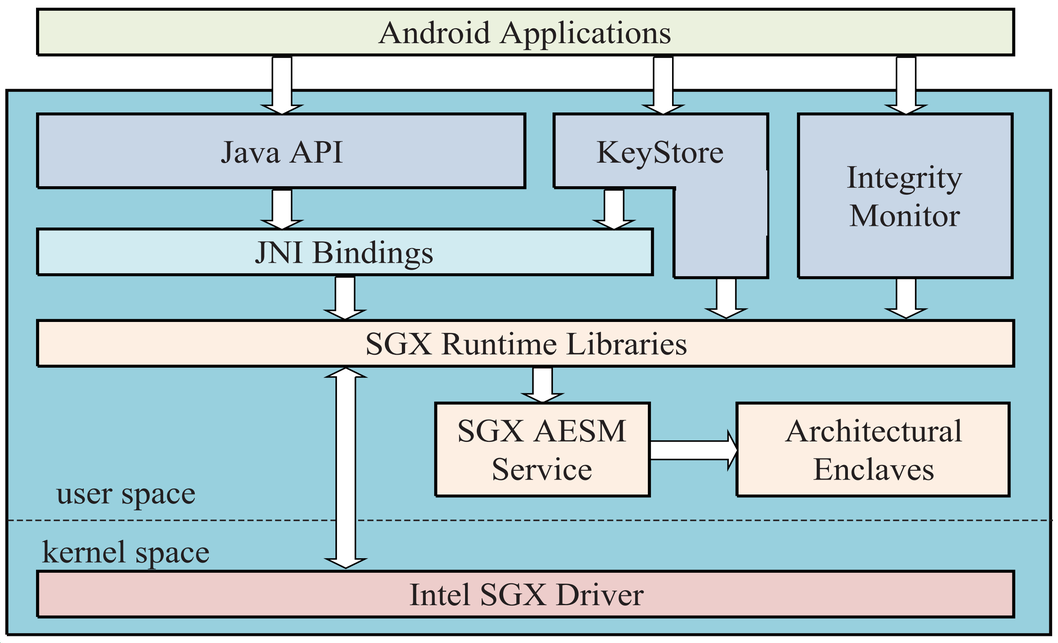
# Andromeda: A Trusted Execution Framework for Android Based on Secure Enclaves

**Dimitrios Karnikis, Sotiris Ioannidis**
**Distributed Computing Systems Laboratory,**
**Institute of Computer Science, Forth Heraklion Greece**
**{dkarnikis, sotiris}@ics.forth.gr**

## 1. Introduction

Andromeda, is a framework that provides native Intel SGX support for Android x86_64 OS with hardware support. Andromeda offers the first SGX interface for Android and enhances its cryptographic system with enclave support. It offers secure execution of cryptographic functions, a Java API named Vault(), that gives control to enclave, an SGX-Android compliant toolchain and a kernel integrity monitor.

## 2. Andromeda control flow




1. Android applications that perform common calls to the Android System. These are indirect call to the Intel SGX features that reside on the Android System.
2. Perform calls on Java API, Keystore functions or the Integrity Monitor. These events occur inside the Java binder.
3. Using the Java API provided by Andromeda, the flow is handled through JNI bindings to enter the enclave mode.
4. The SGX calls make use of the libsgx_uae_service and libsgx_urts that run natively on Android OS.
5. AESM service validates the integrity of the calls and allows access to the hardware(does the same work as on Unix Systems).
6. Data between transfer is completed between user and kernel space using the Andromeda driver for Intel SGX.

## 3. Andromeda Vault API

| **Constructor Summary** | |
|---|---|
| **Constructor** | **Description** |
| Vault() | Instantiate an enclave class |
| **Type** | **Method Description** |
| int | **store**(byte[] data) Stores the data and returns its index |
| byte[] | **retrieve**(int index) Retrieves the data using its index |
| void | **seal**(String path) Seals the enclave data and stores to file-system |
| void | **unseal**(String path) Unseals the data and populates the enclave |
| void | **destroy**() Destroys the secure enclave |

## 5. Conclusions

- First SGX interface for Android OS
- An SGX API available in native C/C++ and Java JNI bindings allowing developers to integrate SGX in their applications
- Services that enhance the security of Android
- A fully compliant Android SGX cross-compiler ready to use

## 6. References

[1] https://github.com/intel/linux-sgx.

[2] https://www.crystax.net/android/ndk/.

[3] https://github.com/intel/linux-sgx-driver/.

[4] https://developer.arm.com/technologies/trustzone.

[5] https://software.intel.com/en-us/sgx.

## 4. Evaluation

We implemented basic cryptographic functions for the Android Keystore Service like AES-128 CTR and RSA encryption and decryption. As a result, Android applications that use such functions, will make use of the secure enclave mode of Intel SGX.

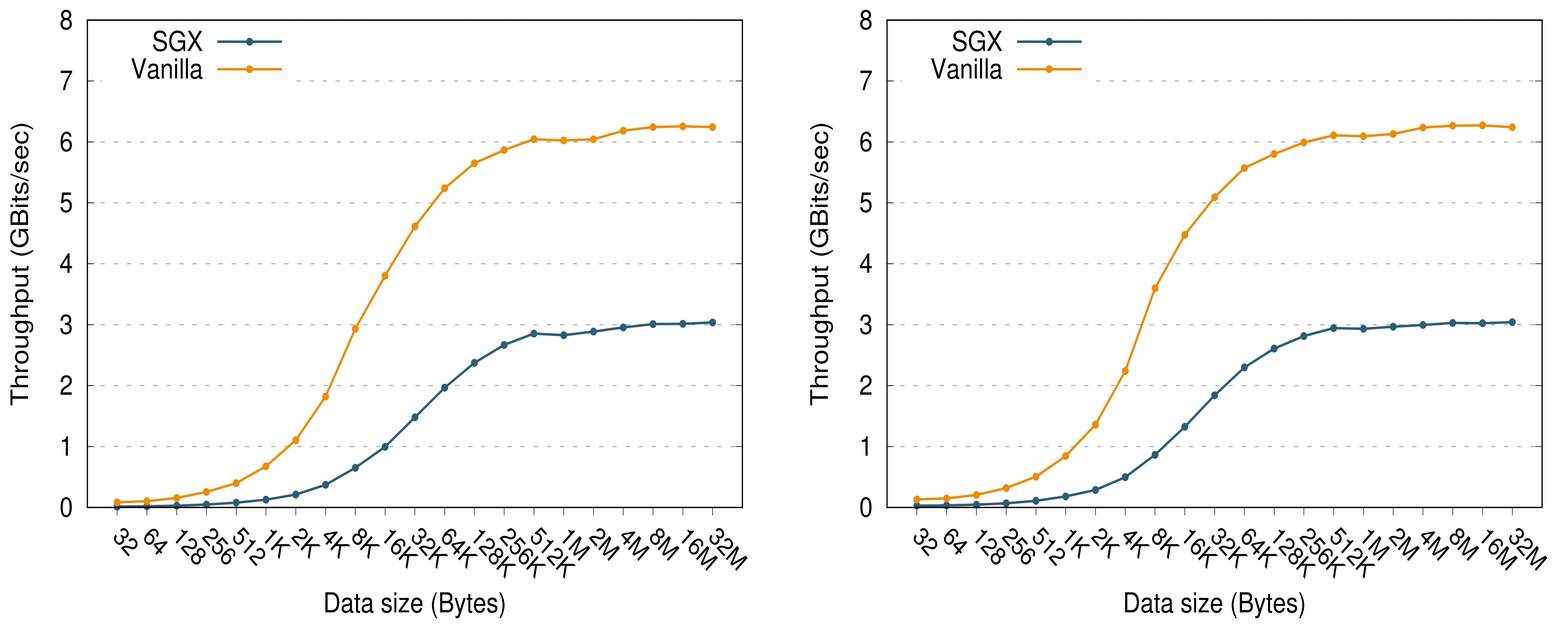



Figure 1: Throughput comparison between the vanilla implementation of AES-128 CTR found in Android's Keystore system and the SGX-enabled implementation provided by Andromeda's Keystore system, depending on the size of data being processed. The SGX-enabled implementation introduces 51%-84% overhead to encryption operations and 51% to 78% to decryption.

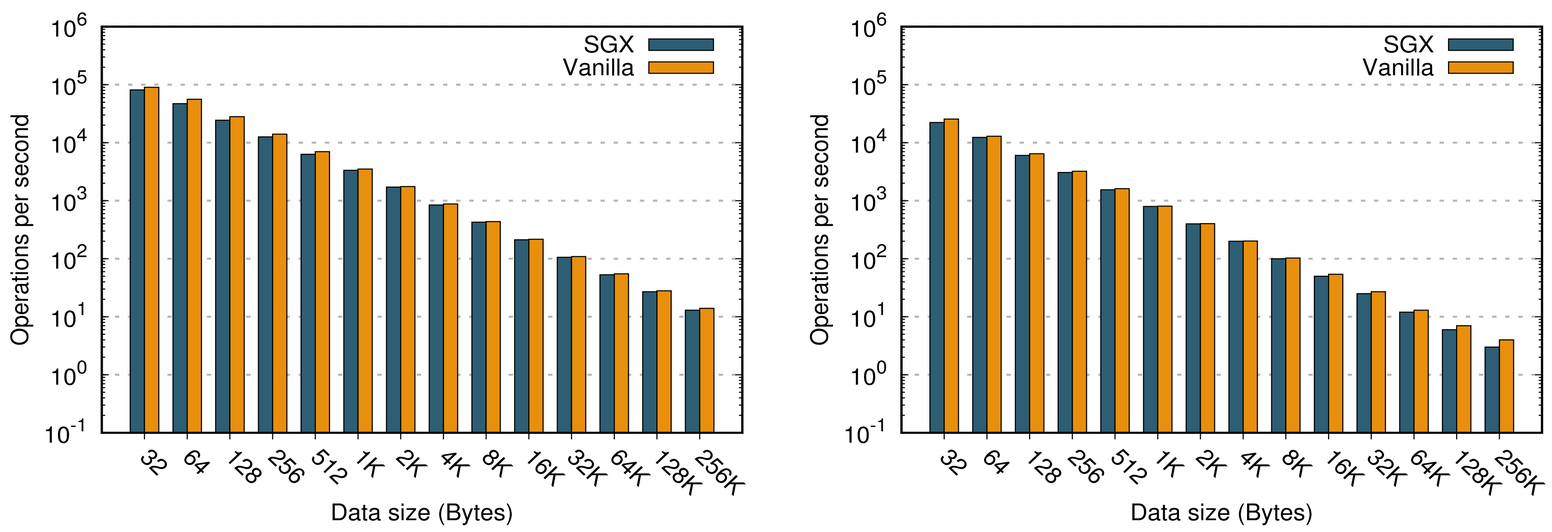



Figure 2: Sustained throughput achieved for the vanilla and the SGX-enabled implementation of the RSA-1024 cryptographic operations in respect to the input data size. The SGX-enabled implementation introduces 0.9%-25% overhead in both encryption and decryption operations.

## 7. Acknowledgements

This work was supported in part by the European commission through the project CIPSEC under Grant Agreement No. 700378