SCRAPS: Scalable Collective Remote Attestation for Pub-Sub IoT Networks with Untrusted Proxy Verifier

Lukas Petzi, Ala Eddine Ben Yahya, and Alexandra Dmitrienko, *University of Würzburg;* Gene Tsudik, *UC Irvine;* Thomas Prantl and Samuel Kounev, *University of Würzburg*

https://www.usenix.org/conference/usenixsecurity22/presentation/petzi

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.

A Artifact Appendix

A.1 Abstract

The artifact is composed of 3 directories. The Blockchain directory contains the implementation of Sawtooth blockchain, including smart contracts and 3 entities implemented also in containers. They're namely the administrator (manufacturer in the paper) and 2 clients with roles of Verifer and Prover. The code is deployed using docker and docker-compose. Deploying this part requires a Linux server with 24GB RAM and allows to verify the workflow of the proposed attestation scheme in our paper. The IoT-Clients directory containes the implementation of IoT client in 2 boards (LPCXpresso55S69 + Mikroe WiFi 10 Click and Atmel MEGA-1284P Xplained). The implementation is written in C. The installation and verification of the code helps confirm the low overhead of the attestation process. The Simulation directory contains the code used in the scalability evaluation. It is implemented in Python and helps evaluate the scalability of our proposed scheme against LegIoT.

A.2 Artifact check-list (meta-information)

Obligatory. Fill in whatever is applicable with some keywords and remove unrelated items.

- **Compilation:** AVR-Toolchain, arm-none-eabi, SDK_2.x_LPCXpresso55S69 [API version=2.0.0, Format version=3.8]
- **Run-time environment:** Ubuntu 18 with sudo rights
- **Hardware:** server with 24GB RAM and 16 CPUs, LPCXpresso55S69 + Mikroe WiFi 10 Click and Atmel MEGA-1284P Xplained. They are all publicly available.
- **Output:** console
- **Experiments:** manual steps for each experiment are provided in README file in each directory.
- **How much disk space required (approximately)?:** 20GB
- **How much time is needed to prepare workflow (approximately)?:** 1 hour.
- **How much time is needed to complete experiments (approximately)?:** 3 hours
- **Publicly available (explicitly provide evolving version reference)?:** https://github.com/sss-wue/scraps
  Release tag 1.0.2-beta
- **Code licenses (if publicly available)?:** Licensed under the Apache License, Version 2.0
- **Archived (explicitly provide DOI or stable reference)?:** https://github.com/sss-wue/scraps/releases/tag/1.0.2-beta

A.3 Description

A.3.1 How to access

The source code is publicly available at https://github.com/sss-wue/scraps/releases/tag/1.0.2-beta

A.3.2 Hardware dependencies

The deployment of the blockchain network requires a server with at least 24GB RAM and 16 CPUs. Experiments with the IoT clients require installing the code on the respective boards: LPCXpresso55S69 + Mikroe WiFi 10 Click, Atmel MEGA-1284P Xplained and Atmel ICE.

The artifact is roughly 32MB of size. The deployment requires downloading and running several docker containers. 20GB of disk is sufficient for the deployment and experimenting.

A.3.3 Software dependencies

The deployment of the blockchain requires docker engine and docker compose to be installed. Installing the IoT clients on the boards requires MCUXpresso IDE v11.4.1 [Build 6260] [2021-09-15] and avrdude.

A.3.4 Data sets

N/A

A.3.5 Models

N/A

A.3.6 Security, privacy, and ethical concerns

N/A

A.4 Installation

In order to experiment with the artifact, it is only required to install the compilers and the software dependencies. Each of the boards needs to be connected to a computer using their debugging ports. LPCXpresso55S69 can be linked using a normal mini usb cable while Atmel MEGA-1284P Xplained is linked using Atmel ICE programmer.

A.5 Experiment workflow

The first step in the experiments is the deployment of the blockchain network. A YAML file is provided to automate the process using docker compose. Instructions on executing attestation process are handed in details in the respective README file.

Compiling and installing the IOT Client in LPCXpresso55S69 board is achieved using MCUXpresso software. A video is included in the respective directory showing the steps and the expected results.

Compiling and installing the IOT Client in LPCXpresso55S69 is automated using a Makefile. Used instructions and expected results are all documented and provided in the respective README file.

The simulation is executed using a Python script. Commands and examples of the results are explained in README file.

A.6 Evaluation and expected results

In our paper, we present a blockchain based attestation scheme for IoT devices. Experimenting with the blockchain network allows examining the workflow of the attestation scheme. Different entities are deployed in the roles described in our paper, such as Manufacturer, Prover and Verifier.

Moreover,installing and running the IoT clients on the respective boards prove that COTS IoT devices can benefit from our proposed scheme without any hardware modification.

Besides, running the simulations with different parameters show how our scheme outperforms the state-of-art scheme LegIoT.

A.7 Experiment customization

A.8 Notes

A.9 Version

Based on the LaTeX template for Artifact Evaluation V20220119.