ZeroAUDIT Artifacts

A. Luthra, J. Cavanaugh, H. R. Oclese, R. M. Hirsch, X. Fu
Contact: Dr. Xiang Fu
Xiang.Fu@hofstra.edu

1 Download

`https://github.com/xfu2006/ACSAC20_Artifact`

2 Description

ZeroAUDIT is a zero-knowledge auditing platform for investment funds. The system implements a variety of zero-knowledge proof auditing protocols on the platform of Hyperledger Fabric.

The artifact consists of a VirtualBox image which contains the ZeroAUDIT software (its source code and compiled binary), the data (reported in Section 5 of the paper), and the scripts for reproducing the data.

3 System Requirements

A computer with at least 16GB RAM and CPU virtualization support enabled in BIOS setting.

4 Installation

Download the VirtualBox image (12GB) from the github website listed in Section 1. Import the image in VirtualBox. The password for user `xiang` is `goodyear`. This user is a `sudoer`.

The system is located at `/home/xiang/Desktop/ZeroAudit/v5ZeroAudit`. We refer to this folder as `$ZA_HOME`.

5 Initial Test

The system should be tested initially to verify that the VM guest running speed is acceptable. Hyperledger Fabric will timeout transactions exceeding 30 seconds. This first step is to run a baseline measurement of the CPU speed of the image.

1. Navigate to `$ZA_HOME`.
2. Run script ./run.sh. You will see a message such as "`ERROR. STOP HERE.`" **This is normal** as long as you see the data printed out.

3. The reported performance should be better than the following. Otherwise, the Hyperledger Fabric transactions might be timed out.
   (a) Powmod Exp (base: 3072-bit, exp:256-bit): 750 microseconds
   (b) SECP256r1 Elliptic Curve Op: 18 microseconds

6 Reproduce Experimental Data

Once the VM guest speed test passes, we can proceed to reproducing the experimental data reported in Section 5 of the paper.

1. Edit `zeroAUDIT/src/za_perf/suits/perfAcsac20.go`, identify function `ACSAC2020_Data()`, and recover the rest of function calls (`perfParams()`, `perfPrfs()`, `perfScale1()`, `perfClient()`, `perfScale2()`) that generate the evaluation data.
2. Then run the following:
   (a) `./stop_fab.sh`, to clean up the Hyperledger Fabric framework.
   (b) `./start_fab.sh`, to re-install docker containers after clean up.
   (c) `./compile.sh`, to install ZeroAudit chaincode.
   (d) The above step should take no more than 1 minute. Then run `./run.sh`. This step may cost around 24 hours.

   You will see a message such as "`ERROR. STOP HERE.`" **This is normal** as long as you see the data printed out.
3. Once the data collection completes, the results should be saved to folder `$ZA_HOME/RESULTS`. The following is the breakdown the data files:
   (a) `params.txt`: This corresponds to Table 2: System Parameters.
   (b) `perfExp.txt`: This corresponds to the baseline CPU speed discussed earlier.
   (c) `perfPrfs.txt`: This corresponds to Table 3: Zero Knowledge Proof Cost.
   (d) `perfClient.txt`: This corresponds to Table 4: Hyperledger Fabric Cost.
   (e) `perfScale1.txt`: This corresponds to Figure 3(a),(b): Scalability of Verify Time and Proof Size.
   (f) `perfScale2.txt`: This corresponds to Figure 3(c): Scalability on Security Strength.
4. In folder `$ZA_HOME/ACSAC20_DATA`, we present the data generated by running the same scripts on a bare-metal Linux 18.04 with Intel i5-8350U 1.70GHz CPU.

7 Installation On Bare Metal Linux

It is possible to install the system on a bare metal Linux server. We recommend the following system configurations: Linux 18.04 LTS, GO Lang 1.14 and Hyperledger Fabric 2.0.0. As the system source code has some folders hard-coded, it is recommended to copy the source code at the exact folder location of `$ZA_HOME`, and create a *sudoer* account `xiang`, to avoid any broken configuration.

8 Brief Walk-Through of Source Code

Folder `zeroAudit/src` contains 25k Go-lang source code. Sub-directory `za_zkp` contains the implementation of all the zero knowledge proof protocols presented in the paper. `za_fund` encodes invest funds, price server as clients of Hyperledger Fabric. `za_utils` have implementations of a variety of algorithms such as the Pippenger's multi-product algorithm. Folder `srcfab` contains the Hyperledger Fabric server chaincode of ZeroAUDIT.