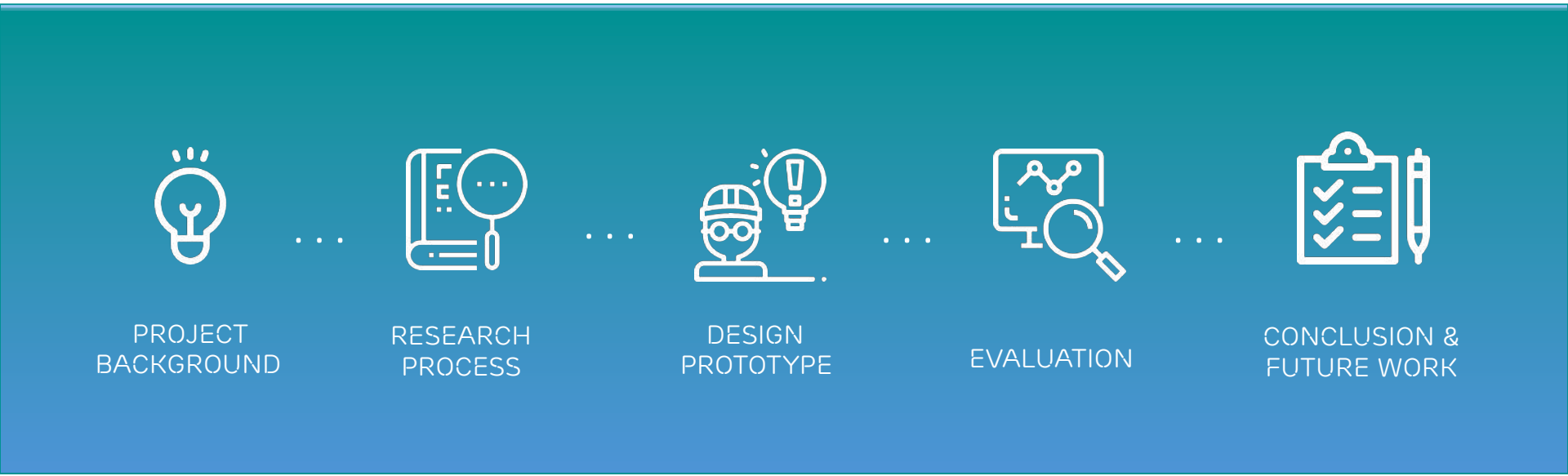
IMPROVING THE SECURITY OF KMS ON A CLOUD PLATFORM USING TRUSTED HARDWARE

Nov, 15, 2018

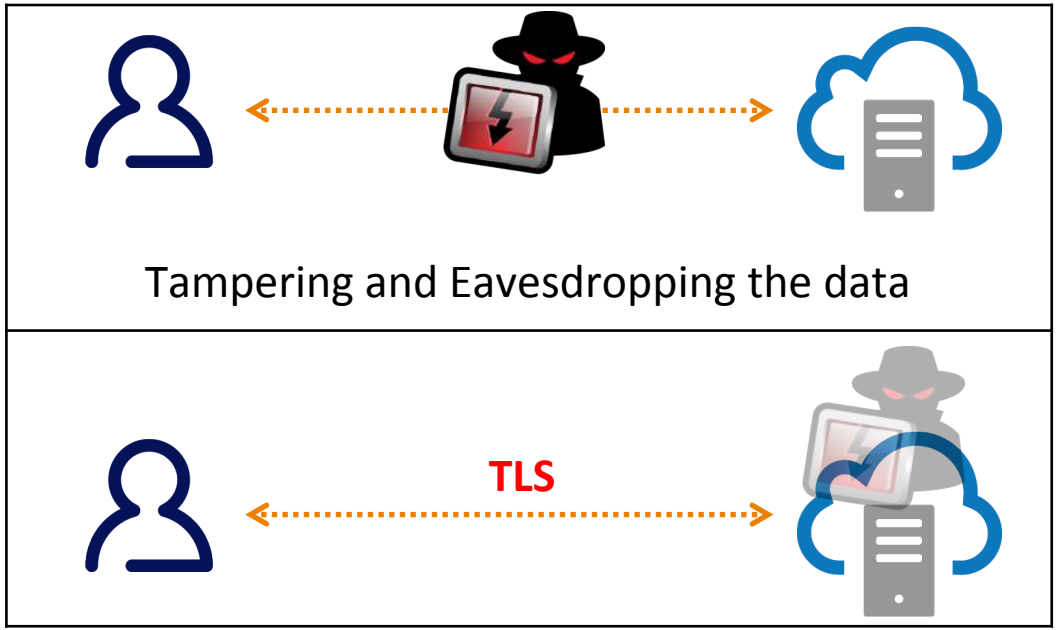
Shan Kuan


PROJECT BACKGROUND
RESEARCH PROCESS
DESIGN PROTOTYPE
EVALUATION
CONCLUSION & FUTURE WORK

Type of Attacks

- Outside Attack
- Inside Attack

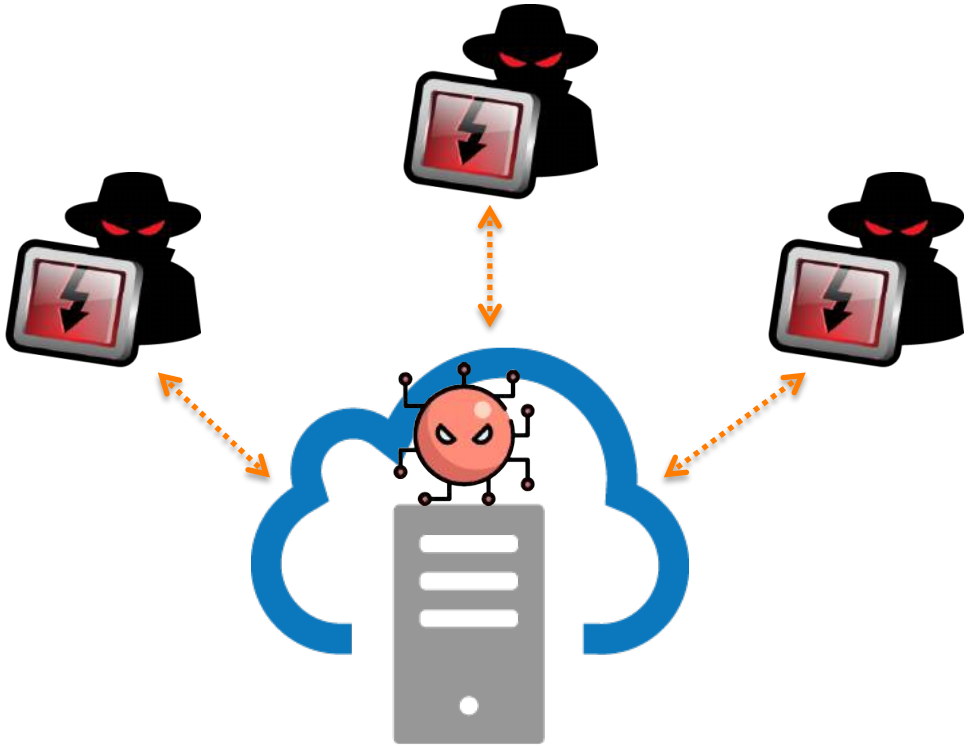
Outside Attack



Type of Attacks

- Outside Attack
- Inside Attack

Inside Attack

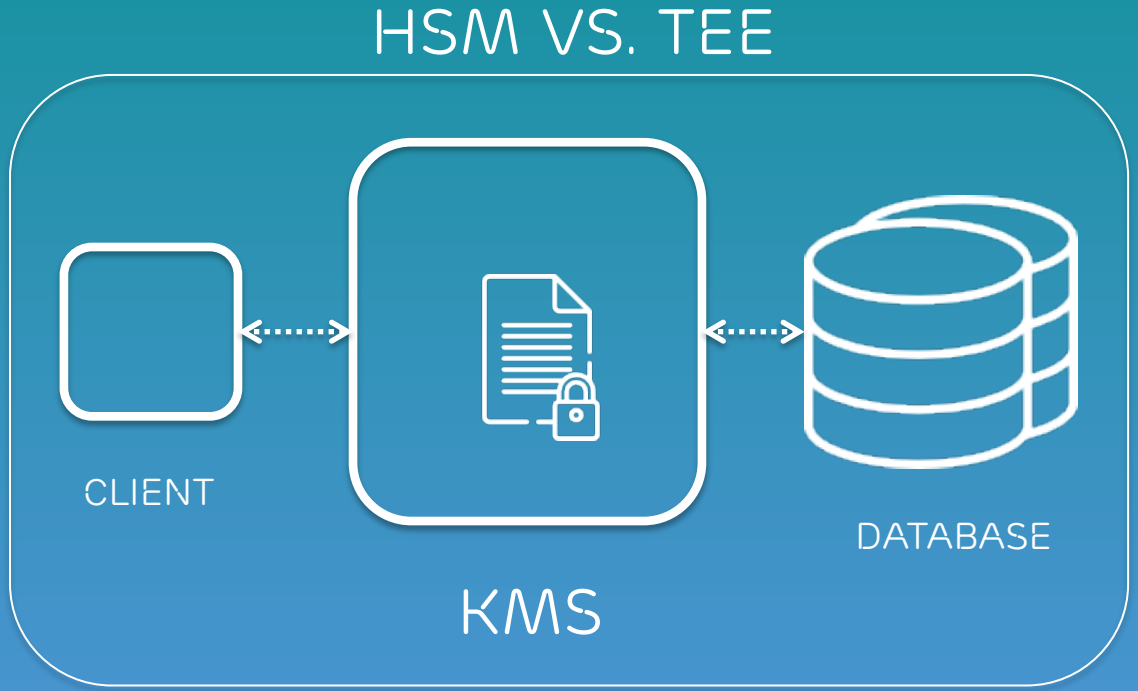


Tampering and Eavesdropping the data

Approach Overview:

- **OpenStack-Barbican with SGX:**
  Plugin TEE technology into KMS (Key Management Service)

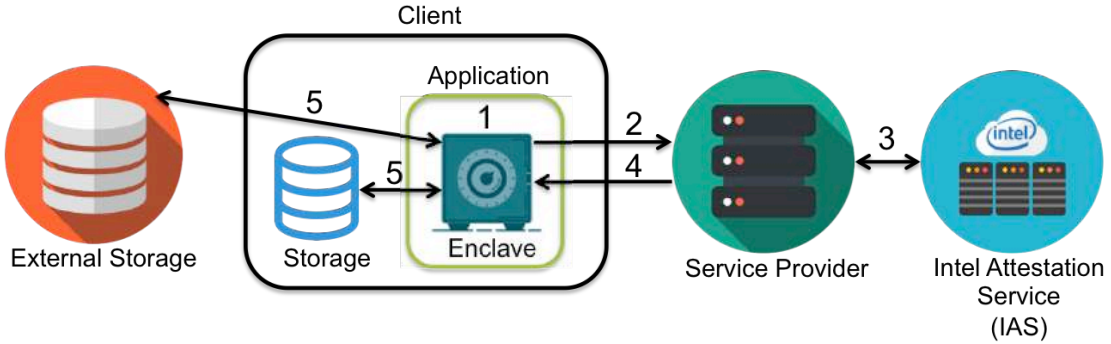
Approach



Intel SGX Mechanism:

- Enclave
- Attestation
- Sealing

Lifecycle of Intel SGX Enclave



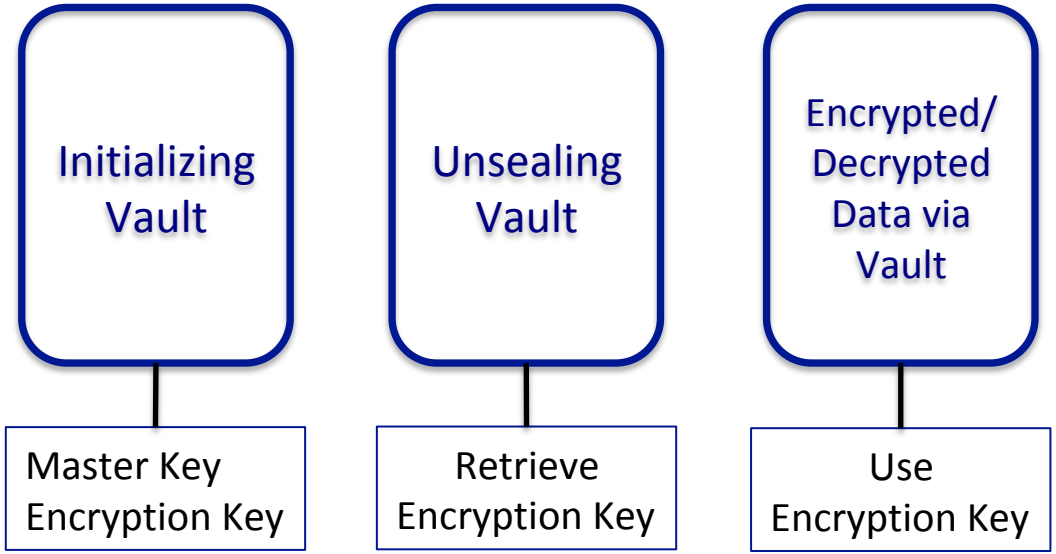
1. Enclave Launch
2. Attestation
3. Verification
4. Provisioning
5. Sealing/Unsealing

Research Process

Vault:

- Pluggable Backend Architecture
- Open Source
- AES-GCM for Data Being Stored
- Build for General Purpose
- Premium Service: HSM

Three Important Stages of Vault

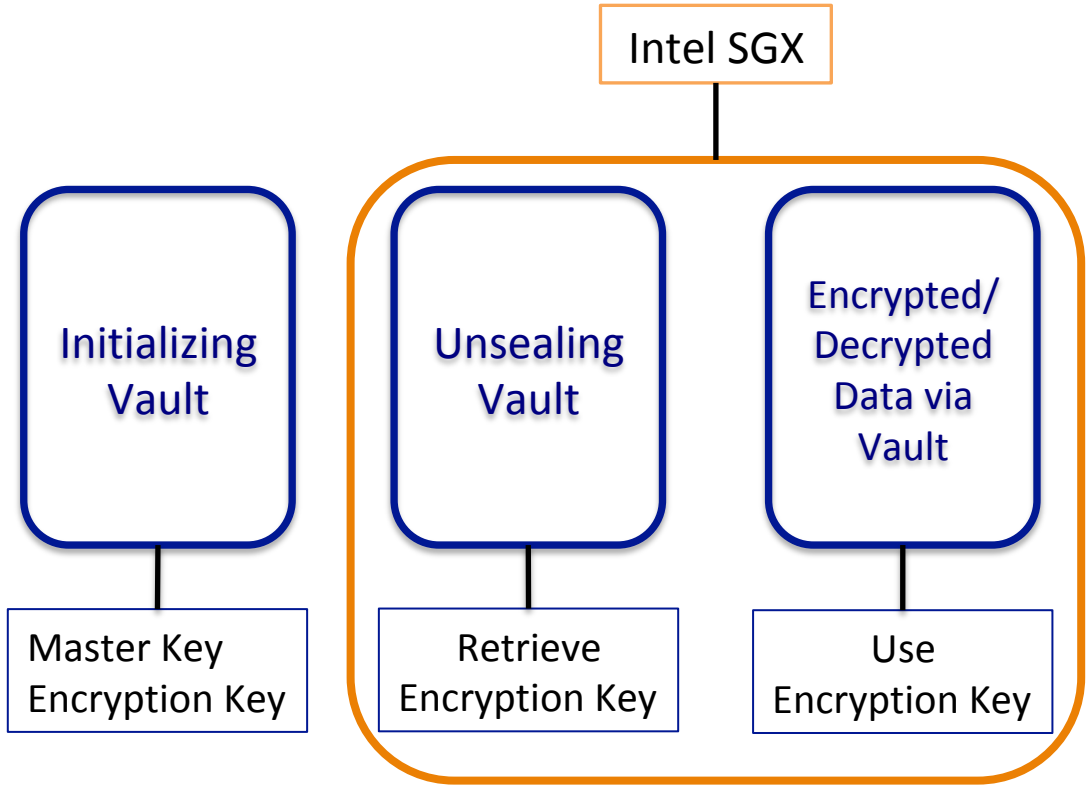


Research Process

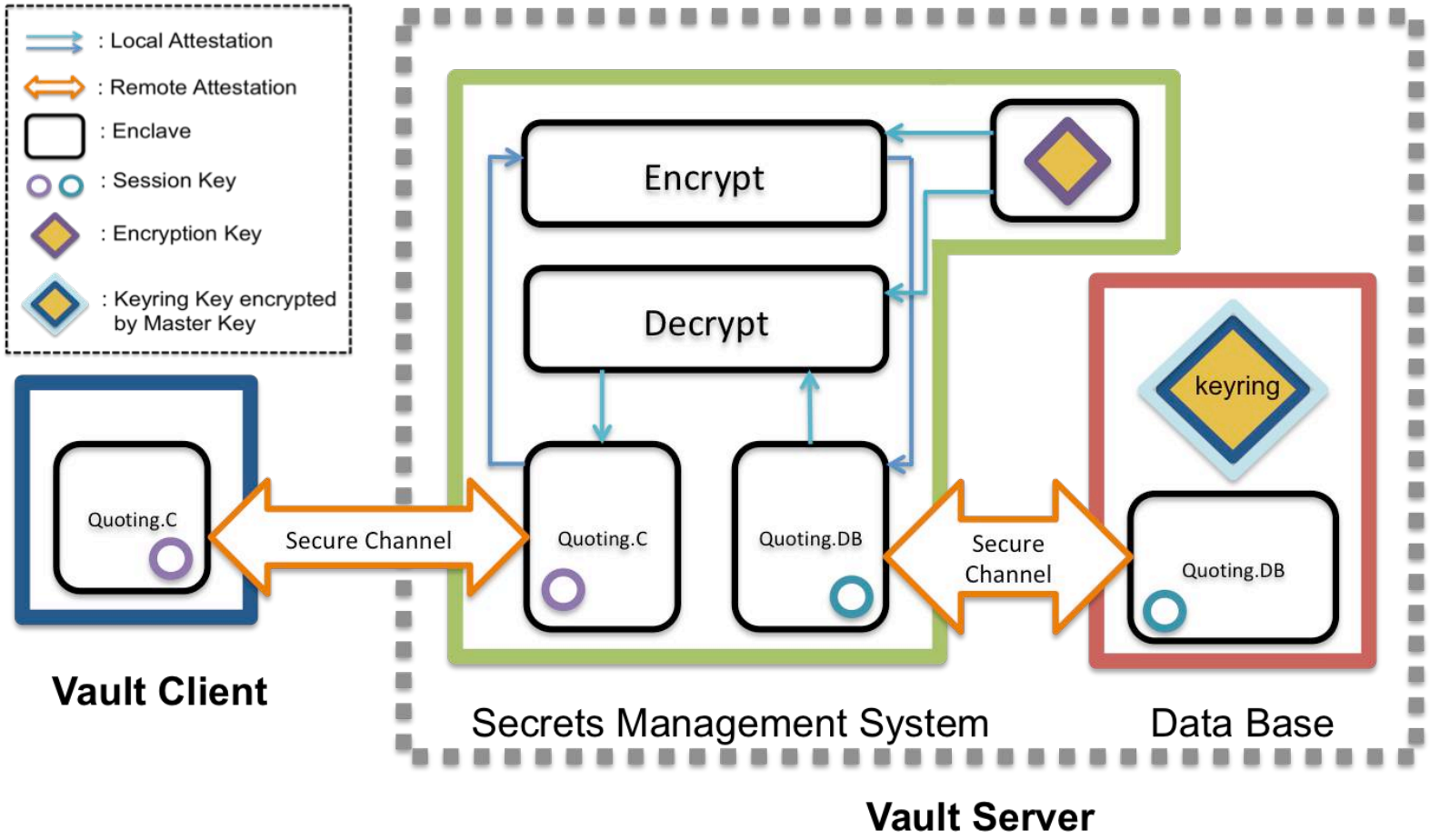
Vault:

- Pluggable Backend Architecture
- Open Source
- AES-GCM for Data Being Stored
- Build for General Purpose
- Premium Service: HSM

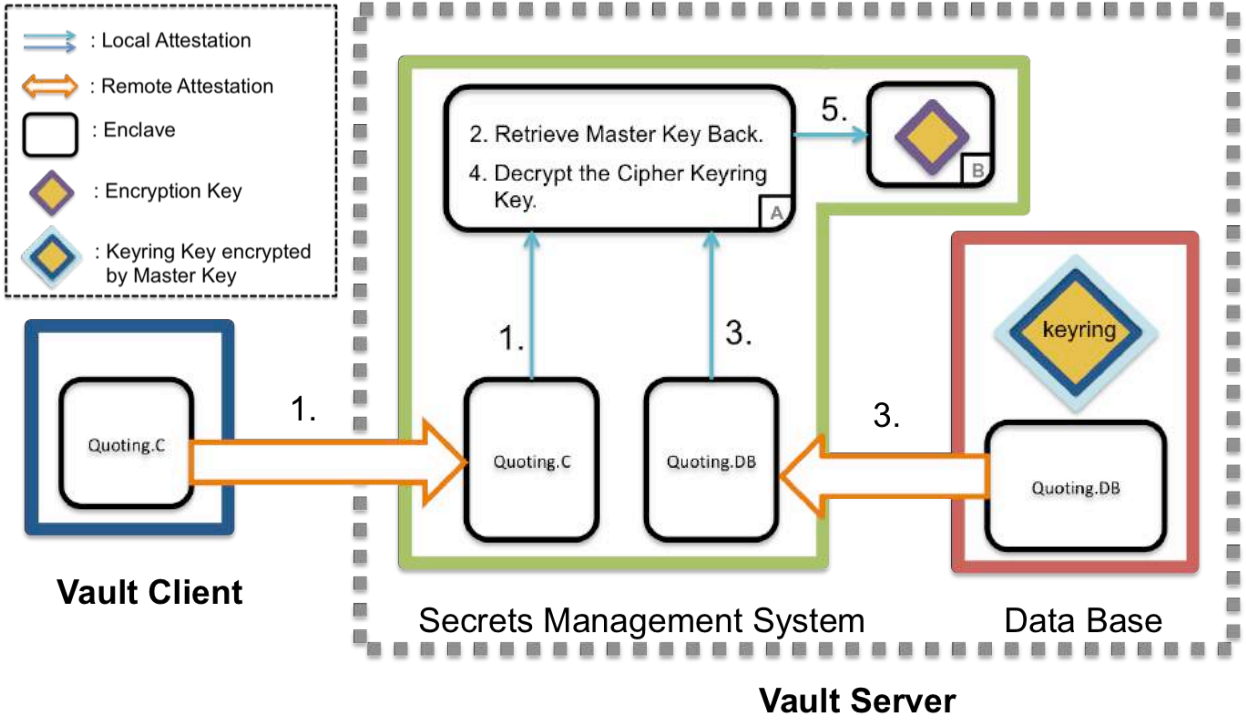
Three Important Stages of Vault



Vault-SGX

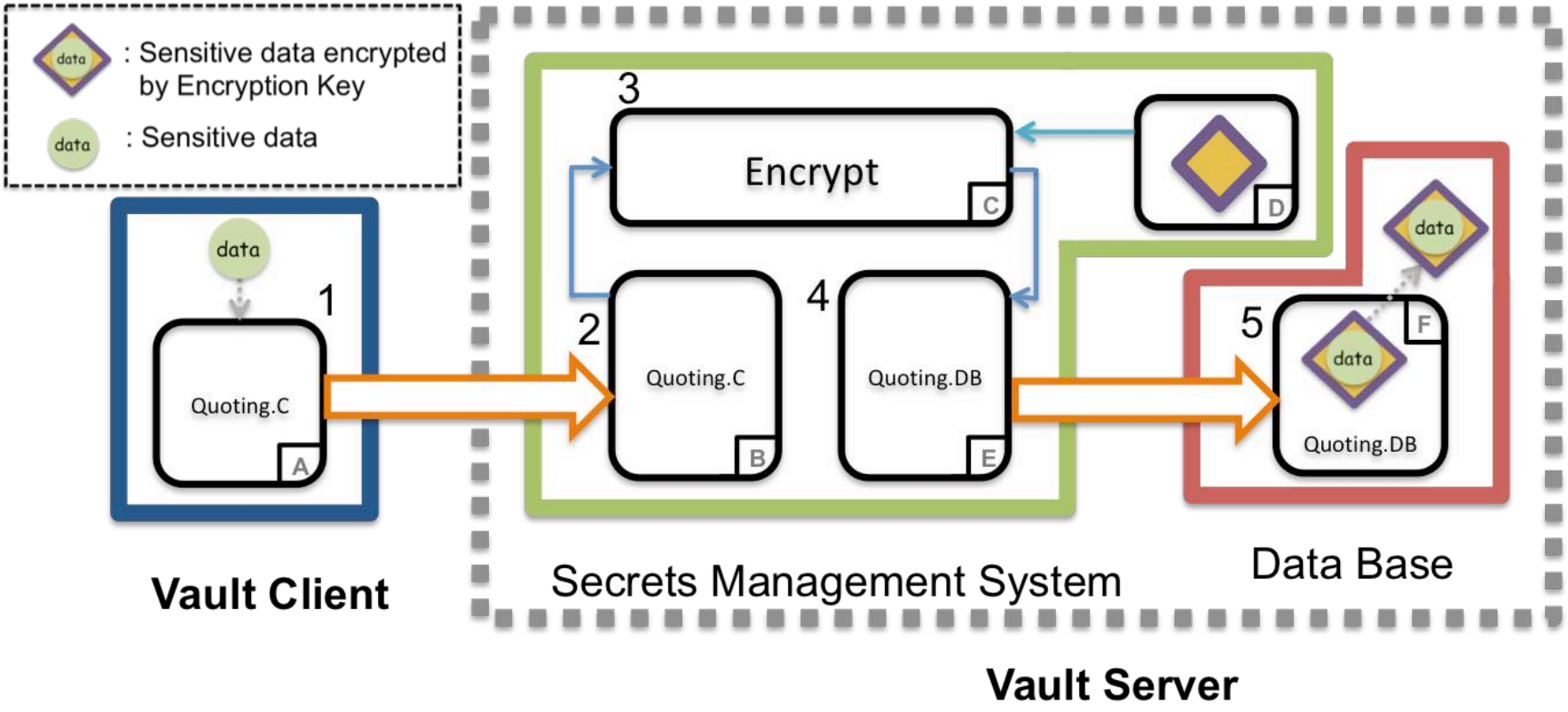

: Local Attestation
: Remote Attestation
: Enclave
: Session Key
: Encryption Key
: Keyring Key encrypted by Master Key
Encrypt
Decrypt
keyring
Quoting.C
Secure Channel
Quoting.C
Quoting.DB
Secure Channel
Quoting.DB
Vault Client
Secrets Management System
Data Base
Vault Server

Unsealing Vault-SGX



- Keyring key is composed of Master key and Encryption Key
- AES_{Master_key} (Keyring key) -> Initializing Vault
- $AES_{Encryption\_key}$(Master Key) -> Initializing Vault

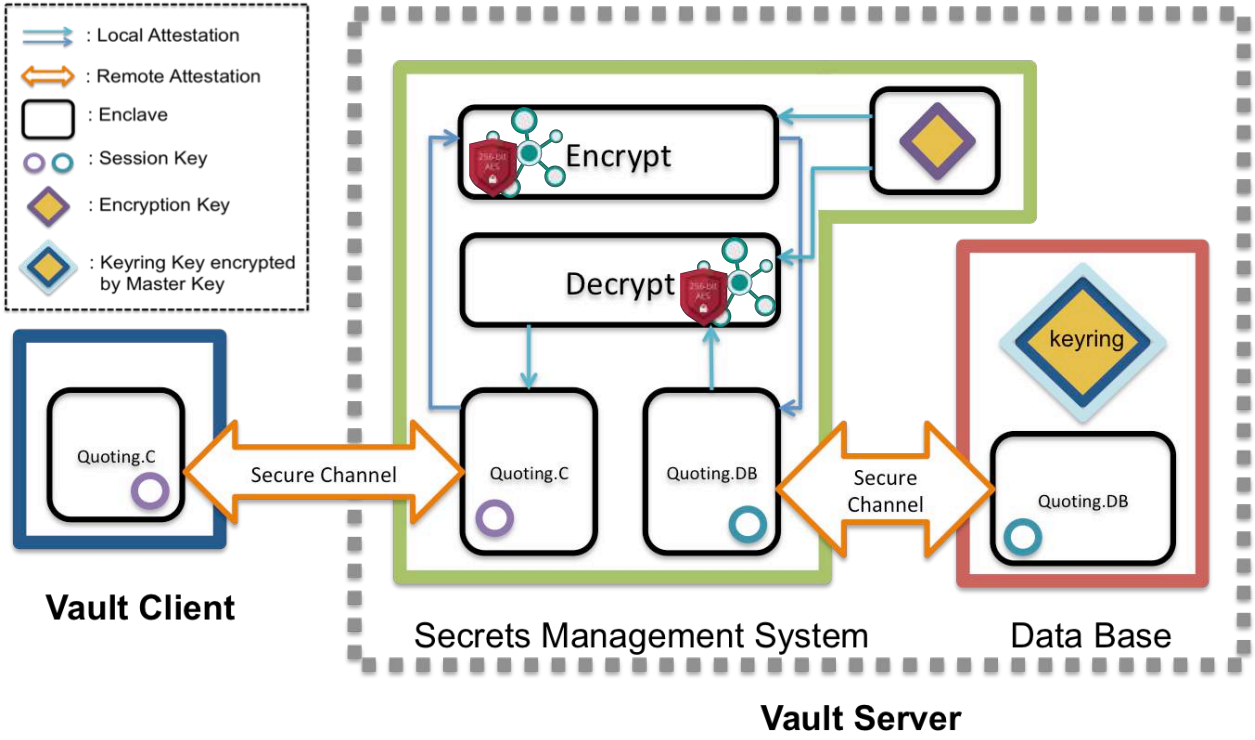
Stored Data in Database



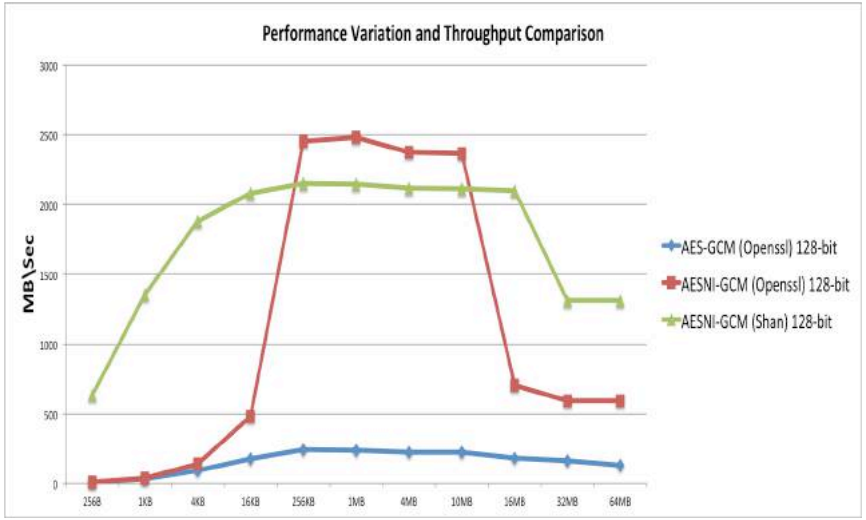
Contribution:

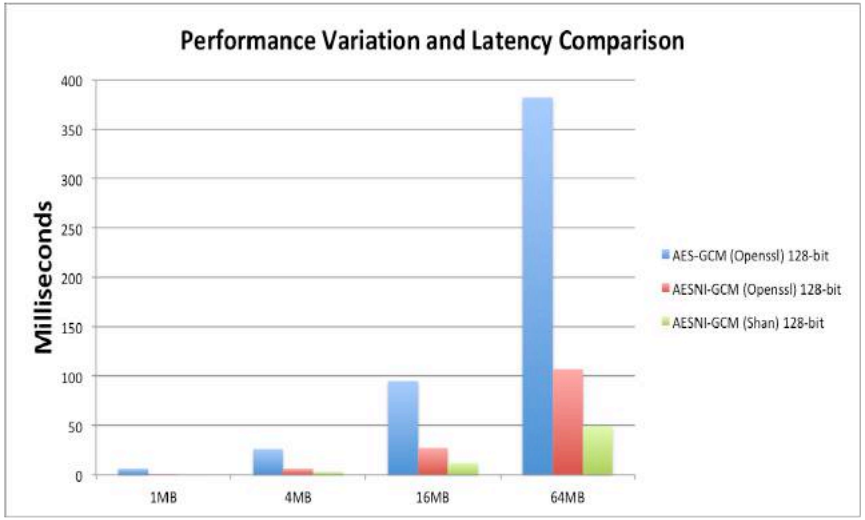
- AESNI-GCM in SGX
- Vault-SGX

Vault-SGX



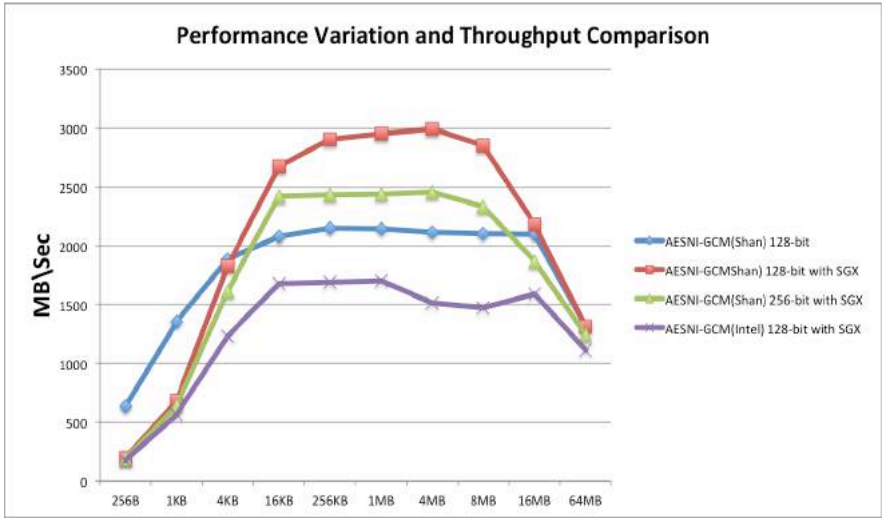
OpenSSL vs. Our Solution [Without Intel SGX]

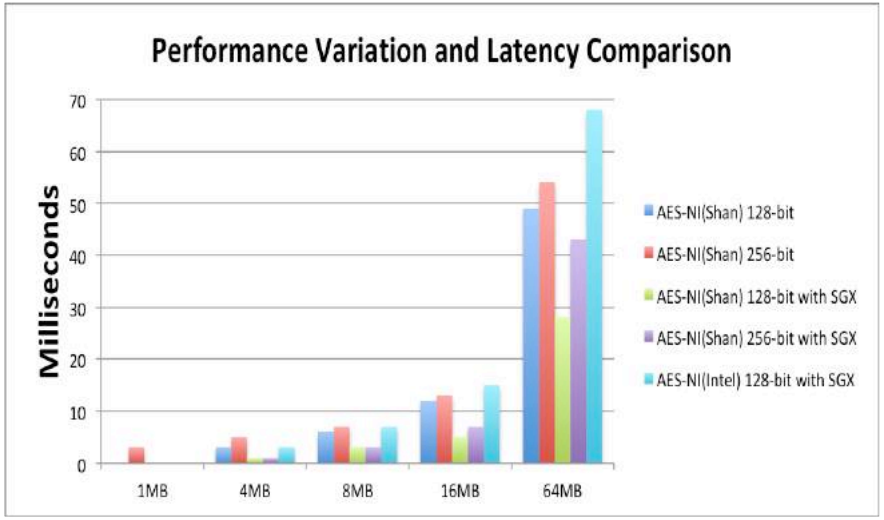




- Why OpenSSL > Our Solution (Throughput)?
- Why OpenSSL < Our Solution (Latency)?
- Why the throughput drops ?
  (Software Input Output Translation Lookaside Buffer)

Intel Solution vs. Our Solution [With Intel SGX]



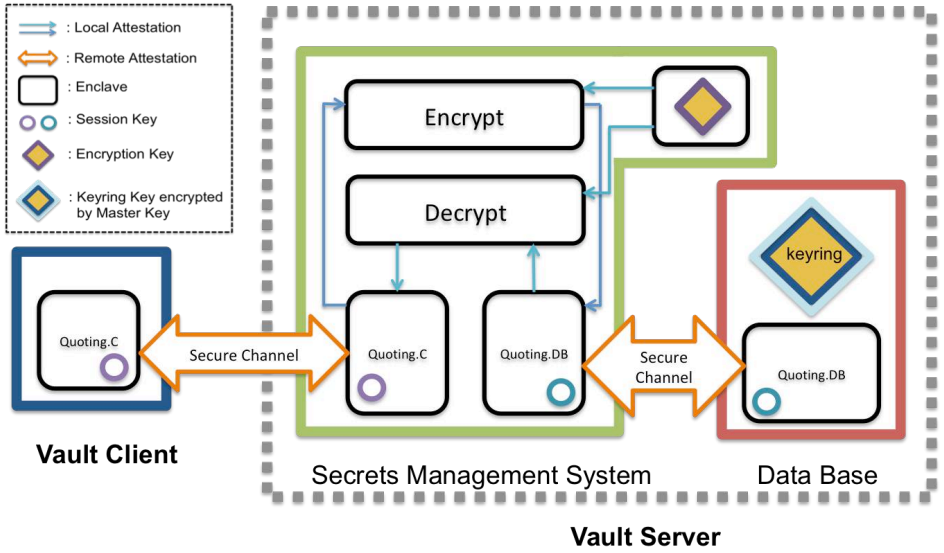


- Why AESNI-GCM with SGX (Red Line) < AESNI-GCM (Blue Line) when execute small data (Throughput)?
- Why AESNI-GCM with SGX (Red Line) > AESNI-GCM (Blue Line) when execute large data (Throughput)?

Vault-SGX vs. Vault

	Vault	Vault SGX
Unsealing	0.073s	0.115s
Read	0.021s	0.040s
Write	0.020s	0.043s

Vault-SGX



Conclusion & Future Work

Ongoing:

- Vault-SGX
  - Remote Attestation
  - Enclave ID Register Service
- Deploy Vault-SGX on Kubernetes Cluster.

Extension:

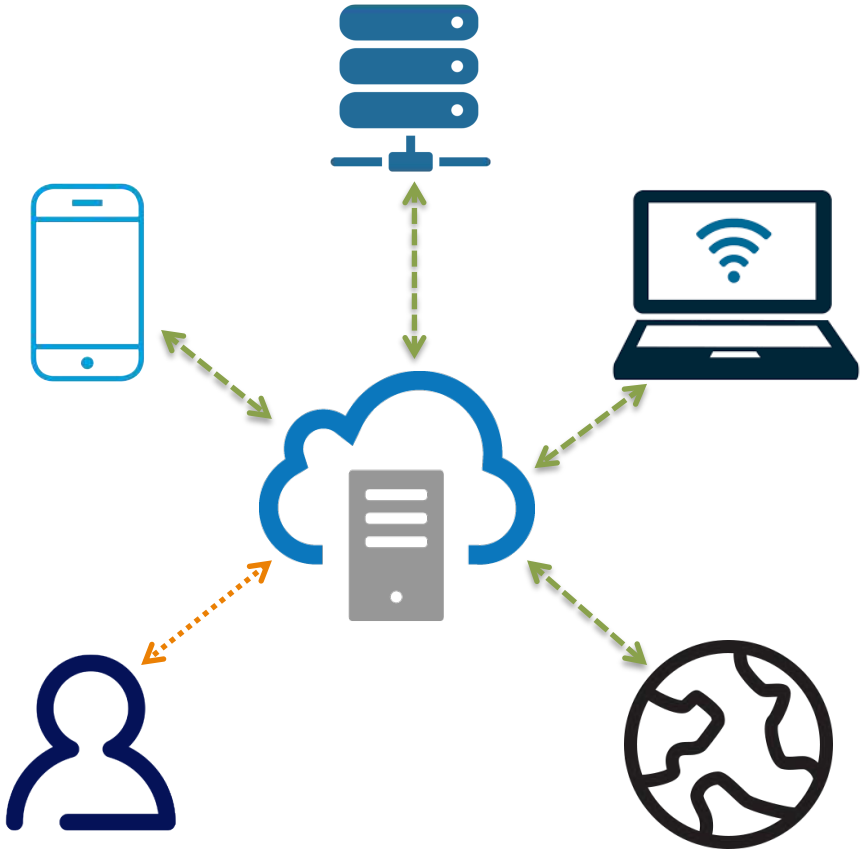
- ARM TrustZone


Q & A

Project Background

Why we use cloud?

- Flexibility
- Efficiency
- Strategic Value

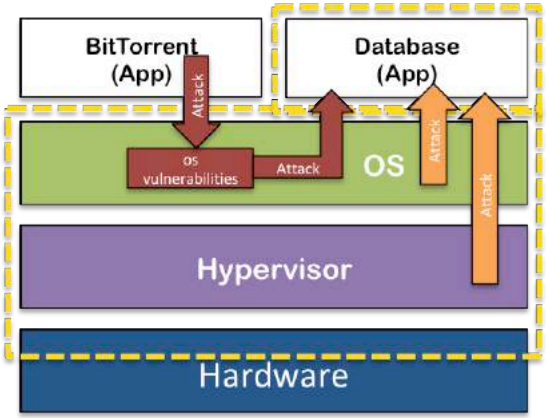


TEE Solutions of the Different CPU vendor:

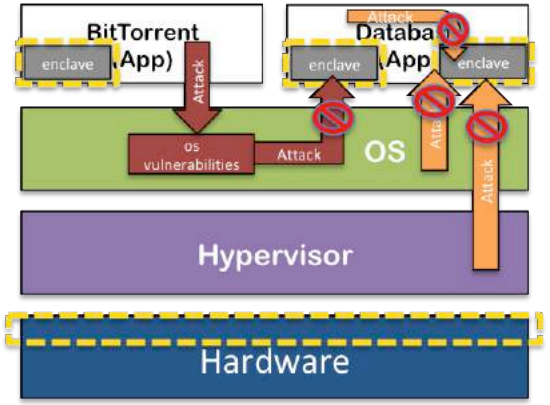
- Intel SGX
- ARM TrustZone
- AMD SEV

TCB of Intel SGX

: TCB



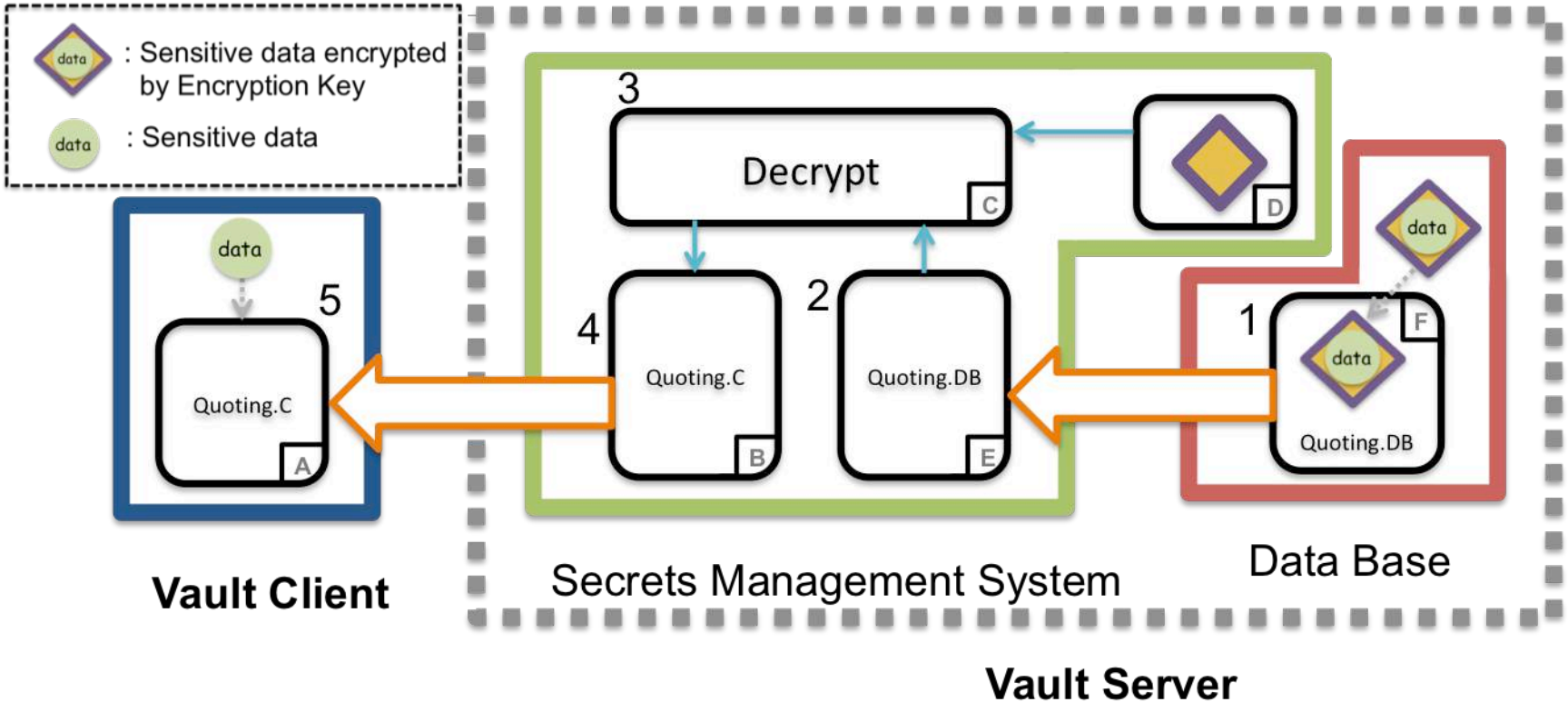
Huge TCB in the ordinary devices



Intel SGX enable device

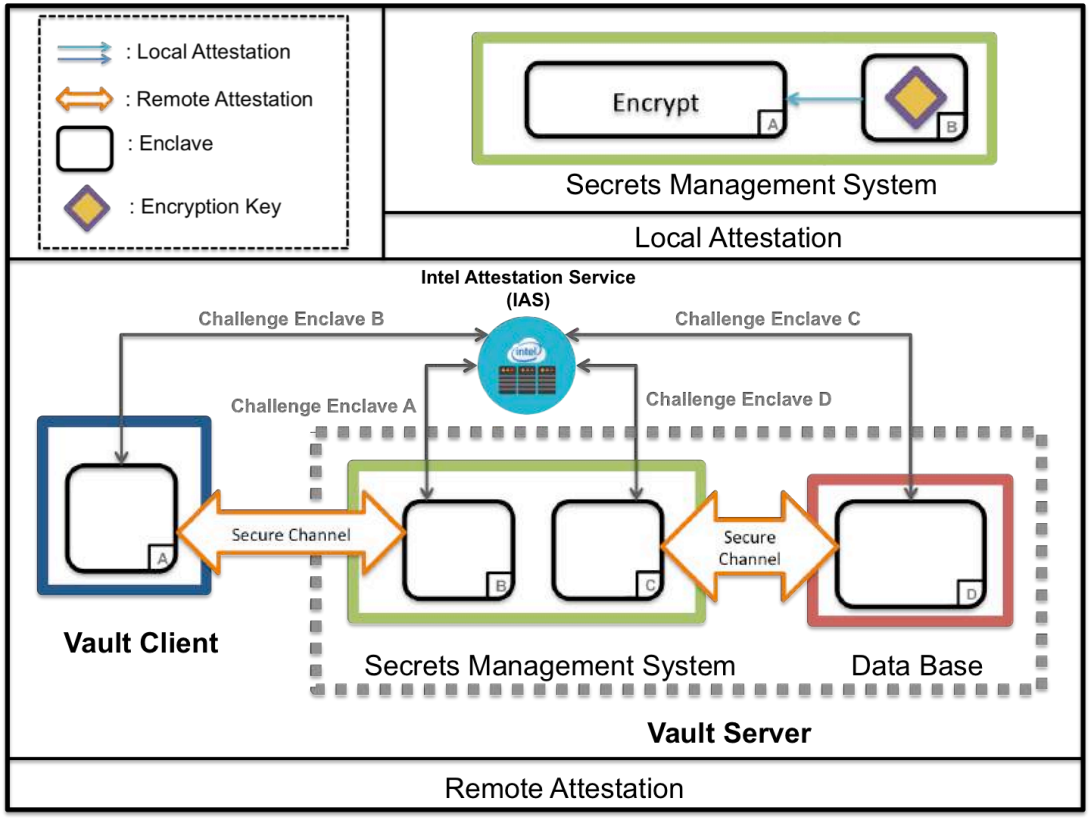
*TCB: Trusted Computing Base

Retrieved Data from Database



Local/Remote Attestation

Attestation Mechanism:
- Local Attestation
- Remote Attestation



Evaluation

Encrypted 4MB Data

- OpenSSL: 24MB
- Our Solution: 8MB

Memory Used

```
root@nuc7i5tee-NUC7i5BNK:~# free
              total        used        free      shared  buff/cache   available
Mem:       16183784      438168    11309352      225344     4436264    15102976
Swap:       3906556           0     3906556
```

The total amount of memory used without data encryption processing.

```
top - 11:03:14 up 14 days, 22:48,  4 users,  load average: 0,64, 0,35, 0,17
Tasks:   1 total,   1 running,   0 sleeping,   0 stopped,   0 zombie
%Cpu(s): 25,0 us,  0,1 sy,  0,0 ni, 74,9 id,  0,1 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 16183784 total, 11301140 free,   446252 used,  4436392 buff/cache
KiB Swap:  3906556 total,  3906556 free,        0 used. 15094764 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 3320 root      20   0   25588  10964   2744 R 100,0  0,1   0:39.44 aesni_shan
```

The total amount of memory used when executing the AESNI-GCM (Shan) with 4MB data size.

```
top - 11:01:52 up 14 days, 22:47,  4 users,  load average: 0,47, 0,28, 0,13
Tasks:   1 total,   1 running,   0 sleeping,   0 stopped,   0 zombie
%Cpu(s): 25,1 us,  0,0 sy,  0,0 ni, 74,9 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 16183784 total, 11283904 free,   463528 used,  4436352 buff/cache
KiB Swap:  3906556 total,  3906556 free,        0 used. 15077524 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 3299 root      20   0   44312  28544   3632 R 100,0  0,2   0:31.75 aesni_openssl
```

The total amount of memory used when executing the AESNI-GCM (OpenSSL) with 4MB data size.

Evaluation

Encrypted 10 MB Data:
OpenSSL: 64MB

Memory Used