Keystone Project Proposal (derived from commit e8277a4)

1. Name of Project

Keystone

or

The Keystone TEE Framework

2. Project Description (what it does, why it is valuable, origin and history)

Keystone is a software framework for building TEEs on commodity RISC-V platforms.

Keystone provides software components, optional modules and extensions, SDKs, and more for both system designers and application developers building trusted systems on RISC-V. Our goal is to provide the standard set of tools for all trusted systems projects using RISC-V.

The Keystone project was started at UC Berkeley to develop new designs for TEEs, specifically for use on RISC-V platforms. At that time, there was no general-purpose TEE solution for RISC-V. As Keystone developed, we recognized and capitalized the potential for a generic TEE framework, compatible with a wide variety of RISC-V hardware, threat models, and use cases.

3. How does this project align with the Consortium's Mission Statement

Keystone naturally aligns with the CCC's mission. We provide tools for customizing and specializing novel TEE designs by everyone involved in the lifecycle, from hardware designers to application developers. These tools encourage rapid development, prototyping, and deployment of novel TEE approaches on a wide variety of RISC-V platforms.

4. High level assessment of project synergy with existing projects under the CCC

Keystone directly synergizes with all TEE SDK/middleware efforts such as OpenEnclave. For these projects, Keystone provides another back-end similar to TrustZone or SGX.

We see little overlap with current projects, which are more focused on the 'front end' for TEEs and applications. Keystone instead provides a new set of targets for these projects to integrate with.

5. Describe the Trusted Computing Base (TCB) of the project.

A Keystone TEE has several parts, the core component being the security monitor (SM). Our SM has a TCB of ~2000 LoC (depending on configuration). We then rely on a build of either the Berkeley Boot Loader (bbl) or (once the port is complete) the OpenSBI. We then also rely on a root-of-trust in the target RISC-V hardware platform and a source of randomness.

An application running inside an enclave in Keystone depends on the SM TCB, as well as any additional code the application adds via its runtime component (e.g. via our Eyrie runtime).

Our technical paper has a more detailed breakdown of our TCB and trust model.

6. Project website URL

https://keystone-enclave.org

7. Project Code of Conduct URL.

Our current CoC is based on CC v1.4 as that was the current version when we adopted it. CC v1.4

8. Source control URL

https://github.com/keystone-enclave

9. Issue tracker URL

https://github.com/keystone-enclave (see each respository) Issues are tracked in the relevant repository. See https://github.com/keystone-enclave/keystone for higher-level issues.

10. Project Logo URL or attachment (Vector Graphic: SVG, EPS)

Logo attached to email.

11. Project license.

We use a BSD 3-clause license. As is standard for projects out of UC, it is copyright The Regents of the University of California.

12. External dependencies (including licenses)

We use a number of small open-source crypographic implementations directly imported (with relevant licensing attached).

We also use, but are not required by the core of Keystone:

- musl-libc (as an optional library for applications)
- gcc (RISC-V cross compiling)
- buildroot
- Linux

13. Standards implemented by the project, if any.

No specific standards are associated with the Keystone implementation.

14. Release methodology and mechanics

We use basic Travis CI on all Keystone repositories. With funding we expect to expand our CI significantly.

Keystone has not yet reached a "1.0" release. Current releases are handled via Github's releases mechanism, and are generally based on when compatibility breaking changes to builds/etc are introduced.

Post-1.0 and post eventual integration with a commercial project, we expect to revamp and standardize our releases.

We do not distribute any binary builds, nor do we expect to.

15. Names of initial committers, if different from those submitting proposal

Same as submitting. (Primary initial developers were Dayeol Lee and David Kohlbrenner)

16. List of project's official communication channels

Discussion list: https://groups.google.com/forum/#!forum/keystone-enclave-forum Announcements: https://groups.google.com/forum/#!forum/keystone-enclave

17. Social media accounts

https://twitter.com/KeystoneEnclave

18. Existing financial sponsorship

No direct sponsorships. Primary developers are UC Berkeley PhD students or Postdocs, and are funded by the RISE and ADEPT labs at UC Berkeley.

19. Trademark status

None

20. Project Security Policy

As we have not reached a 1.0 release, we accept all security-related disclosures through our public bug reporting mechanisms.

21. Preferred maturity level (Sandbox, Incubation, Graduation, or Emeritus)

Sandbox

22. Any additional information the TAC and Board should take into consideration when reviewing your proposal.

Keystone is a rapidly growing project with significant academic contributions. There are numerous academic groups already publishing about or using Keystone as a basis for TEE work. We are not, however, limited to academic impact, and we regularly coordinate with several industry partners on potential product applications.

We also organize and host the Open-Source Enclaves Workshop (2019: https://keystone-enclave.org/open-source-enclaves-workshop/) around a variety of secure enclave/TEE topics.