# iOS Security

**Bingkun Guo**, guobingkun (at) wustl.edu (A paper written under the guidance of Prof. Raj Jain) Download

## Abstract:

iOS has been a very advanced and sophisticated mobile operating system ever since it was first released in 2007. In this survey paper, we will first focus on introducing iOS security by talking about the implementation details of its essential building blocks, such as system security, data security, hardware security and app security. Then we will talk about some potential and existing security issues of iOS.

## Keywords:

iOS, Apple, System Security, Data Security, Application Security, Network Security, Internet Service, ApplePay, Encryption, Privacy

## Table of Contents:

---

## 1. Introduction

iOS has been one of the most popular mobile operating system in the world ever since it was first released in 2007. As of June 2014, Appleâ€™s App Store contained more than 1.2 million iOS applications, which have collectively been downloaded more than 60 billion times. [Wiki 05]

iOS was designed and created by Apple Inc, it is distributed exclusively for Apple hardware. iOS protects not only the data stored in the iOS device, but also the data transmitted on networks when using internet services. iOS provides advanced and sophisticated security for iOS devices and itâ€™s also very easy to use. Users donâ€™t need to spend a lot of time on security configurations, as most of the security features have been automatically configured by iOS. iOS also supports biometric authentication (Touch ID), which has recently been incorporated into iOS devices, users can easily use their fingerprints to perform private and sensitive tasks such as unlocking the iPhone and making payments.

This survey talks about the details about how the security elements are implemented in iOS and some issues with the iOS security.

## 2. iOS System Security

System security is central to security in iOS. It makes the hardware and software securely integrated with each other such that every component in iOS is secure and trusted. Fig.1 is a high-level overview of iOS security architecture, the details are explained in the following sections.

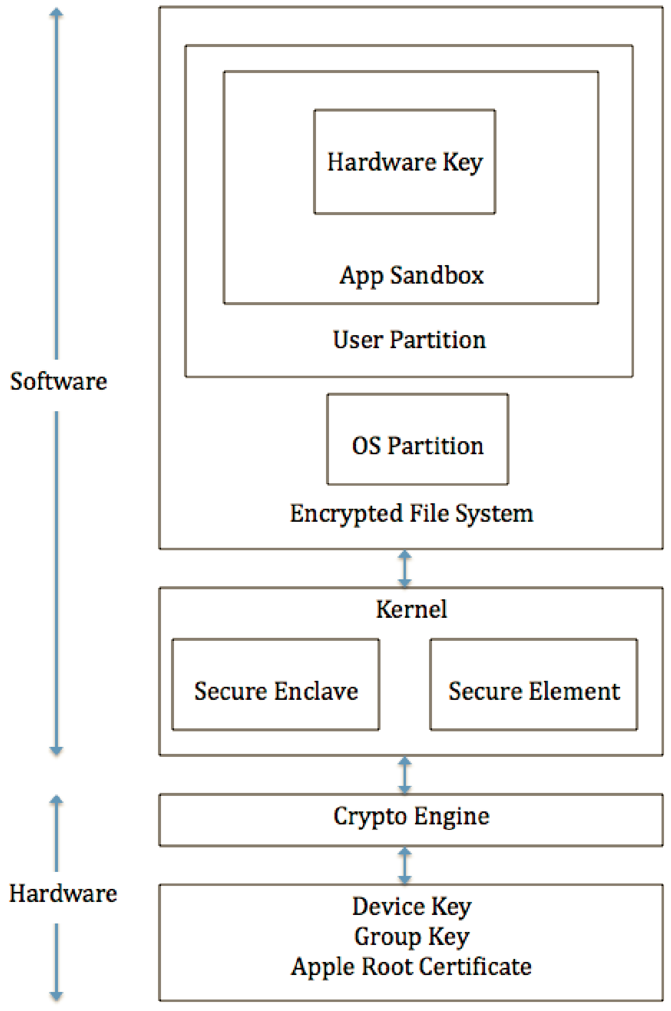


**Fig.1 Security architecture diagram of iOS**

### 2.1 Secure booting process

During the booting process, iOS uses a mechanism called "secure boot chain" to ensure that the low-level software is not compromised and iOS is running on a validated iOS device. Each step in secure boot chain verifies if the next step of chain is valid and signed by Apple. The booting process will only proceed to the next step of chain if the verification succeeds.

When you turn on an iOS device, the processor first executes the code from Boot ROM (read-only-memory). The code in Boot ROM is created during chip fabrication, hence it is trusted and immutable. The code in Boot ROM also contains the Apple Root CA public key, which will be used to verify if Low-Level Bootloader (LLB) is signed by Apple. If LLB is valid, the processor will run the next-stage bootloader, iBoot, which will in turn verify and run the iOS kernel.

### 2.2 Secure Enclave

The Secure Enclave is a coprocessor for Appleâ€™s A-series processor. It has its own secure boot separated from the application processor, communication between it and the application processor is highly encapsulated. Its tasks include key management, processing cryptographic operations and maintaining the data integrity.

Each Secure Enclave comes with a unique ID (UID) during the fabrication. Other parts of the system donâ€™t have access to UID, neither does Apple. UID is used to encrypt the Secure Enclaveâ€™s memory space and data of files stored in the file system.

The Secure Enclave is also responsible for decrypting and processing the fingerprints received from the Touch ID, verifying if the coming fingerprints match the registered fingerprints. The application processor forwards the fingerprints data to the Secure Enclave. Because the fingerprints data is encrypted with a session key between the Secure Enclave and the Touch ID, the application processor canâ€™t read it.

### 2.3 Touch ID Security

Touch ID is a fingerprints sensor that can read fingerprints from the user. A user who passes the fingerprints verification can have secure access to the device, such as unlocking the iOS device, making purchases from the App Store, and making secure payment through Apple Pay (More information in the Apple Pay section).

When the user touches the home button, the capacitive steel ring on the home button detects the finger and activates the Touch ID sensor. Then Touch ID scans the fingerprints and sends a 88-by-88-pixel, 500-ppi raster scan to the Secure Enclave for authentication. The scan is vectorized for analysis and temporarily stored in encrypted memory in the Secure Enclave. After authentication, it is discarded.

## 3. Data Security

Besides making sure that only trusted code and apps can run on the devices, iOS also encrypts and protects userâ€™s data locally and remotely.

### 3.1 Hardware security features

Because cryptographic operations are complicated and CPU consuming, they could introduce battery life problems if they are not well implemented.

Every iOS device has a dedicated AES-256 crypto engine built into the Directed Memory Access (DMA) path between the flash storage and main system memory, hence making the encryption process highly efficient.

Every device also has a unique UID and a device group ID (GID), which are 256-bit keys. They are created and stored in the application processor during fabrication, no software or hardware can access them directly. The UID is unique to each device while devices that have the common processors have the same GID. Data encrypted using the UID is tied to a particular device, hence if the memory chip is physically moved to another device, the encrypted files will not be accessible. [Apple 01]

Except the UID and GID, all other keys used for encryption are created by the iOSâ€™s random number generator (RNG) using Counter mode Deterministic Random Byte Generator (CTR-DRBG).

### 3.2 File Data Protection

iOS protects the file data by constructing and managing a hierarchy of keys in conjunction with hardware encryption engine. Fig.2 depicts this hierarchy and relations between the keys. When a file is created, the Data Protection system creates a 256-bit key (â€œper-fileâ€ key) and forwards it to the hardware AES engine, which will use the per-file key to encrypt the file.

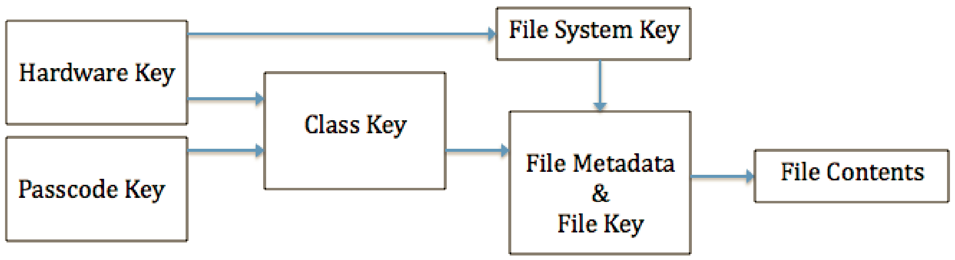


**Fig.2 Hierarchy of iOS file encryption keys**

As shown in Fig.2, the per-file key is wrapped with corresponding class keys. The key wrapping process is implemented using NIST AES key wrapping. The resulting wrapped per-file key is stored in the fileâ€™s metadata. To open a file, iOS first decrypts the fileâ€™s meta data using the file system key, having the wrapped per-file key and classes which are used to wrap the per-file key. Then iOS unwraps the per-file key using the corresponding class keys, and forwards the per-file key to the hardware AES engine. Then the AES engine decrypts the file contents read from the flash storage.

### 3.3 Passcodes

Passcode is an important element to iOS security. By setting up a passcode, Data Protection is automatically enabled by iOS. Most people think that passcode is just a 4 digit passcode for unlocking the iOS devices. Nevertheless, it can be set as an arbitrary length alphanumeric passcode. More importantly, the passcode is also used for generating encryption keys. Hence the stronger your passcode is, the stronger encryption keys are generated.

### 3.4 Data Protection Class

When a file is created on the iOS device, it is assigned a class by the app that creates it. A class is used to specify the policies on when the data is accessible. Some examples of typical classes are Complete Protection, Protected Unless Open, Protected Until First User Authentication, No Protection.

### 3.5 Keychain Security

iOS apps deal with a lot of sensitive data, such as passwords. In order to store these passwords securely, iOS keychain is used.

An keychain item contains metadata, such as the creation/modification timestamps, and the access group of this keychain. Besides, a keychain contains the version number, the access control list, a value indicating which protect class the item is in, a per-item key wrapped with the protection class key and a dictionary of attributes describing the item. All these contents are encrypted using AES 128 in Galois/Counter Mode (GCM).

## 4. App Security

Applications are one of the most important elements in iOS. As of June 2014, Appleâ€™s App Store contained more than 1.2 million iOS applications, which have collectively been downloaded more than 60 billion times. While apps bring users incredible productivity and pleasing user experience, the existence of unauthorized malware is still a big threat to iOS security.

To make sure all the apps running on iOS are not performing malicious tasks, Apple encores that all the apps must be reviewed and approved by Apple before being available on the App Store.

### 4.1 App review and code signing

In order for an application to be installed on iOS devices, Apple requires that all executable code be signed by Apple issued certificate. This requirement significantly reduces the threats of malware and filters out buggy apps.

In order for developers to develop and sell their apps on the App Store, developers must register and join the Apple Developer Program, obtaining a verified developer certificate. Developers must use this certificate to submit their apps to App Store, after being reviewed and approved by Apple, the apps will be available on the App Store.

### 4.2 Runtime process security

iOS uses a mechanism called Sandbox to prevent apps from compromising other apps installed on the devices. Apps are restricted from accessing files associated with other apps and making secure changes to the device. In addition, 3rd-party apps have limited resources when performing background tasks. iOS allows only system provided APIs to be called by 3rd-party apps in background. [Al-Qershi 02]

### 4.3 Data Protection in Apps

iOS Software Development Kit (SDK) provides a lot of APIs such that developers can easily write Data Protection code. Examples of these APIs are NSFileManager, CoreData, NSData, SQLite.

## 5. Network and Internet Services Security

There are a lot of ongoing network activities while people are using their iOS devices. To ensure that the data and userâ€™s privacy are protected during the transmission over network. iOS incorporates lots of advanced technologies and the latest standard network protocols.

### 5.1 SSL, TLS

iOS supports Secure Socket Layer (SSL v3) as well as Transport Layer Security (TLS v1.0, TLS v1.1, TLS v1.2). Built-in apps such as Safari, Mail use these protocols securely communicate with remote servers. iOS also provides APIs such as CFNetwork and SecureTransport so that developers could easily set up and maintain a secure SSL or TLS networking session, though the details of implementation are not open to public.

### 5.2 Wifi

For Wi-Fi, iOS supports WPA2 Enterprise, which uses 128-bit AES encryption. Also, iOS supports 802.1X and Remote Authentication Dial In User Service (RADIUS), the following 802.1X wireless authentication methods are supported: EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0, PEAPv1, and LEAP. In addition, a randomized Media Access Control (MAC) address is used when iOS is performing Preferred Network Offload (PNO) scans. PNO scans happen when an iOS device is not connected to a Wi-Fi network and its processor is asleep, then PNO scans decide if the device can connect to a preferred Wi-Fi network. Because the MAC address changes, it is impossible for an attacker or an observer to track an iOS deviceâ€™s traffic, which brings much more security.[Apple 01]

### 5.3 AirDrop security

AirDrop allows users to share files on their iOS or OS X devices with nearby users. When a user enables AirDrop, a 2048-bit RSA identity and its hash are created and stored on the device. When AirDrop is open, a signal is emitted through Bluetooth Low Energy such that nearby devices that also have AirDrop turned on can receive it. After the sender chooses to whom he/she wants to send, a TLS connection is created between the sender and the receiver, with iCloud identity certificates being exchanged. After the receiver accepts the files to transfer, the transmission begins.

### 5.4 iMessage Security

iMessage is a very popular message service provided by Apple. Users can communicate with each other through iMessage as long as one of them is using an iOS or OS X devices. Now iMessage even supports receiving SMS text messages on an OS X device. Besides text, users can also send attachments such as photos and documents. iMessage extensively uses Apple Push Notification service (APNs), which is responsible for propagating information to iOS and OS X devices in a secure way. [Apple 14]

When iMessage is turned on, an RSA 1280-bit key for encryption and an Elliptic Curve Digital Signature Algorithm (ECDSA) 256-bit key for signing are generated. The private keys for those two keys are stored in the deviceâ€™s keychain. The public keys are sent to Appleâ€™s directory service, which is responsible for storing userâ €™s identification, his/her associated public keys and devicesâ€™ APNs addresses.

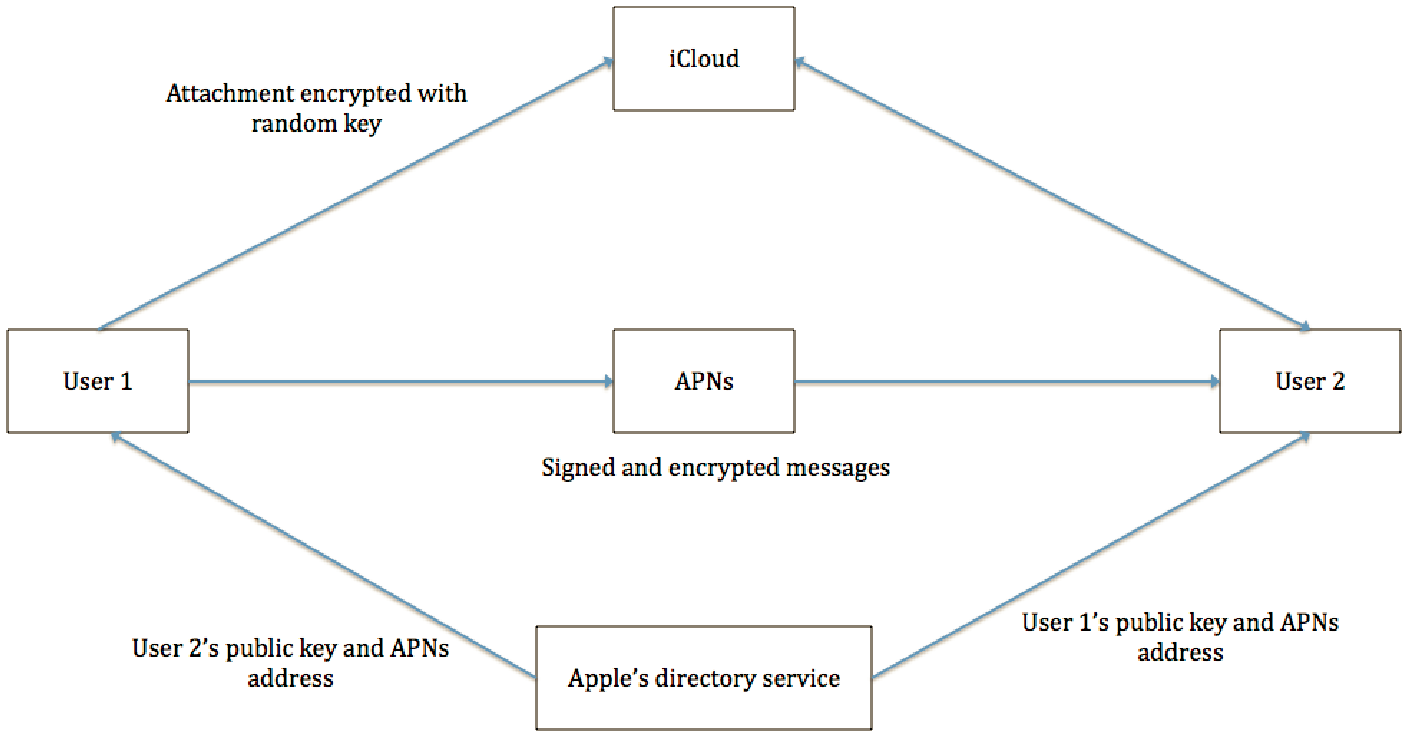

iCloud
Attachment encrypted with random key
User 1
APNs
User 2
Signed and encrypted messages
User 2's public key and APNs address
Apple's directory service
User 1's public key and APNs address

**Fig.3 iMessage flowchart**

As depicted in Fig.3, when the user starts a new iMessage conversation, after entering the email address/phone number or the name of the receiver, iOS will query Appleâ€™s directory service for the receiver's public keys and APNs addresses (A user can sign in iMessage on multiple devices). The message sent by the sender is encrypted by AES-128 in CTR mode for each device of the receiver, and it is signed using the senderâ€™s private key. Then the message will be pushed to the APNs with unencrypted metadata, such as the timestamp and APNs routing information. If the message is too long, for example, a large size attachment. Then instead of pushing it to the APNs, the attachment will be encrypted with a random key and uploaded to iCloud, then the receiver will fetch that attachment from iCloud.

During the transmission, the message is delivered as a copy from APNs to the receiver, once the message is successfully delivered, APNs will delete it.

### 5.5 FaceTime Security

Users can make video calls on iOS or OS X devices with others using FaceTime. It establishes an end-to-end connection between correspondents using Session Initiation Protocol. The contents of the communication is encrypted such that only the sender and receiver can decrypt them.

### 5.6 iCloud Security

iCloud is used to store contacts, photos, calendar and other documents such that they are synchronized on all of the userâ€™s iOS devices. Recently, iCloud drive is released so that users can upload any kind files to iCloud. iCloud keychain is used to synchronize userâ€™s passwords across different devices.

Files stored on iCloud are broken into blocks and encrypted using AES-128 and a key that is derived from SHA-256 hash of the blockâ€™s contents. Note that the encrypted blocks of files, without userâ€™s identity information, are not stored on Appleâ€™s data center, instead they are stored on 3rd-party storages such as Amazon S3 and Windows Azure.

### 5.7 Continuity and Handoff

Starting from OS X Yosemite, Apple introduces Continuity and Handoff such that iPhone can be more synchronized with other iOS and OS X devices. With Continuity, a userâ€™s Mac, iPad that share the same Wi-Fi network with his/her iPhone can also make and receive phone calls as userâ€™s iPhone. The audio received from iPhone is transmitted to associated Mac/iPad through an encrypted end-to-end connection. The connection is established through APNs. Handoff is similar to Continuity, a user can conveniently continue working on the same task when he/she switches to another device. Similar to Continuity, two devices establish a Bluetooth Low Energy 4.0 connection through APNs. Then each generates a 256-bit AES key. Key exchange happens at the beginning of the communication. Exchanged keys will be used to encrypt and authenticate the messages sent through Bluetooth in GCM.

## 6. Apple Pay Security

Apple Pay to makes payments so easy that users only need to touch the Home Button to authorize and complete the transaction. Besides its easiness, Apple Pay also ensures that the transaction is performed in a secure, private way. One thing worth noticing is that Apple doesnâ€™t collect any transaction data during this process, and Apple doesnâ€™t know where you bought, what you bought and how much you paid.

### 6.1 Apple Pay Component

**Secure Element**: A certified chip hosts the applets that are used to manage Apple Pay. Encrypted credit/debit card information is sent from these applets.

**Near Field Communication (NFC) controller**: As a gateway to the Secure Element, NFC controller handles the communications between the application processor and Secure Element, and between the Secure Element and the Point-of-sale terminal.

**Passbook**: Users add and manage credit/debit card using Passbook. Users can also view the card information (Last 4 digits of the card number, the Device Account Number) and the privacy policy of the bank.

**Secure Enclave and Touch ID**: Users authorize each payment using Touch ID or his/her passcode. The Secure Enclave verifies if the fingerprints or the password is valid, if so the payment can proceed.

**Apple Pay Servers**: The Apple Pay Servers communicate with the devices and the payment network servers to manage the state of the credit/debit cards. [Apple 01]

### 6.2 Credit and debit card provisioning

When a new debit/credit card is added to Apple Pay, Apple securely sends the card information to the cardâ€™s bank, the bank will then decide whether this card can be used in Apple Pay. All the communication sessions with the bank are encrypted using SSL.

A unique Device Account Number is created for each newly added card. This Device Account Number is encrypted and stored in the Secure Element, which is isolated from iOS and Apple Pay server. The cardâ€™s bank can deactivate this Device Account Number if the user no longer wants to use it in Apple Pay. One thing worth noting is that the cardâ€™s information (card number, expiration date, etc) is not stored on the device.

### 6.3 How the payment is done by Apple Pay

When making a payment, the user can use Touch ID to authenticate his/her fingerprints. Once it is validated, the Secure Enclave will notify the Secure Element that the authorization is valid. Then the Secure Element will go ahead and allow the payment. The secure communication between the Secure Element and Secure Enclave is performed using a shared pair key which is created during the fabrication process. The encryption and authentication of this communication is based on AES, with cryptographic nonces on both sides to prevent replay attacks.

During the transaction, a one-time dynamic security code is generated using a key thatâ€™s provisioned in the Security Element applet, along with a counter that increments for each new transaction, a random number generated by the applet and another random number generated by the server or the terminal. The payment network and the bank can verify this transaction using the dynamic security code and the Device Account Number, hence the full credit/debit card number is never sent to the merchant

## 7. Issues with iOS security

Because iOS is a very advanced and secure mobile operation system, and Apple has strict app review procedure, it is almost impossible for malicious-looking apps to be published on the App Store and perform malicious tasks on iOS devices. However, there still exists some threats and loopholes in iOS security.

### 7.1 Benign apps could become evil

A research team claims that they successfully published their malicious app (called Jekyll app) on the App Store by hiding the malicious behaviors of the app during the app review process.

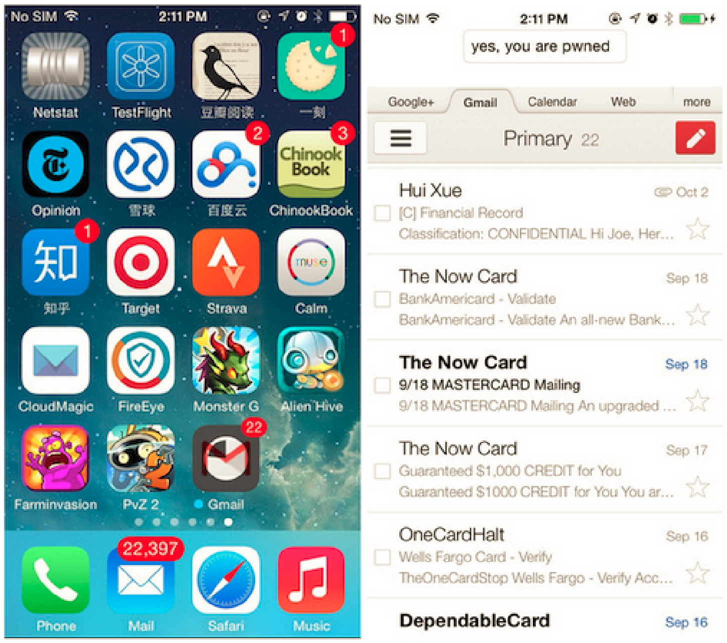
As they described in their paper, to walk around Appleâ€™s strict app review and code signing procedure, malicious attackers hide malicious behaviours in the app such that it looks legitimate and benign during the review process. After the user downloads and runs the app, the attacker can remotely activate the malicious functionalities (i.e., backdoor) in the app. Specifically, as presented in the paper, Jekyll app can post tweets, send emails and SMS texts without the userâ€™s knowledge. [Wang 03]

### 7.2 Unauthorized origin-crossing threats

In computing, the same-origin policy is an important concept in the web application security model. The policy permits scripts running on pages originating from the same site â€“ a combination of scheme, hostname, and port number â€“ to access each other's DOM with no specific restrictions, but prevents access to DOM on different sites. In contrast, the cross-origin is a mechanism that allows many resources (e.g., fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated. [Wiki 06][Wiki 07]

Unlike web browsers that have the same-origin security policy, mobile operating systems donâ€™t have such policy, as a result, usersâ€™ sensitive data could be exploited by malicious web origin. Researches found cross-origin issues in popular SDKs and high-profile apps such as Facebook and Dropbox, which can be exploited to access their usersâ€™ authentication credentials and other confidential information. [Wang 04]

### 7.3 Masque Attack



yes, you are pwned

Hui Xue — Oct 2
[C] Financial Record
Classification: CONFIDENTIAL Hi Joe, Her...

The Now Card — Sep 18
BankAmericard - Validate
BankAmericard - Validate An all-new Bank...

**The Now Card** — Sep 18
**9/18 MASTERCARD Mailing**
9/18 MASTERCARD Mailing An upgraded ...

The Now Card — Sep 17
Guaranteed $1,000 CREDIT for You
Guaranteed $1000 CREDIT for You You ar...

OneCardHalt — Sep 16
Wells Fargo Card - Verify
TheOneCardStop Wells Fargo - Verify Acc...

**DependableCard** — Sep 16

**Fig.4 Screenshot after Masque Attack[FireEye 08]**

A mobile security research team called FireEye recently claim that attackers can fool users to install malicious iOS apps which disguises itself as other commonly used apps, such as Gmail app or even your bank app. As long as the two apps use the same bundle identifier, the genuine app can be replaced by the malicious app in a way that looks exactly the same as the process of updating an existing app, as shown in Fig.4. This kind of masquerade increases the chance of tricking careless users. [FireEye 08]

This Masque attack could result in severe consequences. If the malicious app replaces your Gmail or bank app, after you run the app, malicious code will be executed and send your login credentials to the attacker. Besides, the malicious code can access the local files of the replaced app, resulting in more private data being stolen.

## 7.4 iOS jailbreaking

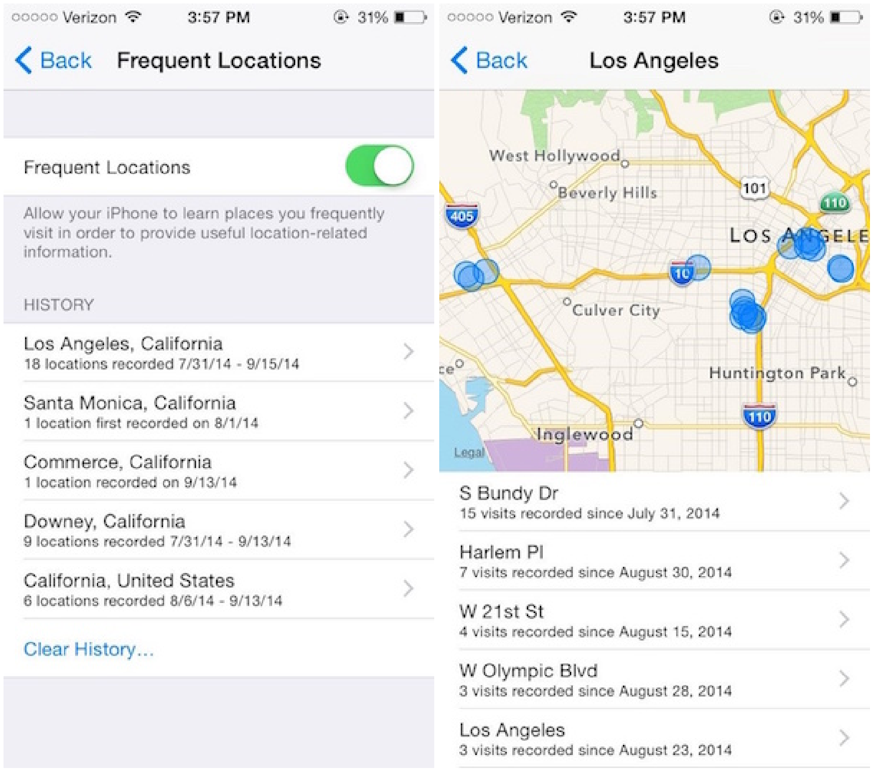
iOS jailbreaking is the process of removing limitations on iOS, Apple's operating system on devices running it through the use of software and hardware exploits. The reason that many people want to jailbreak their iOS devices is because it gives users root access to iOS, making users have more privileges on customizing their iOS devices. Such privileges include installing applications or extensions that are not provided by Appleâ€™s App Store, and personalizing the user interface of iOS. [Wiki 11]

The biggest problem of jailbreaking is that it puts users into huge security risks since it disables the Sandbox feature of iOS. Sandbox is an important iOS security feature during runtime process, it separates the applications installed on the device such that apps are restricted from accessing files associated with other files during runtime. If a user installs a malicious app on the jailbroken iOS device, the malicious code will have access to address book, photos, location data and other private data without telling the user. Besides the security risks brought by jailbreaking, once a user jailbreaks his/her iOS devices, the warranty (AppleCare) provided by Apple will be void immediately. Meanwhile, as Apple has constantly been incorporating new features into the latest version of iOS, the benefits brought by jailbreaking are becoming less and less. [O'Donnell 12] ][Emspak 13]

## 7.5 Privacy issues

Nowadays, companies are desperately trying to collect their usersâ€™ data in order for them to put appropriate advertisements in their app. However, users are not aware of their personal information or even sensitive data being secretly transmitted. In the recent release of iOS 8 by Apple, iOS enables some private settings by default while users might not know them at all. Specifically, here are some notable privacy concerns. [Aguilar 09][Whittaker 10]

**iOS keeps track your moves all the time**
Since iOS 7, iOS starts tracking your frequent locations by default, such as your home and workplace, as shown in Fig.5. While it might be somewhat useful for traffic alert, most people donâ€™t want Apple to know their frequent moves.



ooooo Verizon 3:57 PM 31%

Back Frequent Locations

Frequent Locations

Allow your iPhone to learn places you frequently visit in order to provide useful location-related information.

HISTORY

Los Angeles, California
18 locations recorded 7/31/14 - 9/15/14

Santa Monica, California
1 location first recorded on 8/1/14

Commerce, California
1 location recorded on 9/13/14

Downey, California
9 locations recorded 7/31/14 - 9/13/14

California, United States
6 locations recorded 8/6/14 - 9/13/14

Clear History...

ooooo Verizon 3:57 PM 31%

Back Los Angeles

S Bundy Dr
15 visits recorded since July 31, 2014

Harlem Pl
7 visits recorded since August 30, 2014

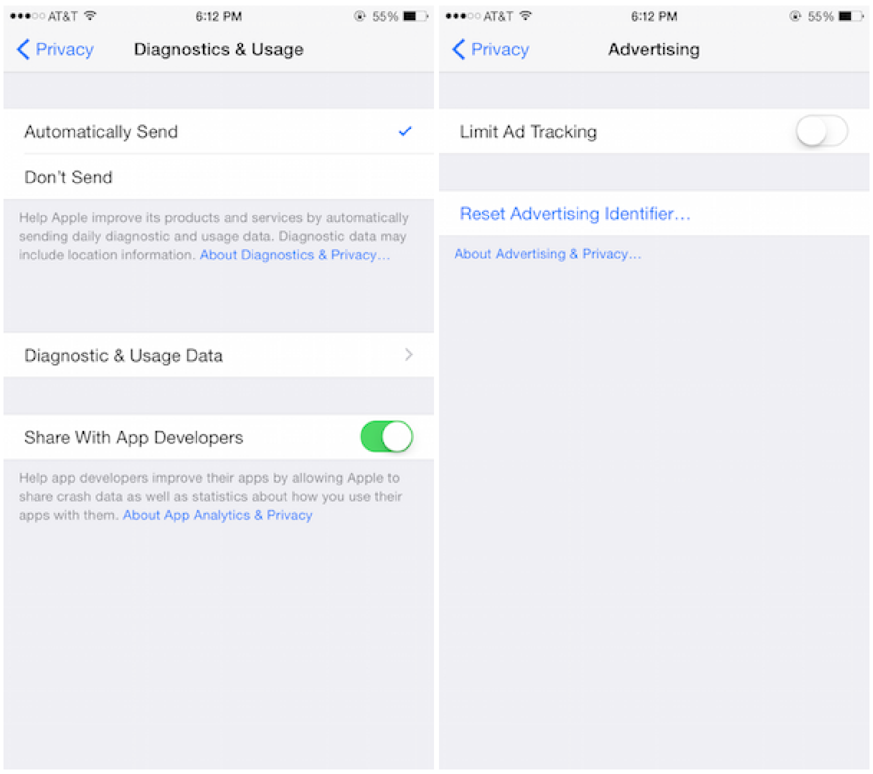
W 21st St
4 visits recorded since August 15, 2014

W Olympic Blvd
3 visits recorded since August 28, 2014

Los Angeles
3 visits recorded since August 23, 2014

**Fig.5 Screenshots of system location privacy settings[Aguilar 09]**

**Information sent to advertisers and developers**
iOS automatically sends your diagnostics and usage data to advertisers and developers by default, as shown in Fig.6.



●●●○○ AT&T 6:12 PM 55%

Privacy Diagnostics & Usage

Automatically Send ✓

Don't Send

Help Apple improve its products and services by automatically sending daily diagnostic and usage data. Diagnostic data may include location information. About Diagnostics & Privacy...

Diagnostic & Usage Data

Share With App Developers

Help app developers improve their apps by allowing Apple to share crash data as well as statistics about how you use their apps with them. About App Analytics & Privacy

●●●○○ AT&T 6:12 PM 55%

Privacy Advertising

Limit Ad Tracking

Reset Advertising Identifier...

About Advertising & Privacy...

**Fig.6 Screenshots of diagnostics and usage data settings**

## 8. Summary

By introducing various aspects of iOS security, we can conclude that iOS is one of the safest mobile operating systems in the world. For operating system security, iOS has a secure booting process to prevent low-level software from being modified. The incorporation of TouchID and passcodes also enhances the access security of iOS devices. The sophistication of operating system security ensures that the hardware and software can work together securely. For data security, iOS devices have a built-in AES crypto engine, which provides advanced encryption technique and accelerates the encryption process. iOS also encrypts all the keychains and files that are stored on the device. For network security, iOS incorporates the latest technologies and implements a set of industry-standard protocols such as SSL, TLS, WPA2 and so on, which enhances the security of the communications between iOS devices and the internet. For application security, Apple enforces that all the apps running on the iOS devices must be reviewed and approved by App Store, which successfully prevents malicious apps from being released to public. During the runtime process, iOS enforces that apps are sandboxed from each other such that each app is strictly restricted from accessing the data and resources of other apps. This mechanism effectively ensures the runtime security of iOS. Last but not least, compared with other mobile platforms, iOS has much fewer reported major security threats and attacks due to its sophistication, which makes it one of the most popular mobile operating systems in the world.

## 9. References

[Apple 01] Apple, Inc. â€œiOS securityâ€, Oct.2014
https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf

[Al-Qershi 02] Al-Qershi, Fattoh, Muhammad Al-Qurishi, Sk Md Mizanur Rahman, and Atif Al-Amri. "Android vs. iOS: The security battle." In Computer Applications and Information Systems (WCCAIS), 2014 World Congress on, pp. 1-8. IEEE, 2014.
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6916629&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6916629

[Wang 03] Wang, Tielei, Kangjie Lu, Long Lu, Simon Chung, and Wenke Lee. "Jekyll on iOS: When Benign Apps Become Evil." In Usenix Security, Aug.2013, pp.559-572
https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/wang_tielei

[Wang 04] Wang, Rui, Luyi Xing, XiaoFeng Wang, and Shuo Chen. "Unauthorized origin crossing on mobile platforms: Threats and mitigation." In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 635-646. ACM, 2013.
http://dl.acm.org/citation.cfm?id=2516727

[Wiki 05] "iOS", 2014
http://en.wikipedia.org/wiki/IOS

[Wiki 06] "Cross-origin resource sharing", 2014
http://en.wikipedia.org/wiki/Cross-origin_resource_sharing

[Wiki 07] "Same-origin policy", 2014
http://en.wikipedia.org/wiki/Same-origin_policy

[FireEye 08] "Masque Attack: All Your iOS Apps Belong to Us", 2014
http://www.fireeye.com/blog/technical/cyber-exploits/2014/11/masque-attack-all-your-ios-apps-belong-to-us.html

[Aguilar 09] Nelson Aguilar, "14 iOS 8 privacy settings everyone needs understand and probably change right now", 2014
http://ios.wonderhowto.com/how-to/14-ios-8-privacy-settings-everyone-needs-understand-and-probably-change-right-now-0157424/

[Whittaker 10] Zack Whittaker, "Seven privacy settings you should change immediately in iOS 8", 2014
http://www.zdnet.com/seven-privacy-settings-you-should-change-immediately-in-ios-8-7000033598/

[Wiki 11] "iOS jailbreaking", 2014
http://en.wikipedia.org/wiki/IOS_jailbreaking

[O'Donnell 12] Andy O'Donnell, "Is Jailbreaking Your iPhone Safe", 2014
http://netsecurity.about.com/od/iphoneipodtouchapps/a/Is-Jailbreaking-Your-iPhone-Safe.htm

[Emspak 13] Jesse Emspak, "How Jailbreaking Puts Your iPhone at Risk", 2014
http://www.tomsguide.com/us/iphone-jailbreak-risks,news-18850.html

[Apple 14] "Local and Remote Notification Programming Guide", 2014
https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html

[Apple 15] "Cryptographic Services Guide", 2014
https://developer.apple.com/library/mac/documentation/security/Conceptual/cryptoservices/SecureNetworkCommunicationAPIs/SecureNetworkCommunicationAPIs.html

## 10. List of Acronyms

| | |
|---|---|
| LLB | Low-Level Bootloader |
| UID | Unique ID |
| GID | Group ID |
| NIST | National Institute of Standards and Technology |
| DOM | Software Development Kit |
| APNs | Apple Push Notification service |
| RADIUS | Remote Authentication Dial In User Service |
| ROM | Read-Only-Memory |
| SSL | Secure Socket Layer |
| TLS | Transport Layer Security |

| | |
|---|---|
| AES | Advanced Encryption Standard |
| RNG | Random Number Generator |
| GCM | Galois/Counter Mode |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| NFC | Near Field Communication |
| SDK | Software Development Kit |
| DMA | Directed Memory Access |
| CTR-DRBG | Counter mode Deterministic Random Byte Generator |
| MAC | Media Access Control |
| PNO | Preferred Network Offload |

---

Last Modified: December 1, 2014
This and other papers on current issues in network security are available online at http://www.cse.wustl.edu/~jain/cse571-14/index.html
Back to Raj Jain's Home Page