TreeSync: Authenticated Group Management for Messaging Layer Security

Théophile Wallez, *Inria Paris;* Jonathan Protzenko, *Microsoft Research;*
Benjamin Beurdouche, *Mozilla;* Karthikeyan Bhargavan, *Inria Paris*

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

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

USENIX'23 Artifact Appendix: TreeSync: Authenticated Group Management for Messaging Layer Security

Théophile Wallez
Inria Paris

Jonathan Protzenko
Microsoft Research

Benjamin Beurdouche
Mozilla

Karthikeyan Bhargavan
Inria Paris

A Artifact Appendix

A.1 Abstract

The artifact contains an executable specification of MLS, and proofs for its TreeSync sub-protocol. It also contains code to run the official MLS test vectors.

A.2 Description & Requirements

Running this artifact requires a computer with either Nix or Docker installed. With Nix, the computer architecture must be x86_64.

A.2.1 Security, privacy, and ethical concerns

None.

A.2.2 How to access

https://github.com/Inria-Prosecco/treesync/tree/7ea27ead0abc4e6bf47033f35a7eada233ac244e

A.2.3 Hardware dependencies

A computer with 8GB of RAM is enough to build and run the artifact.

A.2.4 Software dependencies

All the dependencies are managed by Nix or Docker, hence Nix or Docker are the only dependencies required to build the artifact.

A.2.5 Benchmarks

As mentioned in the paper (section 5, table 2), we do run some benchmarks. The benchmarks requires no additional data, and they are executed at the same time as the tests, when running the `make check` command.

A.3 Set-up

A.3.1 Installation

With Nix:

```
# This command will compile Z3, F* and
# other dependencies to the correct
# version, and start a shell with the
# correct environment.
nix develop
```

With Docker:

```
# Build the docker image.
# This will compile Z3 and F* to the
# correct version.
docker build . -t treesync_artifact
# Start the image and start a shell with
# the correct environment.
docker run -it treesync_artifact
```

A.3.2 Basic Test

In a shell with the correct environment:

```
cd mls-star
# This command will verify MLS*
make
# This command will run tests of MLS*
make check
```

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): MLS* can be used as a reference implementation of MLS. This is proven in the experiment (E1) that runs the official test vectors.

(C2): The security of TreeSync is formally proved. This is proven by experiment (E2) where we link theorem stated in the paper, and formal theorems in F*.

(C3): Doing formal proofs for TreeSync allowed us to propose improvements to the standard. This is proven in the experiment (E3) where we give links to pull-requests.

A.4.2 Experiments

(E1): [15 human-minutes + 1 compute-hour + 4 GB disk]: Run official MLS test vectors.
Preparation: Set up a correct environment, either with Nix or Docker.
Execution: Go in the `mls-star` repository and launch `make check`.
Results: The tests should succeed. For each test vector, it should print something similar to:
Secret Tree: running tests for 2/7 ciphersuites...
Secret Tree: success!

(E2): [30 human-minutes + 1 compute-hour + 4 GB disk]: Check that the formal theorems correspond to the prose.
Preparation: Set up a correct environment, either with Nix or Docker.
Execution: Go in the `mls-star` repository and launch `make`: this will verify with F* all of our theorems. Every theorem mentioned in the paper is listed in the README file, organized by section, with a link to the corresponding source code. Check that they correspond to the prose.
Results: the theorems mentioned in the paper and the formal theorems should correspond.

(E3): [30 human-minutes + 0 compute-minutes]: Look at our MLS pull-requests.
Preparation: Start your favorite web-browser.
Execution: Compare section 6 of our paper, and the following pull-requests:

- (disambiguate signatures) https://github.com/mlswg/mls-protocol/pull/526
- (use tree-hash in parent-hash) https://github.com/mlswg/mls-protocol/pull/527
- (strengthen the parent-hash link) https://github.com/mlswg/mls-protocol/pull/713

Results: the pull-requests and our claims of standard improvement should match.

A.5 Notes on Reusability

This artifact can be used for various purposes.

As a reference implementation, it can serve as a companion to the standard to help understanding it, and understand the security guarantees given by TreeSync.

As a proved specification, it can serve to test changes to the protocol, by modifying the F* specification and update the TreeSync Authentication Theorem (section 4.5).

A.6 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/.