DEMO: OpenHaystack Mobile – Tracking custom Find My Accessories on Smartphones

Lukas Burg*
Max Granzow*
Department of Computer Science
TU Darmstadt, Germany
lukas.burg@hemalu.de
max.granzow@stud.tu-darmstadt.de

Alexander Heinrich
Secure Mobile Networking Lab
Department of Computer Science
TU Darmstadt, Germany
aheinrich@seemoo.de

Matthias Hollick
Secure Mobile Networking Lab
Department of Computer Science
TU Darmstadt, Germany
mhollick@seemoo.de

ABSTRACT

In 2021, OpenHaystack on macOS was the first step into liberating Apple's Find My technology to be integrated into any Bluetooth-capable device. By using custom firmware for microchips like the ESP32, it was possible to build custom trackable accessories similar to an Apple AirTag in size and functionality.

OpenHaystack Mobile is a cross-platform mobile application that runs on Android and iOS. It extends previous work, which was tied to macOS, to be available to more users. It provides BLE-based location tracking functionality based on Apple's Find My network. We built a cross-platform capable app by using the Flutter framework for the user interface and the cryptographic operations. The user experience is improved compared to a desktop-based app because it allows tracking lost valuables anywhere. Furthermore, the app allows directly route users to lost devices with smartphone navigation apps like Google Maps.

CCS CONCEPTS

• **Networks** → **Network protocols**; • **Security and privacy** → **Mobile and wireless security**; Public key encryption.

KEYWORDS

cross-platform, mobile development, location tracking, Bluetooth Low Energy, key finder

ACM Reference Format:
Lukas Burg, Max Granzow, Alexander Heinrich, and Matthias Hollick. 2022. DEMO: OpenHaystack Mobile – Tracking custom Find My Accessories on Smartphones. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3507657.3529655

1 INTRODUCTION

In 2019 Apple turned their device ecosystem with the Find My network into the most extensive crowd-based location tracking system with about one billion devices participating [2]. Though, it remains, as so often with Apple products, completely proprietary, and closed to non-Apple users. With last year's release of OpenHaystack for macOS, the first step to open up the Find My network was taken. OpenHaystack offers firmware for microchips, such as the *ESP32* or the *micro:bit*, turning them into so-called accessories, which the Find My network can track. Subsequently, those accessories can be created, deployed, and located with the OpenHaystack for macOS application.

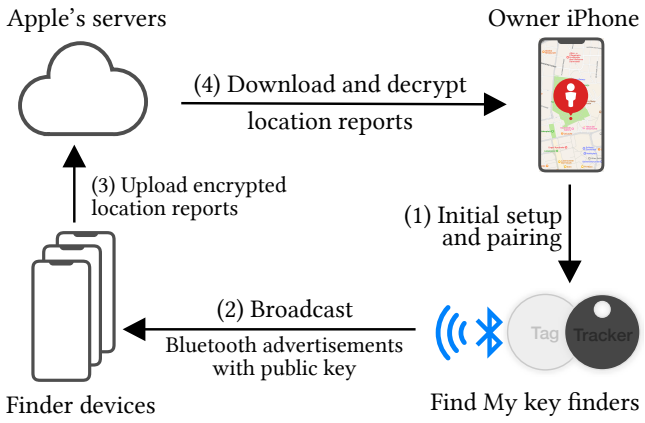


Figure 1: Simplified demonstration of the Find My network by [5]

Nevertheless, most people do not have access to macOS and cannot profit from it. Furthermore, a desktop-based solution can be impractical because the loss of valuables can happen on holidays or at locations where access to the home desktop is restricted.

OpenHaystack Mobile is the next step to liberate the Find My Network for all people. Utilizing the Flutter [14] framework, we implement OpenHaystack cross-platform. Users on different operating systems can now create and locate accessories via the Find My network. With this, we were able to further reduce the need to buy into Apple's ecosystem and enable even more people to use the Find My network.

2 BACKGROUND

This section introduces Apple's Find My network and its cryptography employed as first described by [8].

2.1 The Find My network

Figure 1 gives a brief overview of the Find My Network.

*Both authors contributed equally to this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WiSec '22, May 16–19, 2022, San Antonio, TX, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9216-7/22/05.
https://doi.org/10.1145/3507657.3529655

(1) OpenHaystack generates a new private-public key pair. The public key is transferred to the accessory during the setup. The related private key remains in the OpenHaystack app, which stores it in the secure key storage of the dedicated operating system [7].
(2) Devices publicly advertise an elliptic-curve public key using BLE [9].
(3) When surrounding Apple *finder* devices with a GPS module receive an advertisement, they determine their location, encrypt the location, and upload the encrypted location report to Apple's servers.
(4) OpenHaystack can then download, and decrypt the location reports.

Cryptography. When *finder* devices receive such an advertisement, they generate a new ephemeral key pair and perform an elliptic-curve Diffie-Hellman key agreement (ECDH) with the generated private key and the received public key to derive a shared secret. A symmetric encryption key is derived from this shared secret and then used to encrypt the location. At last, the encrypted location, the public key of the new generated key pair, and the hash of the received public key are uploaded to Apple's server. With this, the location report is end-to-end encrypted and the uploaded location reports are identifiable with a hash of the advertised public key.

Using the inverse of the procedure mentioned above, the users can decrypt the locations. Namely, they perform an ECDH with the private key and the downloaded public key of the *finder* device. Thus, derive the same shared secret as the *finder* device and can derive the same symmetric key to decrypt the location report. With this, the Find My network implements anonymity for the *finder* devices uploading the locations and ensures the confidentiality of the location data.

2.2 OpenHaystack for macOS

The OpenHaystack implementation for macOS (see on GitHub [6]) currently supports the creation of trackable non-Apple accessories. Additionally, it acts as an interface to the Find My network, allowing for tracking the accessories previously created with OpenHaystack. It employs a plugin for Apple Mail, which allows OpenHaystack to inherit Apple Mail's entitlements needed to authenticate against Apple's servers and download location reports [6, 7].

3 OPENHAYSTACK MOBILE

The main goal of our project was the portability of the created application. To achieve this, we used the cross-platform Flutter framework developed by Google [14]. Flutter apps use a single code base that can be compiled to a native executable for each target platform. Compared to other cross-platform frameworks based on a web view, this provides better support for native features and avoids performance and security problems with these solutions [3]. Creating a separate application for each target platform is not feasible because it is not economical and sustainable in development. The compiled application can be provisioned via an app store or sideloading.

Flutter projects can use a package repository with premade open-source packages [13]. Unfortunately, several packages do not offer support for desktop operating systems. If this support will be added in the future, *OpenHaystack Mobile* would also run on Linux, Windows and macOS without any further changes.

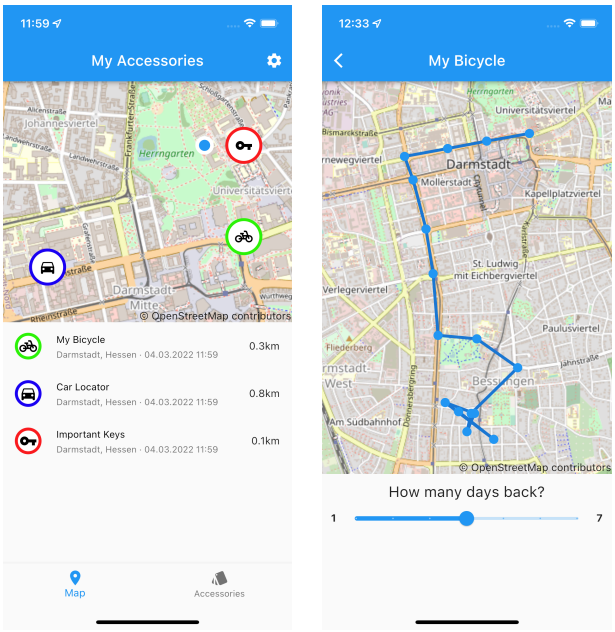


Figure 2: Screenshots of *OpenHaystack Mobile*. To the left the dashboard showing all saved accessories. To the right the history view showing locations of an accessory for the last four days.

3.1 Accessing Location reports

A custom authentication mechanism is required to access Apple's iCloud servers and to download location reports for an owned accessory. This mechanism is only available on Apple devices and in iCloud for Windows [7].

We retrieve location reports over a macOS-based backend to remove the need for an Apple ID. The backend server connected with an iCloud account, and we implemented a modified version of the OpenHaystack macOS app that runs an HTTPS server in the background. To future-proof this setup we recommend disabling system integrity protection, a mechanism that allows deep system access and therefore, allows access to necessary authentication tokens even if Apple would disable current methods used by OpenHaystack for macOS. Users who use our backend server API do not need to provide any authentication. The end-to-end encrypted location reports disallow accessing sensitive user data (e.g., location data) on the OpenHaystack server.

3.2 Cross-Platform Cryptography in Flutter

To implement the decryption we required the NIST P-224 Elliptic-Curve (EC), the AES block cipher in Galois/Counter Mode (GCM), the SHA256 digest for the key derivation function, and the elliptic-curve Diffie-Hellman (ECDH) operation. Those requirements are only met by the *pointycastle* [4] package maintained by Bouncy

Castle Inc. They also manage the identically named cryptography library from the Java and C# world. Although *pointycastle* lacks support for ECDH, it implements the multiplication operator for BigInt with a point on an EC. As the ECDH is a simple multiplication of the public key point on the EC with the private key – which is a simple number, thus a BigInt – we were able to implement the ECDH ourselves. With this, we implemented all necessary cryptographic operations for the decryption of the location reports from the Find My network, and all necessary functions to create new trackable accessories.

3.3 Evaluation

In our current implementation, the app fetches up to 2000 location reports from Apple's servers via our backend server. Each location report is individually end-to-end encrypted.

With our prototype, we managed to implement the decryption in a cross-platform capable way, leveraging the *pointycastle* package for cryptographic operations. However, the decryption of two thousand location reports takes about 30s on a Google Pixel 3a. To alleviate this, we used Darts concurrency model with Isolates [12] to prevent the user interface from freezing during the decryption process.

The accuracy of the location reports is the same as in the existing OpenHaystack application. However, it is easier to use on mobile devices and removes the need for Apple hardware on site.

4 DEMO

In the presented demo, the participants can see the complete functionality of *OpenHaystack Mobile*. More precisely, they will see how new accessories can be created and deployed. We present the import and export of OpenHaystack accessories using a JSON encoded file or a manual input of the necessary parameters. Most importantly, the demo will show the tracking of accessories and how they can be followed by using the history view. See figure 2 for two screenshots of *OpenHaystack Mobile*, the left one shows the accessory list and the right one the history view of one accessory.

5 WHAT COMES NEXT?

Based on the current state, additional features can be added to improve usability and functionality.

Desktop support. With the steadily growing user base of Flutter [10], the number of libraries, as well as the platform support by these packages, will likely increase heavily in the future. If the necessary support for packages will be added in the future, the application can be made available on desktop platforms like Linux and Windows.

Finding Android Smartphones. Another addition would be the ability to turn the device running *OpenHaystack Mobile* into a trackable device itself. The Android smartphone would have to send out the BLE advertisements described earlier for this functionality. For now, this implementation is only possible by using root permissions on Android.

Geofencing. With a new geofencing feature it would be possible to measure location changes of items when they leave a geofence. This could be used to keep track of valuables while on holiday. Furthermore, with geofences it would be possible to determine if the user has moved on while a device has been left at a previous location.

Locating displaced devices. Furthermore, added BLE scanning and distance estimates would allow you to find your own devices when they are misplaced in your apartment.

5.1 Stalking protection

Recent articles by mass media have shown that Find My devices can be misused to stalk people by tracking their location against their will [11]. Similar stalking attacks can be deployed by OpenHaystack accessories at the size of a 2 Euro coin [1]. The Android App AirGuard can find OpenHaystack devices and warn tracked users that they are followed [5]. To limit stalking abilities we propose the following changes to OpenHaystack in future work:

(1) Reduce the number of locations shown in the history view of the app.
(2) Hide the location of moving devices.
(3) Limit the number of location report fetches per day.

AVAILABILITY

We are going to integrate *OpenHaystack Mobile* into the official OpenHaystack GitHub repository [6]. The required macOS backend will also be added to the repository and allow third parties to host such a backend server. We will not provide a hosted backend server for all users due to legal reasons.

REFERENCES

[1] acalatrava. 2021. *OpenHaystack - Alternative Firmware #57.* Secure Mobile Networking Lab. https://github.com/seemoo-lab/openhaystack
[2] Apple Inc. [n. d.]. Apple introduces AirTag. https://www.apple.com/newsroom/2021/04/apple-introduces-airtag/
[3] Andreas Biørn-Hansen, Tor-Morten Grønli, and Gheorghita Ghinea. 2018. A Survey and Taxonomy of Core Concepts and Research Challenges in Cross-Platform Mobile Development. *ACM Comput. Surv.* 51, 5, Article 108 (nov 2018), 34 pages. https://doi.org/10.1145/3241739
[4] Bouncy Castle Inc. 2022. *pointycastle v3.5.1.* https://github.com/bcgit/pc-dart
[5] Alexander Heinrich, Niklar Bittner, and Matthias Hollick. 2022. AirGuard - Protecting Android users from Attacks by Apple Find My Devices *(WiSec '22).* Association for Computing Machinery, New York, NY, USA.
[6] Alexander Heinrich and Milan Stute. 2021. *OpenHaystack.* https://github.com/seemoo-lab/openhaystack
[7] Alexander Heinrich, Milan Stute, and Matthias Hollick. 2021. OpenHaystack: A Framework for Tracking Personal Bluetooth Devices via Apple's Massive Find My Network. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (Abu Dhabi, United Arab Emirates) *(WiSec '21).* Association for Computing Machinery, New York, NY, USA, 374–376. https://doi.org/10.1145/3448300.3468251
[8] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. 2021. Who Can Find My Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System. *Proceedings on Privacy Enhancing Technologies* 2021, 3 (July 2021), 227–245. https://doi.org/10.2478/popets-2021-0045
[9] Apple Inc. 2021. *Find My Network Accessory Specification.* https://developer.apple.com/find-my/
[10] JetBrains. 2021. *Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021.* Retrieved March 20, 2022 from https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/
[11] Ryan Mac and Kashmir Hill. 2021. Are Apple AirTags Being Used to Track People and Steal Cars? *The New York Times* (Dec. 2021). https://www.nytimes.com/2021/12/30/technology/apple-airtags-tracking-stalking.html
[12] Dart Team. 2022. *Dart Concurrency.* Retrieved March 20, 2022 from https://dart.dev/guides/language/concurrency
[13] Dart Team. 2022. *Flutter and Dart Package Manager.* Retrieved March 20, 2022 from https://pub.dev/
[14] Flutter Team. 2022. *Flutter Website.* Retrieved March 20, 2022 from https://flutter.dev