ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE
ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL
ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

USENIX Security ’25 Artifact Appendix: Tracking You from a Thousand Miles Away! Turning a Bluetooth Device into an Apple AirTag Without Root Privileges

Junming Chen
George Mason University

Xiaoyue Ma
George Mason University

Lannan Luo
George Mason University

Qiang Zeng*
George Mason University

A Artifact Appendix

The nRootTag artifact demonstrates a novel attack that transforms computers into trackable “AirTags” without requiring root privileges by exploiting Apple’s Find My network. The attack exploits a vulnerability in Apple’s Find My network, which Apple has acknowledged. We also assisted Apple to reproduce the attack. Apple recently released patches in iOS 18.2, visionOS 2.2, iPadOS 17.7.3, 18.2, watchOS 11.2, tvOS 18.2, macOS Ventura 13.7.2, Sonoma 14.7.2, Sequoia 15.2 to fix the vulnerability. However, the attack remains effective as long as unpatched iPhones or Apple Watches are in the proximity of the computer running our trojan.

A.1 Abstract

The artifact includes all components needed to reproduce the attack: a command & control server setup script for managing tracked devices, a custom database for efficient key storage, a GPU-accelerated key search program, and proof-of-concept trojans for Linux, Windows, and Android systems. The artifact provides comprehensive documentation, screen recordings, and three pre-computed rainbow tables to facilitate the reproduction of the key findings.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Our research involves experimentation with a live third-party system, i.e., Apple Find My network. The Trojan for Linux is designed to write a bytestream to */dev/rfkill* to enable Bluetooth functionality. It attempts to connect to a specified address, defaulting to localhost, to transmit the Bluetooth adapter address and request a public key. Subsequently, it advertises the public key, thereby facilitating the tracking of the test device. The Trojans for Android and Windows are configured to attempt Bluetooth activation with or without user consent and to send Bluetooth advertisements for the exchange of addresses.

Table 1: Hardware configurations used for each component

Component	Hardware Requirement
C&C Server	Alienware Aurora R5 Desktop
Key Seeker	Alienware Aurora R5 Desktop, NVIDIA RTX 4090
Trojan - Linux	Alienware Aurora R5 Desktop
Trojan - Android	Google Pixel 6, Pixel 4
Trojan - Windows	Gigabyte Z690, Bluetooth Adapter RTL8761B

Evaluators shall exercise extra caution when handling the location reports, as possession of the corresponding private key permits decryption of the report and revelation of the location. It is advised that evaluators refrain from extracting private keys from artifacts and from broadcasting lost messages at private locations (e.g., residential addresses); instead, a new key should be generated using Key Seeker. Evaluators should not share the private key or location report publicly.

A.2.2 How to access

We have released the artifact on Zenodo[1] to comply with the Open Science Policy. The artifact includes a README, instructional videos, Docker scripts for setup, reproduction codes for lost message search and advertisement, online key search guidelines, and three precomputed rainbow tables. We also host a corresponding GitHub page[2] for issuing potential bug fixes and accepting pull requests.

A.2.3 Hardware dependencies

Table 1 details the hardware configuration in our experiment.

*Corresponding author.

[1] https://doi.org/10.5281/zenodo.14728530

[2] https://nroottag.github.io/

Table 2: Software configurations used for each component

Component	Software Requirement
C&C Server	Ubuntu 22.04
Key Seeker	Ubuntu 22.04
Trojan - Linux	Ubuntu 22.04
Trojan - Android	Android 13, Android 11
Trojan - Windows	Windows 11

A.2.4 Software dependencies

Table 2 shows the software setup, which is the same as the software environment used in the demonstration videos.

If evaluators have further requirements for confirming the submission and decryption of location reports, evaluators may refer to online projects like *Chapoly1305/FindMy*.[3] We highlight that this process is beyond the scope of our paper and we do not vouch for any of those projects. To comply with the single-blind review policy, we may not be able to decrypt location reports for evaluators.

A.2.5 Benchmarks

None, our work does not compare with other datasets.

A.3 Set-up

In the README file, we provide detailed steps to replicate our work. In addition to the README file, the demonstration videos show the essential steps and results related to the key findings presented in the paper. We recommend that evaluators refer to these materials for a better replication experience.

A.3.1 Installation

The C&C Server is written in Python and uses several libraries. We provide two methods to set up the environment; either method should work.

```
# Assume current directory is nRootTag
# Method A, install from requirements
  python3 -m venv venv
  source venv/bin/activate
  pip install -r requirements.txt

# OR, Method B, use our packed venv
  tar xvf venv.tgz
  source venv/bin/activate

# Lastly, Start Server
  python3 cnc_server.py
```

A.3.2 Basic Test

After the server is launched, evaluators can access the webpage at http://localhost:7898/docs to review the available APIs and their usage. The webpage presents each API with a description and usage examples. Evaluators can also use the "Try it out" feature for each API and follow the inline descriptions to test its functionality.

A.4 Evaluation workflow

A.4.1 Major Claims

We made the following claims in our main paper.

(C1): Key Seeker achieves a key generation and searching speed approximate of 8.23 billion KPS using a single RTX 4090. It is proven by experiment **E1**, as described in Section 6.4 of our paper.

(C2): The finder devices in Apple's Find My network do not distinguish between different types of Bluetooth addresses while processing lost messages. It is proven by experiment **E2**, described in Section 6.4 of our paper.

(C3): As described in Section 6.4 of our paper, the Trojan is capable of turning on Bluetooth on devices running Linux, Windows 11, and Android versions prior to 13, as proven by experiment **E3**. The detailed information of devices is listed in Table 5 and Table 6 of the main paper.

A.4.2 Experiments

(E1): [Key Seeker Benchmark] [0.5 human-hour]

Preparation: In this experiment, the Key Seeker is evaluated for correspondence to **C1**.

Execution: Navigate to nRootTag/executables and run the following commands. In the given example code, **-t 0** disables searching with the CPU, **-gpu** enables searching with the GPU, **-o /dev/null** discards all outputs.

```
chmod +x ./Seeker_CUDA_12
./Seeker_CUDA_12 -t 0 -gpu -o /dev/null -p
```

Results: Employing an RTX 4090, the anticipated rate is approximately 8.23 Gkey/s. Variability in the outcomes may occur based on the GPU's state and potential concurrent usage by other applications. Similar experiments can be conducted with alternative GPU models. A line commencing as follows denotes the average speed of GPU processing.

```
[Time 896.3s][7.51 Gkey/s][GPU 8.23 Gkey/s]
```

(E2): [Lost message advertisement] [1 human-hour + 22 compute-hours]

Preparation: To validate claim **C2**, the experiment entails retrieving the Bluetooth adapter's address from a Linux-based system. Following this, the procedure for

[3] https://github.com/Chapoly1305/FindMy

initiating the private key search is demonstrated. For this purpose, one might employ an RTX 4090 to search for a key pair associated with the Bluetooth address. Achieving a precise match, where all 48 bits of the address correspond to the 48 most significant bits (MSBs), the probability of identifying a matching key is estimated to be 90% within 22 hours, as delineated by Equation 1 in our study. Evaluators might opt to utilize cloud-based GPU platforms and leverage multiple GPU units to decrease the computational time required, as discussed in Section 6.1 of the paper.

Execution: Evaluators shall take note of the device Bluetooth address for later comparison.

```
  hcitool dev
```

Evaluators shall start the server by using the given script and keep the script running. The script will open a port and listen on port 7898.

```
  python3 cnc_server.py
```

The evaluators should deploy the Trojan onto the testing apparatus. Upon the Trojan's execution, a search task is initiated on the server. The extracted Bluetooth address is then output by the Trojan. The console will exhibit the phrase "Advertisement Registered" once a matching key is identified. Following the retrieval of a public key from the C&C server, the Trojan will commence advertising activities.

```
# Launch Trojan
  chmod +x ./Trojans/trojan_linux.py
  bash -c "Trojans/trojan_linux.py"
# Launch Key Seeker
  python3 seeker_standalone.py
```

Results: Evaluators should verify the consistency of the given address across the specified address, the Web API for /search-task, and the Trojan interface. Evaluators should check the public and private keys utilizing the API /review-key. Trojan advertisements can be detected using Bluetooth scanning tools such as *nRF Connect*.[4] Evaluators should look for advertisements with a company ID of 0x004C from the specified public address. To confirm advertisements handled by the locator, evaluators can query Apple through external software like *Chapoly1305/FindMy*. When Apple devices, such as iPhones with the Find My feature activated, are present, evaluators should be able to retrieve location reports promptly.

(E3): [Bluetooth Stealth Enable] [2 human-hours]

Preparation: Evaluators shall install with the operating system as described in **C3**, in accordance with Table 5 and Table 6 of our main paper.

Execution: Evaluators are required to disable the Bluetooth function within the system configurations for each operating system. On Linux, the script *trojan_linux.py* shall be executed. For Android systems, evaluators need to install the application *nRootTag-Android.apk* and initiate the nRootTag Trojan program. Meanwhile, on Windows 11, the file *nRootTagWinRT.exe* should be launched by the evaluators.

Results: On all three platforms (i.e., Linux, Android, Windows), evaluators should observe the Bluetooth becoming active after launching Trojan, irrespective of the status of the address for advertising purposes.

A.5 Notes on Reusability

The artifact can be employed in future research related to the Find My network. Extensive in-line comments are included to facilitate the reuse of the platform's components, particularly Key Seeker, which is the core element of the research and can be utilized for cryptography-related studies. A separate document containing notes is provided in the *executables/nRootTag-Seeker.zip/Developer.md* file. This design document offers a detailed plan for implementing cryptographic algorithms across various elliptic curves, with a specific focus on the transition from SECP256K1 to SECP224R1.

The document provides a comprehensive explanation of the GPU-CPU architecture, memory management, and mathematical optimizations, which can be invaluable for researchers and developers working on similar cryptographic implementations. Furthermore, the document thoroughly discusses performance considerations, such as thread management, batch processing, and buffer sizing, offering practical advice for scaling these systems. The explicit mention of potential challenges, including the need to maintain Y coordinates and the consequences of modifying bit-sizes, helps future developers avoid common pitfalls.

The inclusion of Python code examples and detailed explanations of mathematical operations enhances the accessibility and implementability of the concepts presented. Finally, the documentation of specific modifications necessary for different curves, along with the rationale behind each change, provides a roadmap for adapting the system to other elliptic curves, making it a valuable resource for future cryptographic brute-force search and research in GPU-accelerated cryptography.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.

[4] https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp