Open Enclave SDK

Simon Leet and Andy Schwartzmeyer, Microsoft

github.com/openenclave/openenclave
Control access to data through its lifecycle

### Existing

**At rest**
Encrypt inactive data when stored in blob storage, database, etc.

Example include:
- dm-crypt (block device encryption)
- SQL TDE (transparent database encryption)

### In transit
Encrypt data that is flowing between untrusted public or private networks

Examples include:
- TLS

### New

**In use**
Protect/Encrypt data that is in use during computation

Examples include:
- Trusted Execution Environments such as Intel SGX & ARM TrustZone
- Homomorphic encryption
Trusted Execution Environments (TEEs)

Protected memory space:
- Secure portion of processor & memory
- Only authorized code runs and is permitted to access data
- Code & data cannot be viewed or modified from outside

Supports proving of identity both locally and remotely (attestation)

Supports persistence of secrets (sealing)

Example of hardware-based TEE
What is Open Enclave SDK?

- **Low-level C/C++ SDK for Confidential Computing Apps**
- **Fully open-source and transparent, not Azure-specific**
- **Write once, run across technologies: SGX, TrustZone**
- **Support for multiple OS platforms: Linux, Windows, OP-TEE OS**
Building the community

- Publicly available Open Source Software project since Oct 2018
  [https://github.com/openenclave/openenclave](https://github.com/openenclave/openenclave)
  - Over 2,000 commits and 4 releases since date of publish
  - Over 40 contributors, 70 forks, 210 stars, and ~1,200 pull requests
  - Continuously active project with ~25-100 commits a week
Part of the Confidential Computing Ecosystem

App Process

Enterprise blockchain application

Confidential Computing Blockchain Framework

App Enclave

Confidential Computing Blockchain
MUSL libc | LLVM libcxx | mbedTLS
Open Enclave runtime

Open Enclave host libraries

Intel SGX Enclave Common

Intel SGX DCAP Libraries

Architectural enclave

Kernel mode

Intel SGX driver
Open Enclave Features

- Provides generalized enclave creation, invocation and attestation methods.
- Includes mbedTLS, MUSL libc, LLVM libcxx libraries by default. Componentized to allow developers to replace these libraries.
- Basic POSIX support for thread and memory primitives. Optional components for I/O to secure file system and sockets.
- Growing syscall-handling infrastructure to support containerized applications with minimal modification.
- Supports Intel SGX on Linux and Windows, ARM TrustZone on OP-TEE
Questions?

github.com/openenclave/openenclave

Simon Leet, @CodeMonkeyLeet
Andy Schwartzmeyer, @andschwa