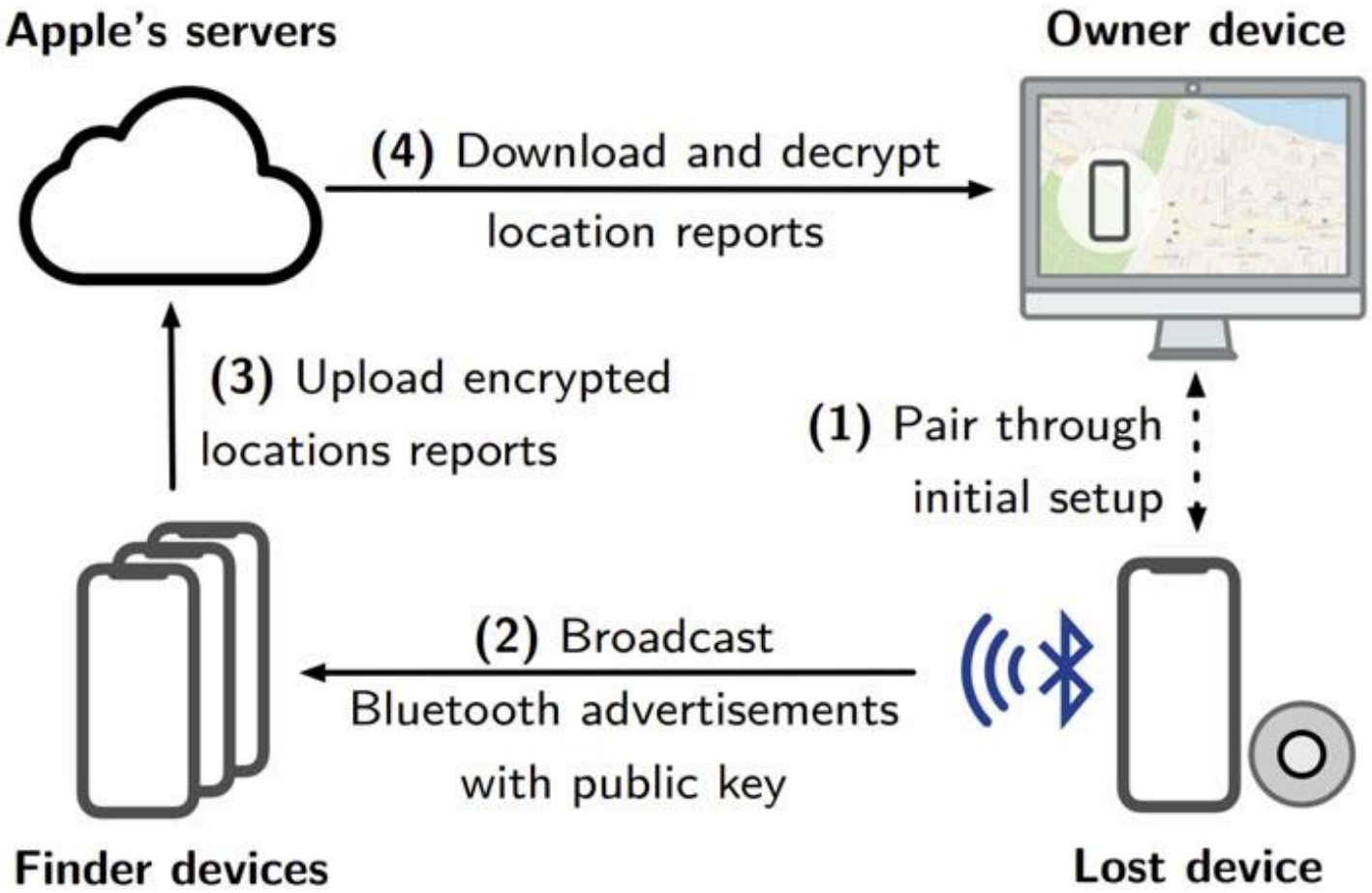
Who can Find My Devices?

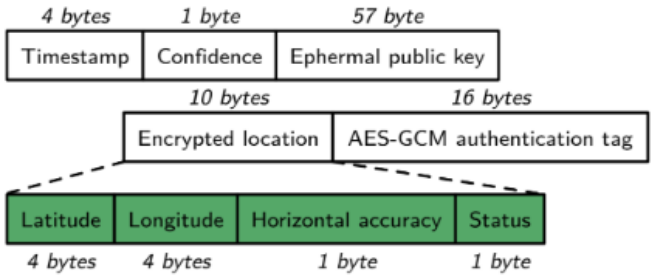
"I'm a software engineer, so I can confirm it works by magic."

One App to find it all.

How does it work?



How does it work? [Detailed]



Apple's servers

Owner device

6) Download and decrypt the report

- Find the right location report using an hash mechanism with p_i
- Computing the symmetric key that finder device used thanks to d_i and the public finder key p'
- Decrypt the finder's report

5) Finder devices upload informations

- Ephemeral Public Key
- Encrypted location report

1) ECDH

$master\ beacon\ key = (d_0, p_0) + SK_i$

Finder devices

Lost device

3) Lost device shares advertisement key -> p_i

2) Operations Done By the Lost Device

$SK_i = KDF(SK_{i-1}, update, 32)$

$(u_i, v_i) = KDF(SK_i, diversify, 72)$

$d_i = (d_0 * u_i) + v_i$

$p_i = d_i * G$

4) Operations Done By Finder Devices

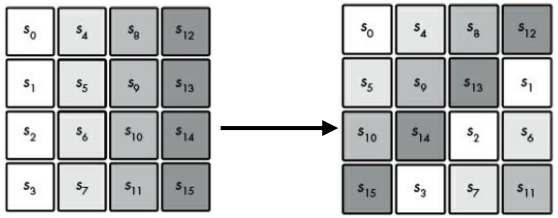
$ECDH\ based\ on\ p_i\ to\ obtain\ a\ shared\ secret$

$SharedSecret_i = KDF(SharedSecret_{i-1}, update, 32)\ with\ p_i\ as\ entropy$

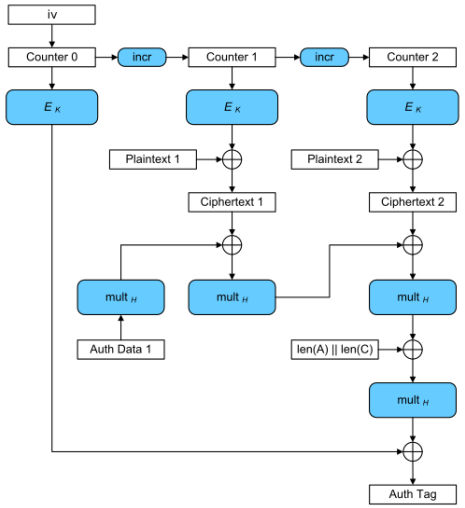
$Split\ SharedSecret_i\ in\ two\ pieces:\ first\ 16\ bytes\ as\ e'\ and\ last\ 16\ bytes\ as\ IV$

$Encrypt\ his\ data\ about\ location\ and\ time\ with\ AES-GCM$

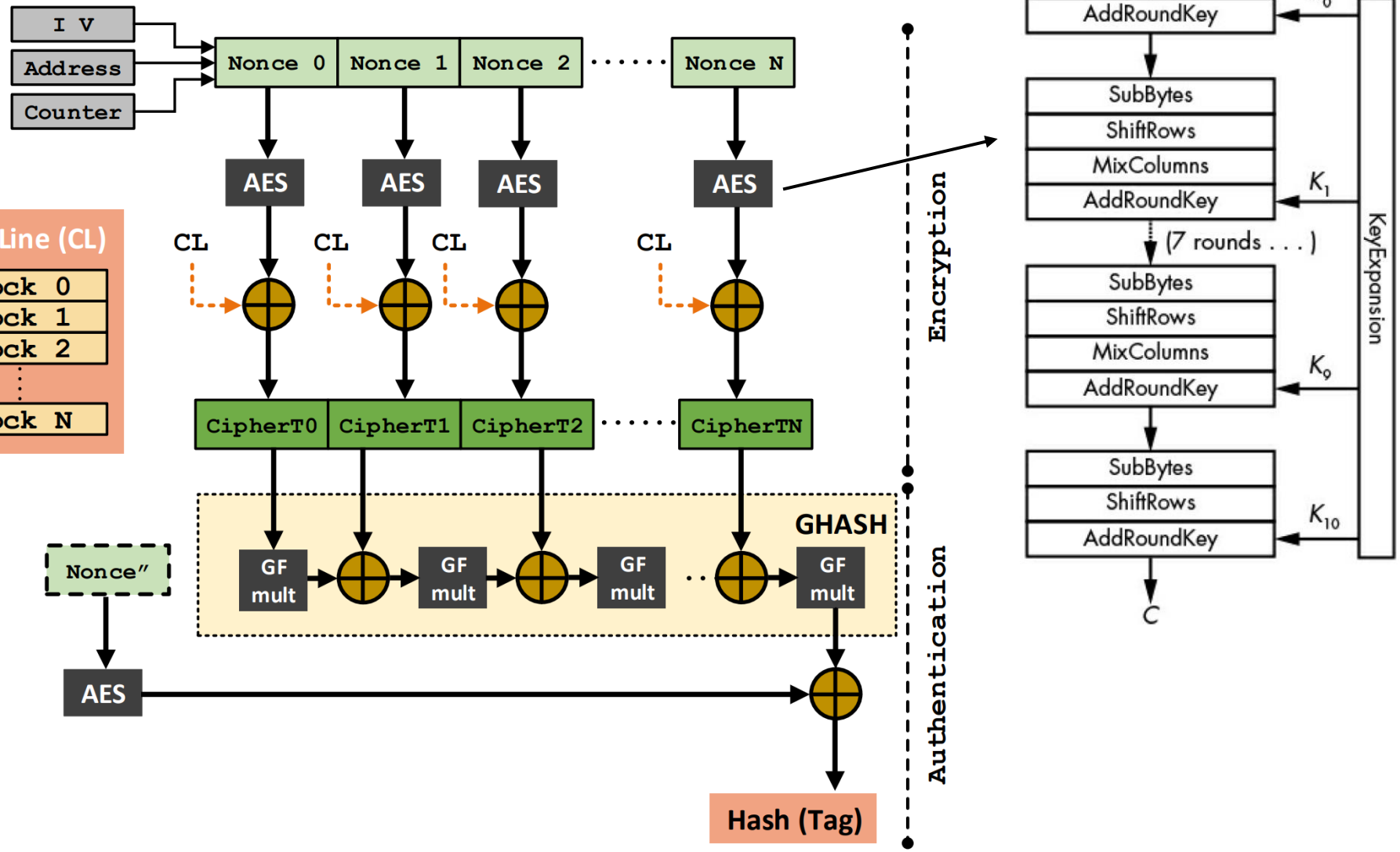
AES-GCM



Example of ShiftRows operation



GMC Internals Scheme



There is something missing…

- How do we know the **EXACT** advertisement key produced by a **missing device**???

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

Device Identification. While we did not recover the exact details of the authentication mechanism, we have observed that both finder and owner devices provide identifiable tokens to Apple's servers. In particular, owner devices provide their Apple ID to access location reports. In § 9, we show that by requesting IDs, Apple's servers are—in principle—able to correlate the locations of different owners.

Our analysis has shown that the advertisement keys are precomputed for up to *nine* weeks into the future, which allows an adversary to continue downloading new reports even after the victim has uninstalled the malicious application.

10.1 Vulnerability

§ 6 describes that the location privacy of lost devices is based on the assumption that the private part of the advertisement keys is only known to the owner devices. The advertisement keys change every 15 minutes and OF supports retrieving location reports from the last seven days, so there is a total of 672 advertisement keys per device, for which there exist potential location reports on Apple's servers. In principle, all of these keys could be generated from the master beacon key (cf. § 6.1) whenever needed. However, Apple decided to cache the advertisement keys, most likely for performance reasons. During our reverse engineering efforts, we found that macOS stores these cached keys on disk in the directory `/private/var/folders/<Random>/com.apple.icloud.searchpartyd/Keys/<DeviceId>/Primary/<IdRange>.keys`. The directory is readable by the local user and—in extension—by any application that runs with user privileges. On iOS, those cache files exist as well, but they are inaccessible for third-party applications due to iOS's sandboxing mechanism.

Alternatives to ECDH?

- Why do we not use Elgamal??? OVERHEAD

8.21 Note (*efficiency of ElGamal encryption*)

(i) The encryption process requires two modular exponentiations, namely $\alpha^k \bmod p$ and $(\alpha^a)^k \bmod p$. These exponentiations can be sped up by selecting random exponents k having some additional structure, for example, having low Hamming weights. Care must be taken that the possible number of exponents is large enough to preclude a search via a baby-step giant-step algorithm (cf. Note 3.59).

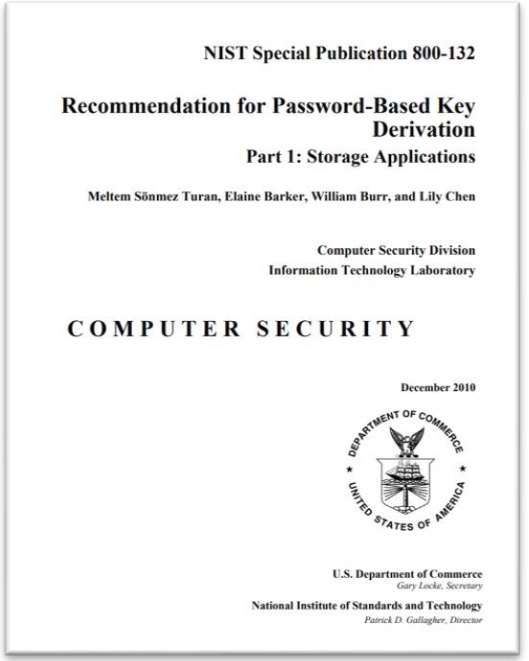
(ii) A disadvantage of ElGamal encryption is that there is *message expansion* by a factor of 2. That is, the ciphertext is twice as long as the corresponding plaintext.

Where Things Break

- **Nonce repetition in AES-GCM**:

$$\begin{aligned}&\mathbf{GHASH}(H, A_1, C_1) \oplus \mathbf{AES}(K, N \| 0) \oplus \mathbf{GHASH}(H, A_2, C_2) \oplus \mathbf{AES}(K, N \| 0)\\ =\ &\mathbf{GHASH}(H, A_1, C_1) \oplus \mathbf{GHASH}(H, A_2, C_2) \oplus (\mathbf{AES}(K, N \| 0) \oplus \mathbf{AES}(K, N \| 0))\\ =\ &\mathbf{GHASH}(H, A_1, C_1) \oplus \mathbf{GHASH}(H, A_2, C_2)\end{aligned}$$

If the same nonce is used twice an attacker can obtain the Hash Key!


NIST Special Publication 800-132

Recommendation for Password-Based Key Derivation

Part 1: Storage Applications

Meltem Sönmez Turan, Elaine Barker, William Burr, and Lily Chen

Computer Security Division
Information Technology Laboratory

COMPUTER SECURITY

December 2010

U.S. Department of Commerce
Gary Locke, Secretary

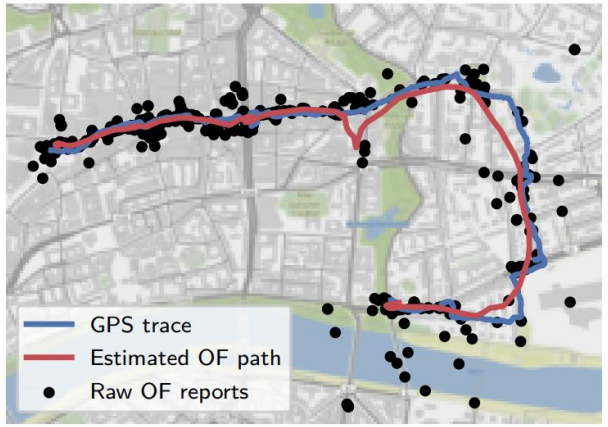
National Institute of Standards and Technology
Patrick D. Gallagher, Director

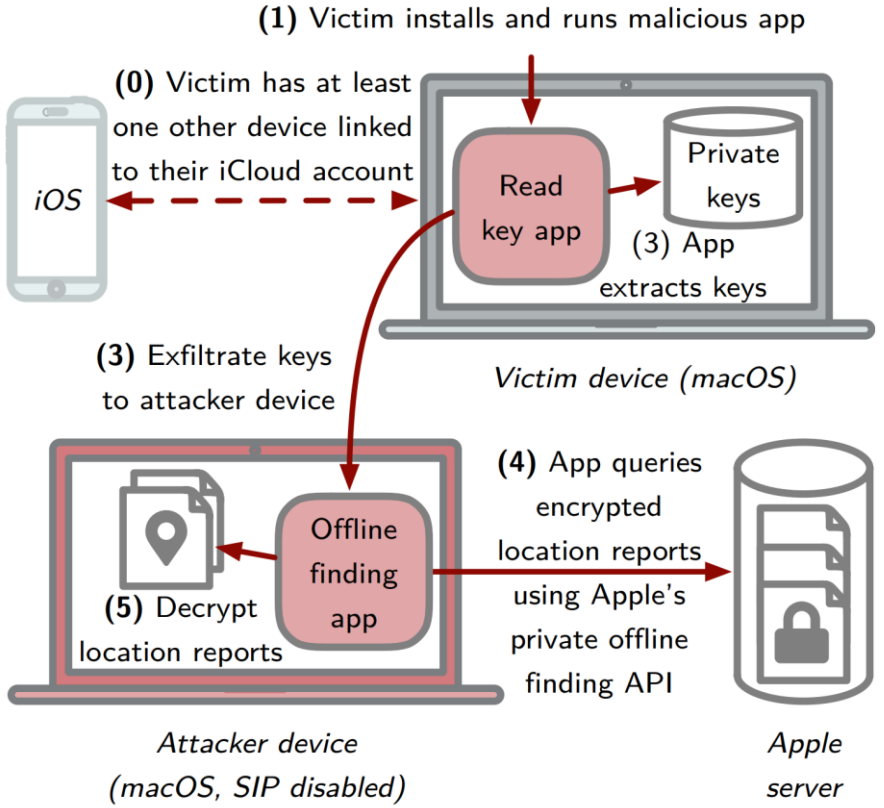
- **Breaking ECDH Using Another Curve:** *"Say that Alice and Bob are running ECDH and have agreed on a curve and a base point, G. Bob sends his public key dBG to Alice. Alice, instead of sending a public key dAG on the agreed upon curve, sends a point on a different curve, either intentionally or accidentally. Unfortunately, this new curve is weak and allows Alice to choose a point P for which solving ECDLP is easy. She chooses a point of low order, for which there is a relatively small k such that kP = O. Now Bob, believing that he has a legitimate public key, computes what he thinks is the shared secret dBP, hashes it, and uses the resulting key to encrypt data sent to Alice. The problem is that when Bob computes dBP, he is unknowingly computing on the weaker curve. As a result, because P was chosen to belong to a small subgroup within the larger group of points, the result dBP will also belong to that small subgroup, allowing an attacker to determine the shared secret dBP efficiently if they know the order of P." p.334,* SERIOUS CRYPTOGRAPHY A Practical Introduction to Modern Encryption by Jean-Philippe Aumasson
- **KDF: low entropy input, extremely short salt**
- **Using the shared secret as key session at the end of (EC)DH**

Security and Privacy Analysis

```
/private/var/folders/<Random>
/com.apple.icloud.searchpartyd/Keys/<DeviceId>
/Primary/<IdRange>.keys
```

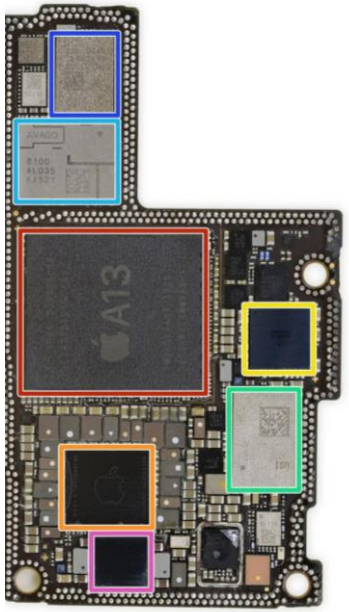
- **Insecure Key Storage**
- **Unauthorized Access of Location History**
- **Excessive advertisement keys caching:** up to 9 weeks into the future
- **Social Tracking**





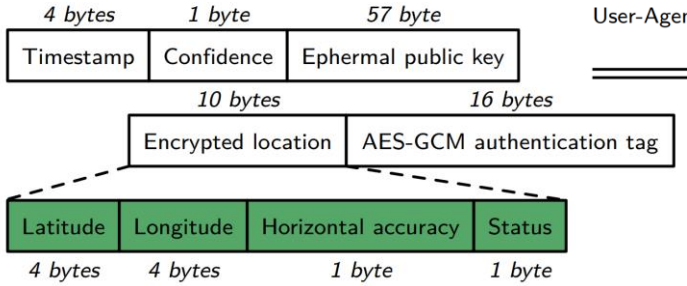
Technical Implementation

Lost iPhone



Bytes	Content (details cf. [6, § 5.1])
0–5	BLE address $((p_i[0] \mid (0b11 \ll 6)) \mid\mid p_i[1..5])$
6	Payload length in bytes (30)
7	Advertisement type (0xFF for manufacturer-specific data)
8–9	Company ID (0x004C)
10	OF type (0x12)
11	OF data length in bytes (25)
12	Status (e.g., battery level)
13–34	Public key bytes $p_i[6..27]$
35	Public key bits $p_i[0] \gg 6$
36	Hint (0x00 on iOS reports)

Finder's iPhone



HTTP request headers for uploading location reports

Request Header	Value
X-Apple-Sign1	Device identity certificate (base64)
X-Apple-Sign2	SHA-256 hash of the signing CA (base64)
X-Apple-Sign3	Device ECDSA signature (ASN.1)
X-Apple-I-TimeZone	Client's time zone (e.g., GMT+9)
X-Apple-I-ClientTime	Client's time (Unix)
User-Agent	"searchpartyd/1 <iPhoneModel>/<OSVersion>"

iCloud

Owner's Laptop

HTTP request headers for downloading location reports

Request Header	Value
Authorization	Base64 encoded basic authentication
X-Apple-I-MD	Anisette data
X-Apple-I-MD-RINFO	Anisette data
X-Apple-I-MD-M	Anisette data
X-Apple-I-TimeZone	Client's time zone
X-Apple-I-ClientTime	Client's time (ISO 8601)
X-BA-CLIENT-TIMESTAMP	Client's time (Unix)
User-Agent	"searchpartyd/1 <iPhoneModel>/<OSVersion>"

Demo Time!

- **Language**: Python
- **Libraries**: tinyec, cryptodome, cryptography, secrets

MaxBubblegum47/
PyFind

Documentation

- **Who Can Find My Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System** *(Alexander Heinrich, Milan Stute, Tim Kornhuber and Matthias Hollick)*
- **SERIOUS CRYPTOGRAPHY A Practical Introduction to Modern Encryption** *(Jean-Philippe Aumasson)*
- (Discrete Mathematics and Its Applications) *Alfred Menezes, Paul van Oorschot, Scott Vanstone* - **Handbook of applied cryptography**
- **Practical Cryptography for Developers** *(Svetlin Nakov)*
- **SMARTS: Secure Memory Assurance of RISC-V Trusted SoC** (Ming Ming Wong, Jawad Haj-Yahya, Anupam Chattopadhyay, Nanyang Technological University (NTU) Singapore anupam@ntu.edu.s)

Questions?