MPK-Based In-Process Crypto Key Vault : Deliverables

November 4, 2025

Deliverables

Each student is expected to deliver all the components mentioned below. Both phases will be evaluated during the same evaluation session. The separation into phases is only intended to provide a minimal roadmap and clarify the development milestones. All functionalities from both phases will be examined together during the final evaluation slot.

Phase 1

- You should have the code that you have been working on.
- Create a **trusted shared library** that contains all the cryptographic functions.
- The cryptographic functions should use specific keys that are stored in designated virtual memory pages.
- These memory pages must be protected using **Memory Protection Keys (MPK)**.

Only the trusted library should have access to these keys (no other modules, including the main binary) should be able to access them.

We will provide a set of test cases, which will be linked against your shared library. The library must function correctly with our test programs, and no other compilation unit apart from the trusted library should be able to access or manipulate the keys.

Phase 2

In this phase, we will evaluate how your system behaves when `libmpk` APIs are used in the **untrusted** part of the code.

- We will provide test cases that cover various misuse scenarios.
- Your instrumentation should detect and eliminate all occurrences of instructions or functions such as `wrpkru`, `pkey_set`, or any other function responsible for changing access permissions in untrusted code.
- You must choose one specific function or instruction (for example, `pkey_set` or `wrpkru`) that your instrumentation will monitor or restrict.
- Clearly mention your chosen function/instruction in the **README** file of your submission.

This will allow us to prepare the test cases in the required format for your implementation.