USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Treebeard: A Scalable and Fault Tolerant ORAM Datastore

Amin Setayesh, Cheran Mahalingam, Emily Chen, and Sujaya Maiyya, *University of Waterloo*

https://www.usenix.org/conference/usenixsecurity25/presentation/setayesh

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.

USENIX Security '25 Artifact Appendix: Treebeard: A Scalable and Fault Tolerant ORAM Datastore

Amin Setayesh Cheran Mahalingam Emily Chen Sujaya Maiyya
University of Waterloo

A Artifact Appendix

This artifact provides a functional implementation of Treebeard. Treebeard is implemented in Go and includes all the necessary components to demonstrate the core features of the system, including its multilayer architecture (router, stash, and ORAM layers), batching mechanisms, and fault-tolerant coordination using the Raft consensus protocol. The source code is publicly available at: https://figshare.com/articles/software/treebeard_zip/29230676?file=55094552.

A.1 Abstract

This artifact provides a working implementation of Treebeard, a scalable and fault-tolerant ORAM datastore. It includes the full source code, configuration, and instructions for running the system and observing its core functionality.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

This artifact does not involve any real user data, sensitive information, or interaction with external systems beyond local or cloud-based test deployments. All experiments are performed using synthetic workloads. There are no known security, privacy, or ethical concerns associated with running or testing this artifact.

A.2.2 How to access

The source code is publicly available at: https://figshare.com/articles/software/treebeard_zip/29230676?file=55094552.

A.2.3 Hardware dependencies

The system can be run on any general-purpose machine. No specialized hardware (such as SGX, GPUs, or TPUs) is required.

A.2.4 Software dependencies

To run the artifact, you need the following installed locally: Go version 1.20 and Ansible. You will also need access to an Ubuntu server. The Ansible scripts provided will automatically install all required packages and configure the environment on the server. No additional manual setup is necessary beyond running the Ansible playbook.

A.2.5 Benchmarks

A sample workload is provided with the artifact to test the functionality of the system. This workload uses a YCSB-style client to issue GET and PUT operations and verify that Treebeard handles requests correctly. No additional datasets or external inputs are required.

A.3 Set-up

A.3.1 Installation

Install Go v1.20 and Ansible using official instructions.

A.3.2 Basic Test

Please follow the instructions under the "Example Experiment" section of the README.md file in the artifact repository. It provides all necessary installation and configuration steps to prepare the environment.

A.4 Evaluation workflow

A.4.1 Major Claims

None.

A.4.2 Experiments

None.

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.