UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Find My iPhone

Seminar Essay
by Nihal Yadalam Murali Kumar

Arbeitsgruppe
Codes und Kryptographie

Contents

1 Abstract

Apple is an organization that has developed an *Offline Finding* technology that ensures finding the missing devices of the owner in a private and secured manner. The authors of the paper "Who Can Find My Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System" [HSKH21] try to explore the cryptography and the OF components involved in the OF communication. Few security and privacy issues such as Vulnerable key storage, Denial of Service by relaying false reports and, Location correlation attacks were found in OF. Solutions were proposed to these issues found in Apple's OF.

2 Introduction

On the verge of digitization, a large amount of information is transmitted wirelessly through smartphones, laptops, tablets, etc. Security and privacy of mobile devices are essential to protect personal and business data stored in these devices. This scientific report mainly deals with the *Offline finding* capability found in many wireless devices. Authors of the paper [HSKH21] examine the Apple's Offline Finding (OF) feature and it's specifics. Apple's OF is a crowdsourced location tracking system known as "Offline Finding (OF)" developed in 2019, which is a closed-source software exclusive to Apple users. The main idea is that an apple device such as an iPhone use the application "Find My" to detect the missing apple devices or accessories such as airtags, iPad, and iPhone, etc and reports back to the owner using the internet. The crowd-sourced network consists of lots of closely connected devices worldwide and makes location tracking possible. This accounts for the largest network of devices in this era [HSKH21]. Apple Ecosystem plays a vital role in offline finding which consists of several types of devices namely iPhones, iWatches, iPods and MacBooks, etc which can be connected through the apple cloud known as iCloud.

2.1 Goals of Offline Finding Systems

- Ensures location tracking of missing devices even without internet.
- Ensures protection of user identity.
- Ensures confidentiality of location reports that contains confidential data.

The authors of [HSKH21] examine the security and privacy goals of Apple's OF network. They discuss the components involved in Apple's OF and perform security and privacy analysis of OF network. Later, they try to find certain security and privacy-specific vulnerabilities found on Apple devices. Finally, they propose to offer solutions to two of the found vulnerabilities on Apple's OF.

2.2 Simplified mechanism of OF

OF aims at tailing the missing devices in a privacy conserved way. There are 4 steps in tailing the missing devices as shown in fig 1 [HSKH21] :

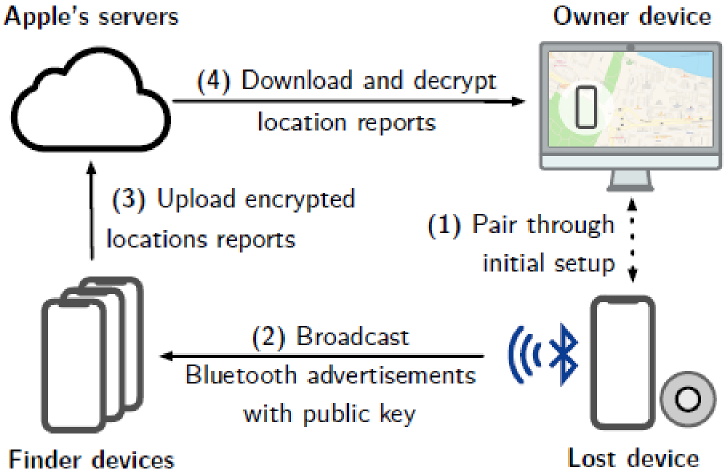
Apple's servers
Owner device
(4) Download and decrypt location reports
(3) Upload encrypted locations reports
(1) Pair through initial setup
(2) Broadcast Bluetooth advertisements with public key
Finder devices
Lost device

Figure 1: Mechanism of OF
Source: [HSKH21]

1. Initially, the owner device needs to pair with the other device that is assumed to be missing. This must be done with the same Apple id i.e., devices associated with the same owner.

2. Missing device emits Bluetooth Low Energy (BLE) advertisements once it loses its internet connection. This signal is caught by one of the finder devices in its environment.

3. The finder device will encrypt its location report and uploads it to iCloud.

4. Owner device can download the location report from iCloud by decryption.

3 Background Knowledge

In this section, authors of [HSKH21] briefly introduce key concepts involved in Apple's OF that need to be interpreted namely BLE (Bluetooth Low Energy) and the cryptography used in OF namely ECDH (Elliptic Curve Cryptography using Diffie-Helman Key Exchange) along with apple-specific components used in OF.

3.1 Bluetooth Low Energy (BLE)

It is the wireless personal area network technology used in devices such as smart bands, smartwatches, and iPhones that have a small battery and has less computational power. These devices emit signals known as advertisements once they lose internet connection. These signals are caught by other similar devices that have finder capabilities and inform their location to the owner [HSKH21]. Most of the mobile operating systems such as ios, Android, Windows and, Linux support BLE.

3.2 Public-key cryptography

In a public-key cryptography scheme, there exist two keys namely public key and private key. The private key is only known to the recipient and public key can be used by any party that is interested in encrypting the plain text. The public key is used by the sender to encrypt the plain text into cipher text whereas, the private key is used by the recipient to decrypt the ciphertext into the plain text as shown in the fig 2.

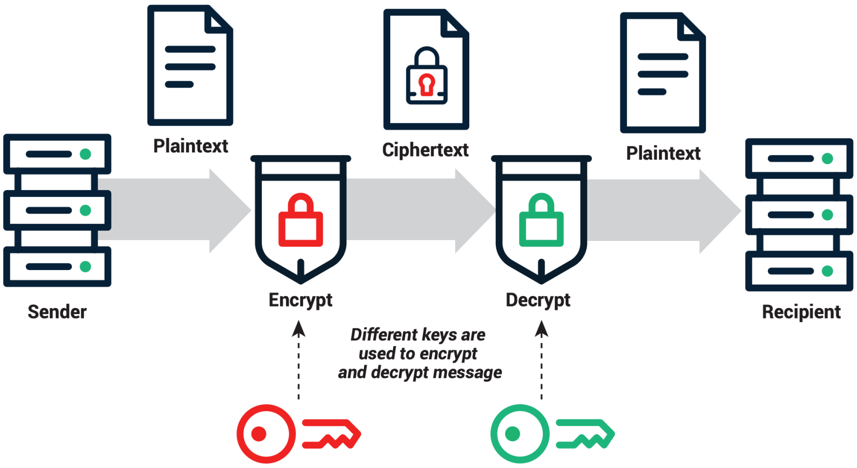

Figure 2: Public-key cryptography
Source: https://sectigo.com/resource-library/public-key-vs-private-key

3.3 Symmetric-key cryptography

In a Symmetric-key cryptography scheme, both the sender and receiver make use of the same secret key for both encryption and decryption. At the sender side, encryption is performed on Plain text to form a Cipher text using a secret key. Cipher text contains random letters, words, or strings. Using the same secret key, the receiver decrypts the Cipher text to Plain text.

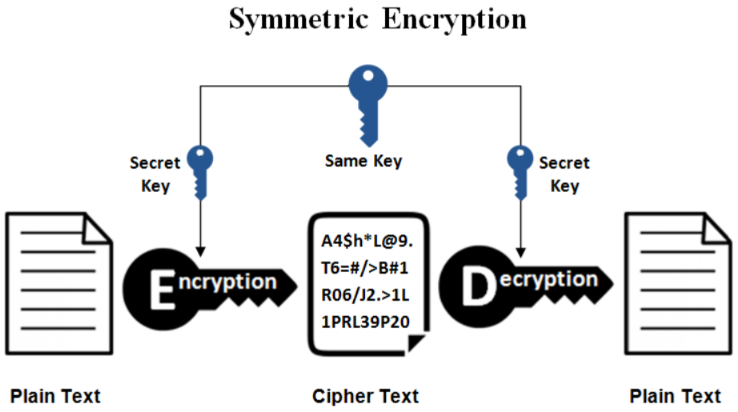

Figure 3: Symmetric-key cryptography
Source: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

3.4 Elliptical Curve Cryptography

Elliptical Curve Cryptography is based on a public-key encryption scheme that makes use of elliptical curves in developing efficient cryptography keys. ECC defines a one-way function meaning, a function can be computed easily in one direction and quite difficult to compute in opposite direction. In the fig 4, author of [Lan20] defines a random point on an elliptical curve as the start point i.e, A. The next point C is computed using a dot function i.e., A dot B to get C on an elliptical curve. Dot function is computed several times until an endpoint E is reached.

The algorithm is as follows :

- A dot $B = C$, extend a line drawn from A to B and reflect the intersecting point on x axis to get C. Reflected line is denoted by a dashed line as shown in first graph of fig 4.
- Now, A dot $C = D$, line drawn from A to C and reflect the intersecting point on x axis to get D. Reflected line is denoted by a dashed line as shown in second graph of fig 4.
- Lastly, A dot $D = E$, extend a line drawn from A to D and reflect the intersecting point on x axis to get E as the end point. Reflected line is denoted by a dashed line as shown in third graph of fig 4.

Here, public key is defined as a pair of x and y coordinates on an elliptical curve and private key is defined as the number of hops required from point A to E.

1. Public key : $(A : startpoint, E : endpoint)$.
2. Private key : 3, number of hops or dot functions required from point A to E.

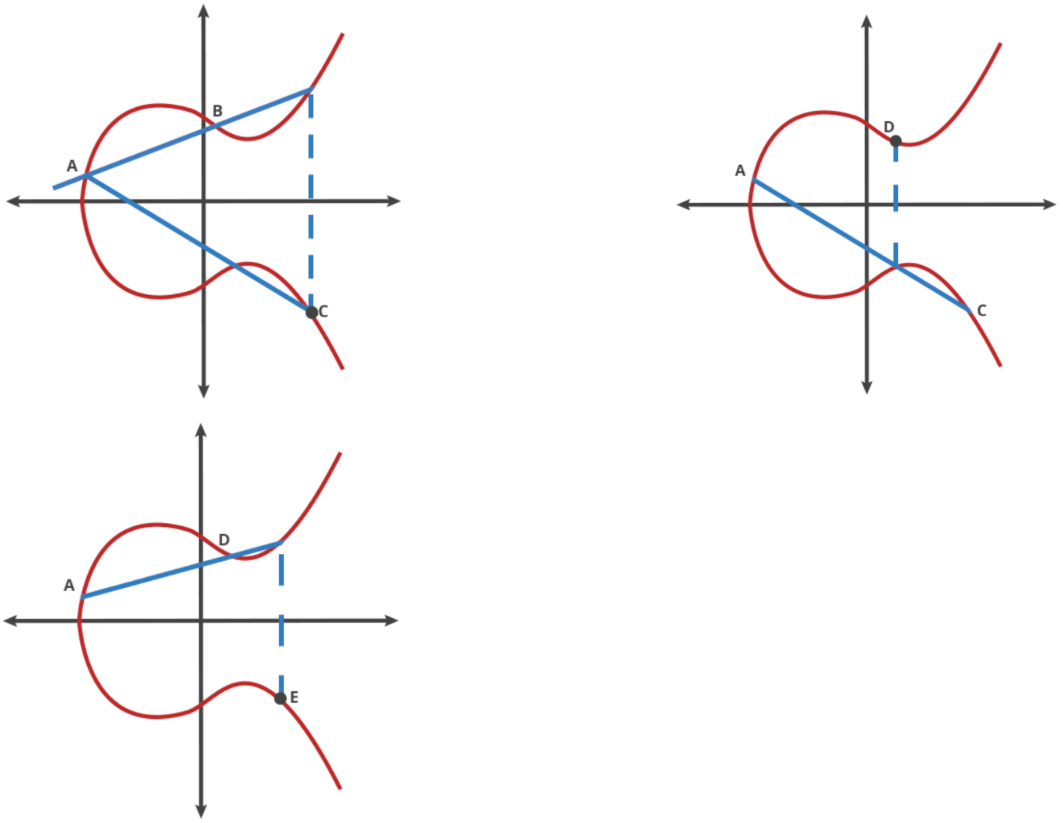

Figure 4: Elliptical Curve Cryptography
Source: https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/2/

3.5 Elliptic Curve Cryptography using Diffie-Helman Key Exchange (ECDH)

This is a type of symmetric cryptography scheme that allows two parties, each with elliptic curve public-private key pair to share secret information over an insecure channel. This works using a basic principle i.e, Elliptical Curve Cryptography.
Author of [Wil21] takes the following example involving two parties namely Alice and Bob. Each create their pairs of private and public keys respectively. Public keys are random coordinate points on an elliptical curve and private keys are integers. Public key is a combination of private key and a random generator G.

The working mechanism of ECDH in fig 5 can be as follows:

1. Alice generates a private key d_A and public key Q_A where $Q_A = d_A * G$.
2. Bob generates a private key d_B and public key Q_B where $Q_B = d_B * G$.
3. Shared secret is derived by exchanging the public keys of Alice and Bob respectively.

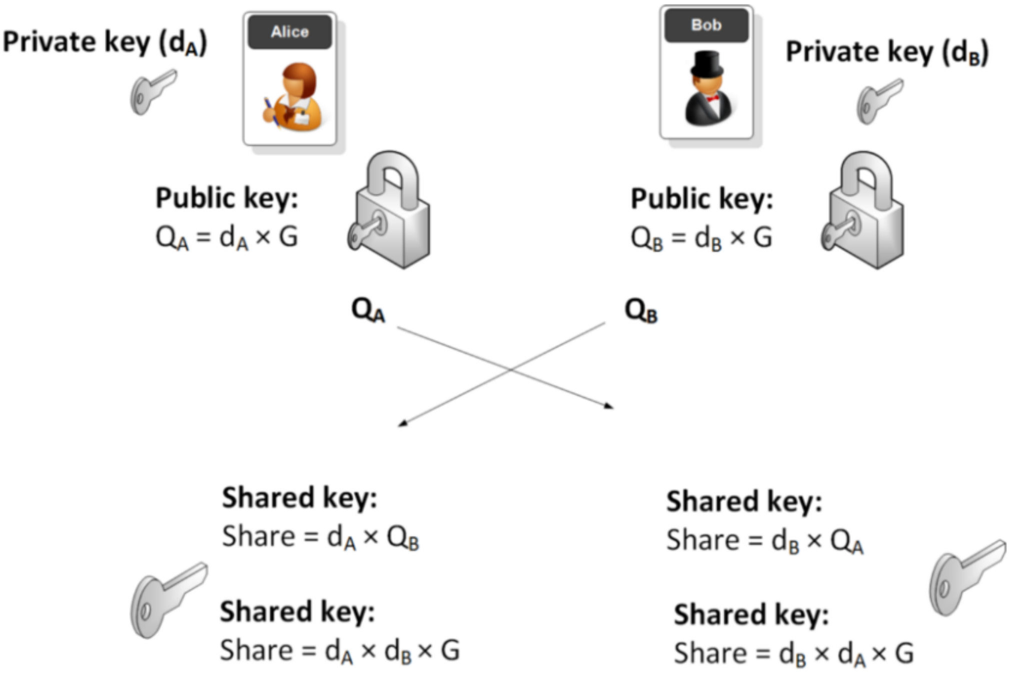

Figure 5: Elliptic Curve Cryptography using Diffie-Helman Key Exchange
Source: https://asecuritysite.com/encryption/ecdh3

4. After key exchange, shared keys of both Alice and Bob are same i.e, $Sharedkey = d_A * d_B * G$. Same shared key is used for encryption and decryption mechanisms.

3.6 OF components in Apple Ecosystem

There are two components that forms an Apple's OF namely iCloud and Apple keychain [HSKH21].

3.6.1 iCloud

It is the centralized cloud service used for storage and synchronization of data and keys to all its users provided by Apple. Downloading and uploading of location reports are enabled for the owner devices. Three types of devices exists in the Apple Ecosystem such as

- **Owner devices:** These are the devices that have a unique Apple id associated and have the capability to download location reports from iCloud. To download location reports, an owner device needs to be paired with other apple devices. Most often owner devices are Macbooks and iMacs.
- **Finder devices:** These are the devices that have the tracking capability of lost

devices and this forms the building block of OF network. Only iPhones and iPads have tracking abilities.

- **Missing devices:** These are the devices that have lost internet connectivity or in flight mode. They emit Bluetooth Low Energy (BLE) signals once they lose internet connection and later can be found using apple finder devices.

3.6.2 Keychain

It is the password management system developed by Apple in 1999 which acts as a system utility. It contains user details such as passwords, private keys and, certificates.

4 Cryptography

The core concept in this paper [HSKH21] lies in the cryptography algorithms used in Apple's OF. There are two types of keys used namely Master Beacon and Advertisement Keys :

4.1 Master Beacon

This is a key associated with every owner device. It contains a private-public key pair (x_0, y_0) (coordinate points on an elliptical curve) plus a 32-bit symmetric key SK_0 forms a Master Beacon. Only with the help of Master Beacon, derivation of advertisement keys (x_i, y_i) (coordinate points on an elliptical curve) is possible. These advertisement keys are emitted by the missing devices in the form of Bluetooth Low Energy (BLE) signals. [HSKH21]

4.2 Advertisement Keys

These are the keys denoted by a private-public key pair (x_i, y_i) (coordinate points on an elliptical curve) which are generated by missing devices and these keys change in every 15 minutes [HSKH21]. These keys can be derived by the Master Beacon (x_0, y_0) to access the location report of the finder devices. Apple's OF calculates advertisement keys with the combination of KDF and SHA-256 (hashing algorithm). KDF is denoted as *key derivation function.*

4.2.1 SHA-256

The hashing algorithm is developed as a one-way function meaning, it is difficult to get the original input data from output data. Secure Hash Algorithms (SHA) are designed to convert any length of Plain text to a fixed length of Cipher text and provide data integrity. In the SHA-256 algorithm, any length of Plain text is converted to a 256-bit Cipher text.

4.2.2 Key Derivation Function (KDF)

A key derivation function (KDF) is a key component of cryptographic systems and its objective is to take an origin of initial keying data, and derive from it to obtain one or more strong secret keys as described by the author of [Kra10]. The keying material here must be arbitrary and should not contain predictable data. It chooses some hashing algorithm such as SHA-256 to generate a key of fixed size. KDF function takes a tuple of three arguments and is denoted by
$DerivedKey = \mathrm{KDF}(OriginalKey, message, outputsize)$ where

- $DerivedKey$ is the derived key from $OriginalKey$ of size $outputsize$ bytes.
- $OriginalKey$ is the secret key that needs to generate one or more $DerivedKey$.
- $message$ is optional argument where it defines the operation to be performed on the $OriginalKey$.
- $outputsize$ defines the output size of $DerivedKey$ in bytes.

4.2.3 Derivation of advertisement keys using Master Beacon with SHA-256

Author of [HSKH21] defines 4 steps namely:

- $SK_n = KDF(SK_{n-1}, "update", 32)$
  The new symmetric key or the derived key SK_n of size 32 bytes is generated from the previous symmetric key or the original secret key SK_{n-1} as defined by the Key Derivation Function with SHA-256 as hashing algorithm. The symmetric key is used for both encryption and decryption mechanisms. Operation performed on SK_{n-1} is updation of keys.
- $(u_n, v_n) = KDF(SK_n, "diversify", 72)$
  The symmetric key SK_n obtained from the last step derives two more keys u_n, v_n of length 36 bytes each (total: 72 bytes) as defined by the Key Derivation Function with SHA-256 as hashing algorithm. Operation performed on SK_n is diversification of a key to give two more unique keys u_n, v_n. Author of [HSKH21] defines these keys as "anti-tracking keys" because, it is difficult for an intruder to predict the keys.
- $x_i = (x_0 * u_n) + v_n$
  The anti-tracking keys derived from the previous step u_n, v_n along with the master beacon private key x_0 generates a advertisement private key x_i. EC point operation is performed by master beacon private key x_0 and an anti-tracking key u_n on an elliptical curve.
- $y_i = x_i * G$
  The advertisement public key y_i is obtained from advertisement private key x_i and the generator G by an EC point operation on an elliptical curve.

Master Beacon key-pair (x_0, y_0) is synchronized in iCloud and can be used in decryption of the location report sent by a finder device. [HSKH21] The working mechanism of encryption and decryption of location reports can be found below :

4.3 Encryption

Finder plays a key role in encrypting it's location once it receives a BLE signal from the missing device. The advertisement emitted by the missing device contains a public key namely y_i and finder make use of this key for encryption. Finder uses OF technology which does ECDH (Elliptical Curve Diffie Hellmann) key exchange to derive the shared confidential data. Authors of [HSKH21] explain the algorithm in the following way:

- Once finder receives a signal from the missing device, it generates a new private-public key pair $(x^{'}, y^{'})$ (coordinates on an elliptic curve).
- Finder performs ECDH key exchange on an elliptical curve and derives a shared secret by exchanging finder private key $x^{'}$ and missing device public key y_i.
$SharedSecret = x^{'} * y_i$ (point operation on an elliptical curve)
- Finder derives a new symmetric key by supplying missing device's public key y_i and SHA-256 as hash function to Key Derivation Function (KDF). Finder uses the beginning 16 bytes as an encryption key $a^{'}$ and ending 16 bytes as the Init vector in the newly generated symmetric key. Init vector is also known as the initialization vector of a specific length used for some iteration purpose. Here, it is an initial SHA-256 hash value. It encrypts the location report using both $a^{'}$ and Init vector.

4.4 Decryption

This is performed by the owner device by sending an HTTP request to the iCloud. Decryption of the location report is exactly opposite to encryption of the location report. Authors of [HSKH21] explain the decryption mechanism in the following way:

- Owner device receives the correct advertised private-public key pairs (x_i, y_i) (coordinates in an elliptical curve) by finding the hashed SHA-256 value on missing device public key y_i.
- Owner device performs ECDH key exchange on an elliptical curve between the missing device private key x_i and finder device public key $y^{'}$ to derive the shared secret.
$SharedSecret = MissingDevicePrivateKey * FinderPublicKey$ (point operation on an elliptical curve)
- Owner decrypts the location report by deriving the symmetric key $a^{'}$ and init vector from the shared secret.

5 Security and Privacy analysis of Apple's OF

5.1 Security and Privacy vulnerabilities found on Apple's OF

The authors of [HSKH21] identified privacy and security vulnerabilities of Apple's OF. They are as follows

5.1.1 Vulnerable key storage

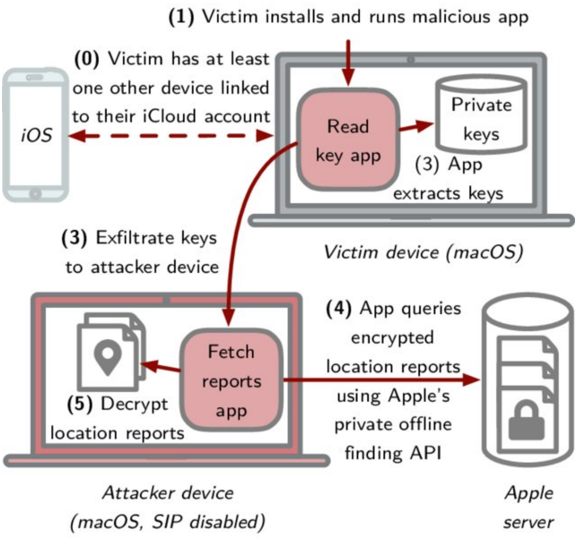
(1) Victim installs and runs malicious app
(0) Victim has at least one other device linked to their iCloud account
iOS
Read key app
Private keys
(3) App extracts keys
(3) Exfiltrate keys to attacker device
Victim device (macOS)
(4) App queries encrypted location reports using Apple's private offline finding API
Fetch reports app
(5) Decrypt location reports
Attacker device (macOS, SIP disabled)
Apple server

Figure 6: Control flow of accessing the victim's location history
Source: [HSKH21]

This vulnerability was found when analyzing the cryptography component of Apple's OF [HSKH21].

Reason: The advertised keys emitted by the missing devices change in a time frame of 15 minutes once they are lost. Hence, numerous amount of keys are generated. Owner devices contain a Master Beacon that is responsible for deriving the advertisement keys. Apple tried to cache advertisement keys in the owner's device to avoid expensive cryptographic operations [HSKH21]. However, a local application on macOS can access these keys because they are cached in the owner's device. This becomes a threat to the owner's device as the attacker can easily access the master beacon.

Flow: Attacker access the victim's location history by the following steps as shown in fig 6 as follows [HSKH21]:

1. Initially, two of the victim's devices such as iPhone and macbook are connected to an iCloud account.

2. Victim installs and runs the app sent by an attacker that is capable of reading the device specific keys.

3. This malicious app gets hold of private keys present in the victim's macbook and exfiltrate these keys to attacker's macbook.

4. SIP (System Integrity Protection) of attacker's device is disabled as he/she can run unauthorised code on it. Now, attacker queries the Apple server for location reports through Apple's private offline finding API using the victim's private keys.

5. Attcker can decrypt the location reports of the victim's device using his keys.

5.1.2 Denial of Service by relaying false reports

DoS is a computer network attack that causes interruption of the services provided by the host or an information system in a network making the resources unavailable to its users. This vulnerability was found in the Bluetooth component of Apple's OF [HSKH21].

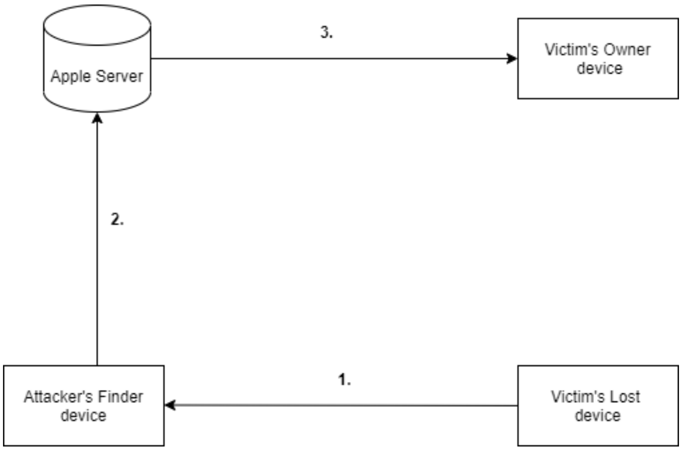

Figure 7: Control flow of DoS attack
Source: flow explained in [HSKH21]

Reason: A missing device emits BLE advertisements which contains it's public key. Attacker's device hearing this advertisement can easily forge the false location report and upload it to iCloud as this public key is not authenticated. This makes way for Denial-of-Service attacks. False location report of the missing device will be downloaded by the owner. [HSKH21]

Flow: Attacker performs DoS attack as shown in fig 7. It follows [HSKH21]

1. Attacker's finder device hears the BLE advertisement emitted by victim's lost device.
2. Attacker's finder generates the error prone location report using the victim's public key and uploads to the Apple server.
3. Victim's Owner device downloads incorrect location report of his lost device from the apple server.

5.1.3 Location correlation attack

This vulnerability was found in Apple's server communication component of Apple's OF. This attack aimed at de-anonymizing various users in a network and revealing their identities [HSKH21].

Reason: Apple as an organization can correlate the locations of its users through their Apple ids when they access the "Find My" application. This occurs when the users try to upload or download location reports of their devices from the apple server. [HSKH21]

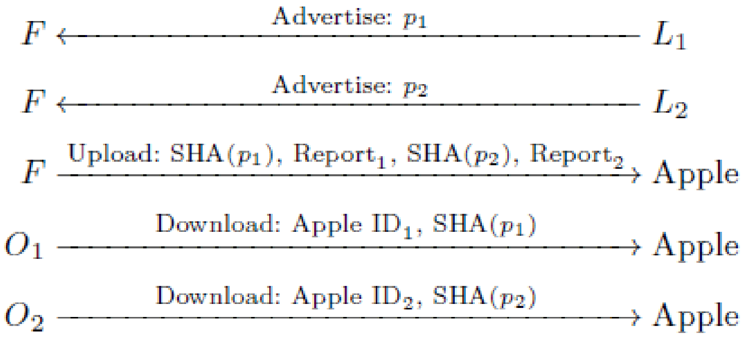
F
Advertise: p_1
L_1
F
Advertise: p_2
L_2
F
Upload: SHA(p_1), Report$_1$, SHA(p_2), Report$_2$
Apple
O_1
Download: Apple ID$_1$, SHA(p_1)
Apple
O_2
Download: Apple ID$_2$, SHA(p_2)
Apple

Figure 8: Control flow of Location correlation attack
Source: [HSKH21]

Flow: Location correlation attack can be performed as shown in fig 8. There are six entities involved in control flow namely Finder F, missing devices (L_1 and L_2), Owner devices (O_1 and O_2) and Apple (organization). It follows [HSKH21]

1. Finder device hears the BLE signals emitted by the missing devices L_1 and L_2 that are close to each other and are identified by their public keys p_1 and p_2. These missing devices belongs to the owners O_1 and O_2.

2. Finder device uploads the encrypted location reports of L_1 and L_2 to Apple sever. Encryption is performed on the public keys p_1 and p_2 belonging to L_1 and L_2 using Secure Hash algorithm (SHA).

3. owner devices O_1 and O_2 downloads the location reports of L_1 and L_2 from the Apple server identified by their $AppleID_1$ and $AppleID_2$. Apple can correlate locations of two owners O_1 and O_2 identified by their AppleIDs.

5.2 Solutions proposed to counter few vulnerabilities of OF

Authors of [HSKH21] proposed to give solutions to two vulnerabilities namely vulnerable key storage and location correlation attack found on Apple's OF.

1. **Vulnerable key storage :** Authors of [HSKH21] propose both temporary and permanent solutions to this vulnerability. A temporary solution is disabling the *Offline Finding* feature on the "Find My" application of Apple devices. Disable option can be found in the 'settings' menu option on an iPhone [App21b]. This prevents an attacker to access the device's internal keys. However, due to this the location of a lost device cannot be tracked as the *Offline Finding* feature is disabled. Three Permanent solutions were proposed namely :

   a) **Caching and encryption of file data:** Caching is a process of storing important information internally in the device that may later require for calculation purpose. Apple's system files containing keys and location data must be encrypted and cached internally. Decrytion key must be stored in Apple keychain. This prevents an attacker to access the keys as files are encrypted safely.

   b) **Limiting the permissions to access device's cached files:** Permissions must be set to the cached files and this combats the attacker in gaining access. These permissions can be set by choosing a cached folder and navigating to "Get Info" property [App21a] on a macbook. Setting privilege to "No Access" can prevent an attacker to access this cached folder.

   c) **macOS sandboxing:** Sandboxing is a mechanism of running a task in the virtual environment to safeguard from the external entities. However, this is resource intensive task and should be done whenever necessary. macOS 10.15.7 implements sandboxing technique to store all the cryptography keys in separate directory [HSKH21].

2. **Location correlation attack:** Authors of [HSKH21] think about hiding the user's identity information in owner and finder device to mitigate location correlation attack. However, finder's information needs to be validated by the apple server when uploading the location report. So, the authentication of owner device should be disabled when downloading of location report from the apple server. This decision must be enforced by Apple to solve privacy issue of its users.

6 Conclusion

Apple has introduced *Offline finding* technology that enables tracking the missing devices of the owner through the internet. This scientific report deals with the cryptography involved in Apple's OF i.e., Elliptic Curve Cryptography using Diffie-Helman Key Exchange (ECDH) and deriving the shared secret. ECDH is performed by both finder and owner device during encryption and decryption of location reports. Authors of [HSKH21] carried out the security and privacy analysis of Apple's OF and found few vulnerabilities such as vulnerable key storage, location correlation, and DoS. Solutions to two vulnerabilities namely vulnerable key storage and location correlation were given.

References

[App21a] Apple. Change permissions for files, folders, or disks on Mac. `https://support.apple.com/guide/mac-help/change-permissions-for-files-folders-or-disks-mchlp1203/mac`, 2021. Change permissions for files, folders, or disks on Mac.

[App21b] Apple. Turn off Find My on your iPhone, iPad, or iPod touch. `https://support.apple.com/en-lamr/HT211149`, 2021. Turn off Find My on your iPhone, iPad, or iPod touch.

[HSKH21] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. Who can find my devices? security and privacy of apple's crowd-sourced bluetooth location tracking system. *arXiv preprint arXiv:2103.02282*, 2021.

[Kra10] Hugo Krawczyk. Cryptographic extraction and key derivation: The hkdf scheme. In *Annual Cryptology Conference*, pages 631–648. Springer, 2010.

[Lan20] Lane Wagner. Basic Intro to Elliptic Curve Cryptography. `https://qvault.io/cryptography/elliptic-curve-cryptography/`, 2020. Basic Intro to Elliptic Curve Cryptography.

[Wil21] William J Buchanan. Elliptic Curve Diffie Hellman (ECDH). `https://asecuritysite.com/encryption/ecdh3`, 2021. Elliptic Curve Cryptography using Diffie-Helman Key Exchange (ECDH).