally, software updates can be cached on a Mac running macOS High Sierra with Content Caching turned on, so that iOS devices don’t need to redownload the necessary update over the Internet. They’ll still need to contact Apple servers to complete the update process.

The boot-time chain-of-trust evaluation verifies that the signature comes from Apple and that the measurement of the item loaded from disk, combined with the device’s ECID, matches what was covered by the signature.

These steps ensure that the authorization is for a specific device and that an old iOS version from one device can’t be copied to another. The nonce prevents an attacker from saving the server’s response and using it to tamper with a device or otherwise alter the system software.

**-Touch ID**

Touch ID is the fingerprint sensing system that makes secure access to iPhone and iPad faster and easier. This technology reads fingerprint data from any angle and learns more about a user’s fingerprint over time, with the sensor continuing to expand the fingerprint map as additional overlapping nodes are identified with each use.

**-Face ID**

With a simple glance, Face ID securely unlocks Apple devices that have that feature. It provides intuitive and secure authentication enabled by the TrueDepth camera system, which uses advanced technologies to accurately map the geometry of your face. Face ID uses neural networks for determining attention, matching, and anti-spoofing, so you can unlock your phone with a glance. Face ID automatically adapts to changes in your appearance, and carefully safeguards the privacy and security of your biometric data.

**-Touch ID, Face ID, and passcodes**

To use Touch ID or Face ID, you must set up your device so that a passcode is required to unlock it. When Touch ID or Face ID detects a successful match, your device unlocks without asking for the device passcode.

This makes using a longer, more complex passcode far more practical because you don’t need to enter it as frequently. Touch ID and Face ID don’t replace your passcode, but provide easy access to your device within thoughtful boundaries and time constraints. This is important because a strong passcode forms the foundation for how your iOS device cryptographically protects your data. You can use your passcode anytime instead of Touch ID or Face ID, but the following operations always require a passcode instead of a biometric:

• Updating your software

. • Erasing your device.

• Viewing or changing passcode settings.

• Installing iOS configuration profiles.

A passcode is also required if your device is **:**

• The device has just been turned on or restarted.

• The device hasn’t been unlocked for more than 48 hours.

• The passcode hasn’t been used to unlock the device in the last 156 hours (six and a half days) and a biometric hasn’t unlocked the device in the last 4 hours.

• The device has received a remote lock command.

• After five unsuccessful biometric match attempts.

• After initiating power off/Emergency SOS.

**Network Security**

In addition to the built-in safeguards Apple uses to protect data stored on iOS devices, there are many network security measures that organizations can take to keep information secure as it travels to and from an iOS device. Mobile users must be able to access corporate networks from anywhere in the world, so it’s important to ensure that they are authorized and their data is protected during transmission. iOS uses—and provides developer access to— standard networking protocols for authenticated, authorized, and encrypted communications. To accomplish these security objectives, iOS integrates proven technologies and the latest standards for both Wi-Fi and cellular data network connections.

**-TLS**

iOS supports Transport Layer Security (TLS v1.0, TLS v1.1, TLS v1.2) and DTLS. It supports both AES-128 and AES-256, and prefers cipher suites with perfect forward secrecy. Safari, Calendar, Mail, and other Internet apps automatically use this protocol to enable an encrypted communication channel between the device and network services.

**-Bluetooth**

Bluetooth support in iOS has been designed to provide useful functionality without unnecessary increased access to private data. iOS devices support Encryption Mode 3, Security Mode 4, and Service Level 1 connections. iOS supports the following Bluetooth profiles:

• Hands-Free Profile (HFP)

• Phone Book Access Profile (PBAP)

• Message Access Profile (MAP)

• Advanced Audio Distribution Profile (A2DP)

• Audio/Video Remote Control Profile (AVRCP)

• Personal Area Network Profile (PAN)

• Human Interface Device Profile (HID)

**- AirDrop security**

iOS devices that support AirDrop use Bluetooth Low Energy (BLE) and Applecreated peer-to-peer Wi-Fi technology to send files and information to nearby devices, including AirDrop-capable Mac computers running OS X 10.11 or later. The Wi-Fi radio is used to communicate directly between devices without using any Internet connection or Wi-Fi Access Point.

When a user chooses AirDrop as the method for sharing an item, the device emits an AirDrop signal over Bluetooth Low Energy. Other devices that are awake, in close proximity, and have AirDrop turned on detect the signal and respond with a shortened version of their owner’s identity hash. AirDrop is set to share with Contacts Only by default. Users can also choose to use AirDrop to share with everyone, or turn off the feature entirely. In Contacts Only mode, the received identity hashes are compared with hashes of people in the initiator’s Contacts app. If a match is found, the sending device creates a peer-to-peer Wi-Fi network and advertises an AirDrop connection using Bonjour. Using this connection, the receiving devices send their full identity hashes to the initiator. If the full hash still matches Contacts, the recipient’s first name and photo (if present in Contacts) are displayed in the AirDrop share sheet.

**-Internet Services**

Apple has built a robust set of services to help users get even more utility and productivity out of their devices, including iMessage, FaceTime, Siri Suggestions, iCloud, iCloud Backup, and iCloud Keychain. These Internet services have been built with the same security goals that iOS promotes throughout the platform. These goals include secure handling of data, whether at rest on the device or in transit over wireless networks; protection of users’ personal information; and threat protection against malicious or unauthorized access to information and services. Each service uses its own powerful security architecture without compromising the overall ease of use of iOS.

**-Apple ID**

An Apple ID is the account that is used to sign in to Apple services such as iCloud, iMessage, FaceTime, the iTunes Store, Apple Books, the App Store, and more. It is important for users to keep their Apple IDs secure to prevent unauthorized access to their accounts. To help with this, Apple requires strong passwords that must be at least eight characters in length, contain both letters and numbers, must not contain more than three consecutive identical characters, and can’t be a commonly used password. Users are encouraged to exceed these guidelines by adding extra characters and punctuation marks to make their passwords even stronger. Apple also requires users to set up three security questions that can be used to help verify the owner’s identity when making changes to their account information or resetting a forgotten password.

**-Two-factor authentication**

To help users further secure their accounts, Apple offers two-factor authentication—an extra layer of security for Apple IDs. It is designed to ensure that only the account’s owner can access the account, even if someone else knows the password. With two-factor authentication, a user’s account can be accessed only on trusted devices, such as the user’s iPhone, iPad, or Mac. To sign in for the first time on any new device, two pieces of information are required—the Apple ID password and a six-digit verification code that’s automatically displayed on the user’s trusted devices or sent to a trusted phone number. By entering the code, the user verifies that they trust the new device and that it’s safe to sign in.

Because a password alone is no longer enough to access a user’s account, two-factor authentication improves the security of the user’s Apple ID and all the personal information they store with Apple. It is integrated directly into iOS, macOS, tvOS, watchOS, and the authentication systems used by Apple’s websites.

**-Two-step verification**

Since 2013, Apple has also offered a similar security method called two-step verification. With two-step verification enabled, the user’s identity must be verified via a temporary code sent to one of the user’s trusted devices before changes are permitted to their Apple ID account information; before signing in to iCloud, iMessage, FaceTime, or Game Center; and before making an iTunes Store, Apple Books, or App Store purchase from a new device. Users are also provided with a 14-character Recovery Key to be stored in a safe place in case they ever forget their password or lose access to their trusted devices. While most new users will be encouraged to use two-factor authentication, there are still some situations where two-step verification is recommended instead.

**-Managed Apple IDs**

Managed Apple IDs function in a way similar to an Apple ID, but are owned and controlled by an educational institution. The institution can reset passwords, limit purchasing and communications such as FaceTime and Messages, and set up role-based permissions for staff members, teachers, and students. Some Apple services are disabled for Managed Apple IDs, such as Apple Pay, iCloud Keychain, HomeKit, and Find My iPhone.

**-iMessage**

Apple iMessage is a messaging service for iOS devices, Apple Watch, and Mac computers. iMessage supports text and attachments such as photos, contacts, and locations. Messages appear on all of a user’s registered devices so that a conversation can be continued from any of the user’s devices. iMessage makes extensive use of the Apple Push Notification service (APNs). Apple doesn’t log the contents of messages or attachments, which are protected by end-to-end encryption so no one but the sender and receiver can access them. Apple can’t decrypt the data.

When a user turns on iMessage on a device, the device generates two pairs of keys for use with the service: an RSA 1280-bit key for encryption and an ECDSA 256-bit key on the NIST P-256 curve for signing. The private keys for both key pairs are saved in the device’s Keychain and the public keys are sent to Apple identity service (IDS), where they are associated with the user’s phone number or email address, along with the device’s APNs address.

As users enable additional devices for use with iMessage, their encryption and signing public keys, APNs addresses, and associated phone numbers are added to the directory service. Users can also add more email addresses, which are verified by sending a confirmation link. Phone numbers are verified by the carrier network and SIM. With some networks, this requires using SMS (the user will be presented with a confirmation dialog if the SMS is not zero rated). Phone number verification may be required for several system services in addition to iMessage, such as FaceTime and iCloud. All of the user’s registered devices display an alert message when a new device, phone number, or email address is added.

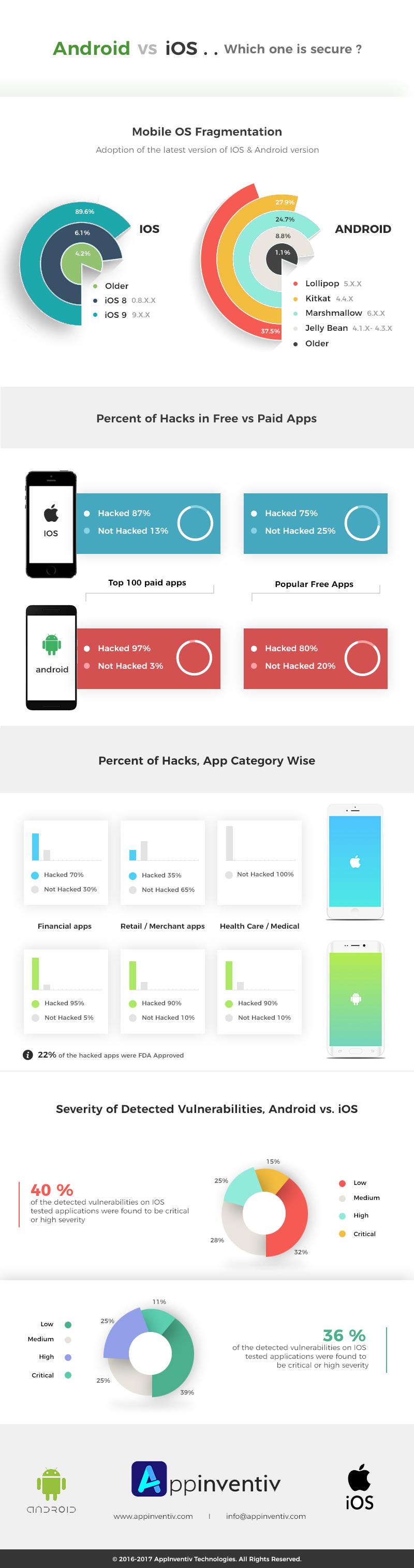
**-FaceTime**

-FaceTime is Apple’s video and audio calling service. Similar to iMessage, FaceTime calls also use the Apple Push Notification service to establish an initial connection to the user’s registered devices. The audio/video contents of FaceTime calls are protected by end-to-end encryption, so no one but the sender and receiver can access them. Apple can’t decrypt the data. The initial FaceTime connection is made through Apple server infrastructure that relays data packets between the users’ registered devices. Using APNs notifications and Session Traversal Utilities for NAT (STUN) messages over the relayed connection, the devices verify their identity certificates and establish a shared secret for each session. The shared secret is used to derive session keys for media channels streamed via the Secure Real-time Transport Protocol (SRTP). SRTP packets are encrypted using AES-256 in Counter Mode and HMAC-SHA1. Subsequent to the initial connection and security setup, FaceTime uses STUN and Internet Connectivity Establishment (ICE) to establish a peerto-peer connection between devices, if possible.

-**iCloud**

iCloud stores a user’s contacts, calendars, photos, documents, and more, and keeps the information up to date across all of their devices, automatically. iCloud can also be used by third-party apps to store and sync documents as well as key values for app data as defined by the developer. Users set up iCloud by signing in with an Apple ID and choosing which services they would like to use. iCloud features, including My Photo Stream, iCloud Drive, and iCloud Backup, can be disabled by IT administrators via MDM configuration profiles. The service is agnostic about what is being stored and handles all file content the same way, as a collection of bytes.

**Difference between android and IOS:**

****

