Polynomial Commitment with a One-to-Many Prover and Applications

Jiaheng Zhang and Tiancheng Xie, *UC Berkeley;* Thang Hoang, *Virginia Tech;*
Elaine Shi, *CMU;* Yupeng Zhang, *Texas A&M University*

https://www.usenix.org/conference/usenixsecurity22/presentation/zhang-jiaheng

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

In this evaluation, we will allow you to run our experiments on our verifiable secret sharing schemes. One scheme is KZG based with trusted setup, and the one is Virgo based without trusted setup. We will use the c++ standard library chrono to time our execution. You need to read about 20 *lines of C++ code to verify our time measurement.*

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

- **Algorithm: KZG polynomial commitment, Virgo zero-knowledge proofs protocol, FFT**
- **Program: C++ program**
- **Compilation: cmake, and we will provide a bash file for fast setup the environment and fast execution.**
- **Run-time environment: Ubuntu**
- **Hardware: Amazon c5a.24xlarge**
- **Execution: We provide a bash file for execution**
- **Metrics: measure time in seconds**
- **Output: the dealer's (prover) execution time and the verifier's execution time**
- **Experiments: our improved KZG execution, and our virgo based VSS execution**
- **How much disk space required (approximately)?: 20GiB for the OS, we do not require any additional space.**
- **How much time is needed to prepare workflow (approximately)?: 1 hour**
- **How much time is needed to complete experiments (approximately)?: 10 hours**
- **Publicly available?: Yes**
- **Code licenses (if publicly available)?: GPLv3**

A.3 Description

A.3.1 How to access

We put the ssh key in our github repo root directory, named "evss_AE.pem". To access the machine, visit this link:

https://bit.ly/3AFpnwk

If the link fails, use information below:

1. USER NAME: AE_EVSS
2. Password: uNe*g!)0H8pzu=0
3. Access key ID: AKIAWVJ5RUVJDZXGF2X7
4. Secret access key: TcVJ-FocQ/ztXUlYA4HdOu3I9dP/SW8NFo1KuanWu
5. Console login link: https://458079970642.signin.aws.amazon.com/console

With the access link, you can now access our AWS account. At region North California, you should be able to find a machine labeled evss_AE. You can start the machine and find it's IP address. Assuming you have the machien IP, use this command to access:

ssh -i evss_AE.pem ubuntu@IP

A.3.2 Hardware dependencies

Amazon AWS c5a.24xlarge, 20Gib disk space.

A.3.3 Software dependencies

libgmp, ate-pairing, xbyak

All dependencies will be installed via our provided script: dependency.sh

A.4 Installation

Obligatory. Describe the setup procedures for your artifact targeting novice users (even if you use a VM image or access to a remote machine).

In the home directory, you should be able to find a folder named eVSS. This folder contains all needed files to run the experiment. Dependencies are pre-installed.

A.5 Experiment workflow

The machine should be ready to directly run the experiment. At the eVSS directory, you can run "./compile.sh" to compile the whole project. (It's already pre-compiled for you, but in case you want to compile it, you can run this command.)

Then to run the experiment for improved KZG-based VSS, run "./trusted_setup_version.sh"

To run the experiment for transparent VSS, run "./transparent_version.sh"

We will only run experiment for verifiable secret sharing. Let t_p be the VSS's prover time, and t_v be the VSS's verification time, you can calculate the DKG time for n parties by the following formula:

$$\mathsf{DKGtime} = (2n \times t_v) + 2 \times t_p$$

A.6 Evaluation and expected results

1. For the KZG-based VSS, we claim:
   (a) the running time for 2^{10} players is 3 second
   (b) the running time for 2^{15} players is 100 second
   (c) the running time for 2^{20} players is 4000 second
   (d) the verification time is constant, 0.001 second
   (e) the proof size is constant, 192 Byte
2. For the transparent VSS, we claim:
   (a) the running time for 2^{10} players is 0.2 second, proof size 223840 Bytes, and verification time 0.003 second.
   (b) the running time for 2^{15} players is 8 second, proof size 324832 Bytes, and verification time 0.004 second.
   (c) the running time for 2^{20} players is 300 second, proof size 467520 Bytes, and verification time 0.01 second.

A.7 Note

For final stable URL, visit

https://github.com/sunblaze-ucb/eVSS/tree/e8f1cd4d6ef086b2ae017ed56560328fdffec491