Powering Privacy: On the Energy Demand and Feasibility of Anonymity Networks on Smartphones

Daniel Hugenroth and Alastair R. Beresford, *University of Cambridge*

https://www.usenix.org/conference/usenixsecurity23/presentation/hugenroth

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 32nd USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

Open access to the Artifact Appendices to the Proceedings of the 32nd USENIX Security Symposium is sponsored by USENIX.

USENIX'23 Artifact Appendix: Powering Privacy: On the Energy Demand and Feasibility of Anonymity Networks on Smartphones

Daniel Hugenroth
University of Cambridge

Alastair R. Beresford
University of Cambridge

A Artifact Appendix

A.1 Abstract

In the submitted paper we present an energy measurement setup for Android smartphones and use it to perform measurements of both individual operations (micro studies) and entire protocols runs (macro studies). In this appendix and our repository we provide instructions for building the required custom hardware, software to perform and analyze measurements, and sample traces. The latter can be used to verify the functionality of the analysis tools without the custom hardware.

Due to the length of the experiments, reproducing all results is not feasible within the scope of the artifact evaluation process. Therefore, we have not applied for the "Results Reproduced" badge. Instead, we chose representative experiments that demonstrate the functionality of all involved components.

Due to the particularities of the hardware prototype the artifact evaluation was performed via an interactive online meeting. In this online meeting we executed all experiments E0, E1, and E2 while sharing our computer screen and capturing the hardware setup with a camera.

A.2 Description & Requirements

We have developed and tested all software on a system running Ubuntu 20.04. The individual programs are written using Rust (serial port logger), Python (data analysis, web service), Android Studio (various apps), and Java (interactive live plot GUI). For all components we provide instructions in the repository for building them from source and give the precise version numbers of the build tools we use. Where possible and helpful, we include Docker files.

Building the hardware requires basic skills in both 3D printing and soldering. Please make sure to follow all necessary safety measures that arise from working with the mentioned equipment and smartphone batteries. If you are interested in testing the software components independently, you can use the provided sample traces in the repository.

A.2.1 Security, privacy, and ethical concerns

HEALTH & SAFETY: While accidents are rare, working with tools and smartphone batteries carries significant risk. Before starting you must familiarize yourself with all applicable safety and compliance requirements. This includes, but is not limited to, departmental, local, federal, and international policies, laws, and regulations. We strongly recommend that you talk to a designated person in your institution that can provide you with the required training and information.

A.2.2 How to access

The repository containing all documentation, software, and other files is available here: `https://github.com/lambdapioneer/powering-privacy/tree/aec-final`. The link points to the tag `aec-final` which references the version used in the artifact evaluation process. We suggest you follow the included `walkthrough.md` file which covers all components and experiments mentioned in this appendix.

A.2.3 Hardware dependencies

The evaluation of the software requires a modern Linux computer with at least 16 GiB RAM, at least 200 GiB of free disk space, a free USB-2.0 port, and a local WiFi network.

For sharing log data from the smartphone with the local computer, the setup requires to run a web service with a IPv4 address that is reachable both from the smartphone and the local computer. The web service can be run on the local computer if its reachable by the smartphone over WiFi.

For building the hardware, you will need a Motorola Moto E6 Plus, a 3D printer, soldering equipment, and more electronic parts as per bill of material. The total costs of all parts including the smartphones are not more than 500 USD.

A.2.4 Software dependencies

The local computer should run a modern Linux distribution with support for the latest versions of Rust, Python, Java, Android Studio, and Docker. Language support might be installed via the package manager or directly using the install scripts from the respective websites. Detailed instructions are included in the repository. We used Ubuntu 20.04 for developing and testing our software.

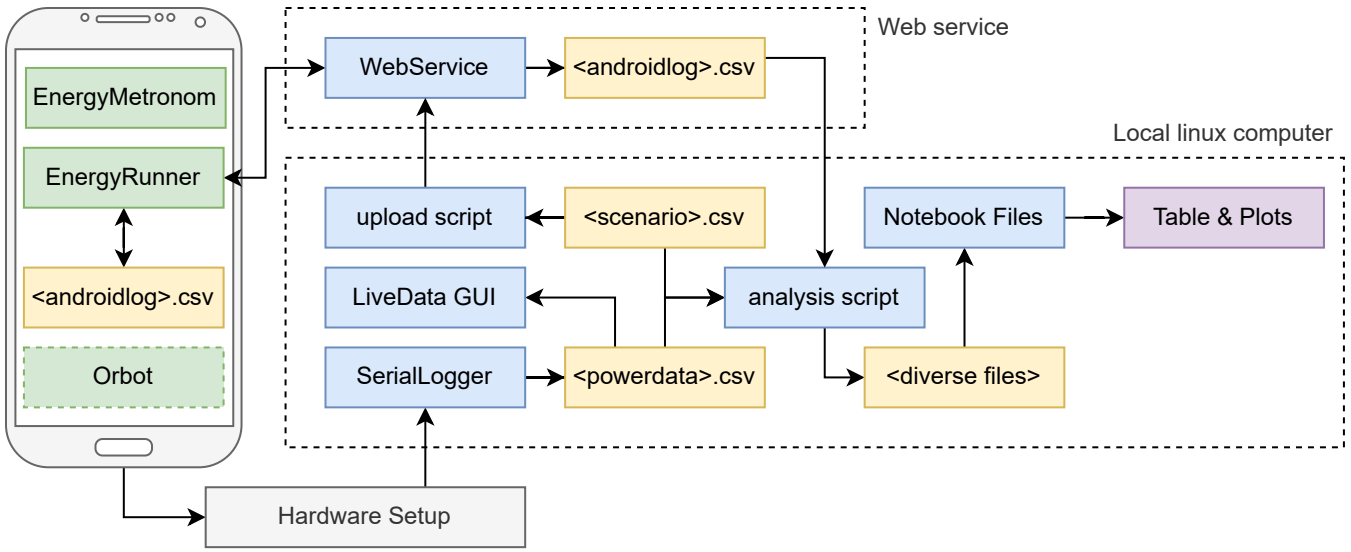

Figure 1: Overview of all components relevant to this artifact appendix. The green boxes are Android app, the blue boxes are Linux executable, the yellow boxes are files, and the purple box is the final result. The arrows indicate the flow of data. The EnergyRunner app both downloads scenario files and uploads its log data.

A.2.5 Benchmarks

Our artifact requires no external data sets. However, we provide sample traces in case the hardware setup is not available to the reviewers.

A.3 Overview of components

In Figure 1 we show all involved components relevant to the experiments described in this document.

A.4 Set-up

We recommend to read all documentation including the walk-through documentation included in the repository before starting.

A.4.1 Installation

Ensure that you are running a compatible Linux distribution, such as Ubuntu 20.04, on your local computer. Clone the repository, checkout tag `aec-final`, and follow the walk-through documentation and the linked `README.md` files to install and build all artifacts. You can skip some of these steps where you already have the required dependencies installed.

A.4.2 Hardware

You can skip this section if you have been provided with a hardware set or plan to only use the sample traces. If you build the hardware yourself, we recommend that you order all required material as early as possible to avoid disappointment due to slow shipping. Building the hardware can be done independently of setting up the software.

A.4.3 Basic Test

The experiment E0 as described in the walk-through document within the repository. It allows to verify that all components, and in particular the hardware setup, are working correctly.

A.5 Evaluation workflow

Our claims cover the functionality of the documentation, hardware setup, and software for both micro and macro studies. While the claims and experiments do not cover all results from our paper, they representatively demonstrate the functionality of the involved components. This is because running all experiments would be unreasonably time consuming for the artifact evaluation.

As described in the paper, many steps cannot be easily automated in order, and require manual operation of the device. For instance, we cannot run a persistent service on the Android device to automate start and stopping the apps, as it would prevent it from reaching its low-power states. We arranged our claims and experiments such that claim there is a one-to-one relationship between C0 and E0, C1 and E1, and so on.

A.5.1 Major Claims

(C0): The documentation and software is complete and allows skilled researchers to perform their own measure-

ments. Users can observe live plots and use data for further analysis. This includes the aforementioned based test in Section A.4.3.

(C1): Using the hardware setup one can perform a micro study that shows that the energy costs for EC operations is lower than for RSA. This reproduces Table 2 on page 8.

(C2): Using the hardware setup one can perform a macro study that shows for access to `.com` using Tor requires more power than doing so directly, but is still feasible in terms of energy costs. This reproduces parts of Figure 9 on page 11.

A.5.2 Experiments

All experiments require the aforementioned hardware and software requirements. If the hardware is not available, the provided sample traces can be used. The given *experiment-hours* provides the blocks of time that experiment runs on the smartphone device and which cannot be interrupted. They are in additional to the *human-hours* that cover software setup and analysis. If the provided sample traces are used, the experiment-time does not apply and the overall required time is reduced drastically. The `walkthrough.md` file in our repository guides you through all experiments step by step.

(E0): *[2-20 human-hours + 1×1 experiment-hour]: Build and test the hardware setup. Building is optional if a hardware kit is provided or the sample traces are used.*
Preparation: Acquire all hardware listed in the documentation. Compile all software as described in documentation. Build and connect the hardware as described.
Execution: Start the serial logger. Start the live logger interface. Turn the screen on and off. Navigate the live plot using keyboard and mouse actions. Stop the experiment. Alternatively, use the `sample-traces/e0/...` files.
Results: The users sees live data changing with the operations on the smartphone. In particular, turning the screen on and off are very visible events.

(E1): *[5 human-hours + 1×2 experiment-hours]: Use the hardware kit to measure individual cryptographic operations.*
Preparation: Start the log collecting web service running on IP *ip*. Update the source code of the Android apps and the analysis scripts with *ip* as per documentation. Upload the scenarios using the Python script. Connect the hardware as described. Install the `EnergyRunner` app and use it to download the scenario.
Execution: Select the scenario file Connect the USB-dongle for clock synchronization. Start the serial logger with the scenario file name. When instructed, turn off the screen and disconnect the USB-dongle. Wait until all operations finished. Stop the serial logger. Stop the app. Alternatively, use the `sample-traces/e1/...` files. Execute the respective analysis Notebook file.
Results: The analysis notebook outputs results that are comparable to those in Table 2.

(E2): *[5 human-hours + 2×1 experiment-hour]: Use the hardware kit to measure longer protocol runs.*
Preparation: Install the Orbot app and enable *Full Connection-Padding* as per documentation. Install the `EnergyMetronom` app and enable "Display over other app" as per documentation. Connect the hardware as described.
Execution: Force-stop Orbot. In the `EnergyMetronom` app select a regular execution of a `.com` website visit and set the interval to every 60 seconds. Start and observe at least one successful website load. Immediately afterwards turn off the screen and start the serial logger. If you like, observe the ongoing experiment using the live logger tool. After 10 minutes stop the serial logger and stop the app. Repeat above steps with Orbot enabled in VPN mode. Alternatively, use the `sample-traces/e2/...` files. Execute the respective analysis Notebook file.
Results: The analysis notebook outputs results that are comparable to the corresponding bars in Figure 9.

A.6 Notes on Reusability

We hope that our hardware and setup will be used by many researchers and fosters new activity in designing, implementing, and evaluating anonymity networks on smartphones. For this we include detailed information on how to measure individual operations as well as longer protocol executions in the linked repository. We are happy to assist others in building copies of our hardware setup and planning experiments. For this please reach-out to us via email.

A.7 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2023/.

Acknowledgements

We would like to thank the chairs and reviewers for the granted special accommodations that allowed our hardware prototype and software to be included in the artifact evaluation process.