CLOUDNATIVE SECURITYCON

NORTH AMERICA 2023

CLOUDNATIVE
SECURITYCON
NORTH AMERICA 2023

Security++: Hide your secrets via a distributed Hardware Security Module (HSM)

Iris Ding
Cloud Software Engineer, shaojun.ding@intel.com

Malini Bhandaru
Senior Principal Engineer, malini.bhandaru@intel.com

Thanks to my colleagues: QimingLiu, HuailongZhang, XintongChen, XinHuang, RuijingGuo, RuoyuYing, ChangranWang, ForrestZhao, SoodKapil, PoussaSakari, PuustinenIsmo,ValluriAmarnath, Venkatasubramanian Sankaranarayanan

Agenda

- Cloud HSM and Challenges
- Distributed HSM
- Use Cases

Hardware Security Module (HSM)

A physical computing device that safeguards and manages secrets (most importantly digital keys), performs encryption and decryption functions for digital signatures, strong authentication and other cryptographic functions. Traditionally a plug-in card or an external device that attaches directly to a computer or network server. A hardware security module contains one or more secure cryptoprocessor chips.

https://en.wikipedia.org/wiki/Hardware_security_module

HSM Market

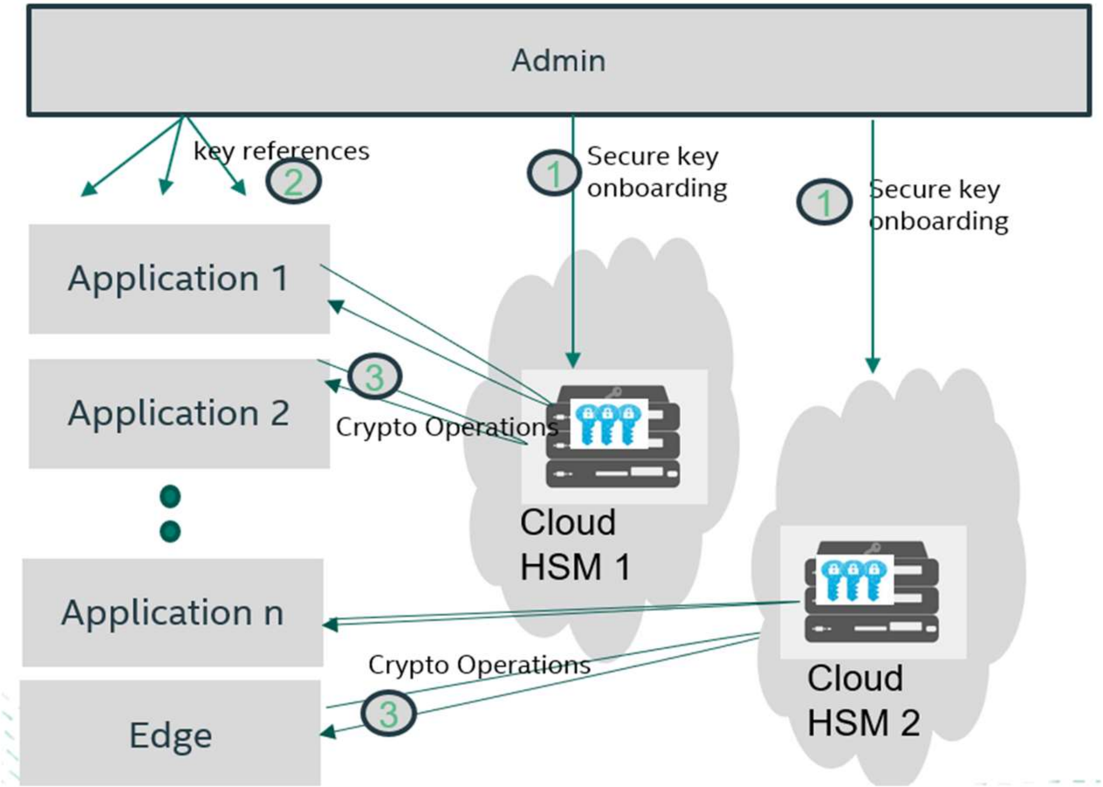
Expected to reach USD **2.0 Billion by 2028,** growing at a **CAGR of 13.1%**

Driven by:

- Growing data breaches and cyberattacks
- Increasing demand for data security in cloud environments

* Data source: https://www.marketsandmarkets.com/Market-Reports/hardware-security-modules-market-162277475.html

Cloud HSM



Pros

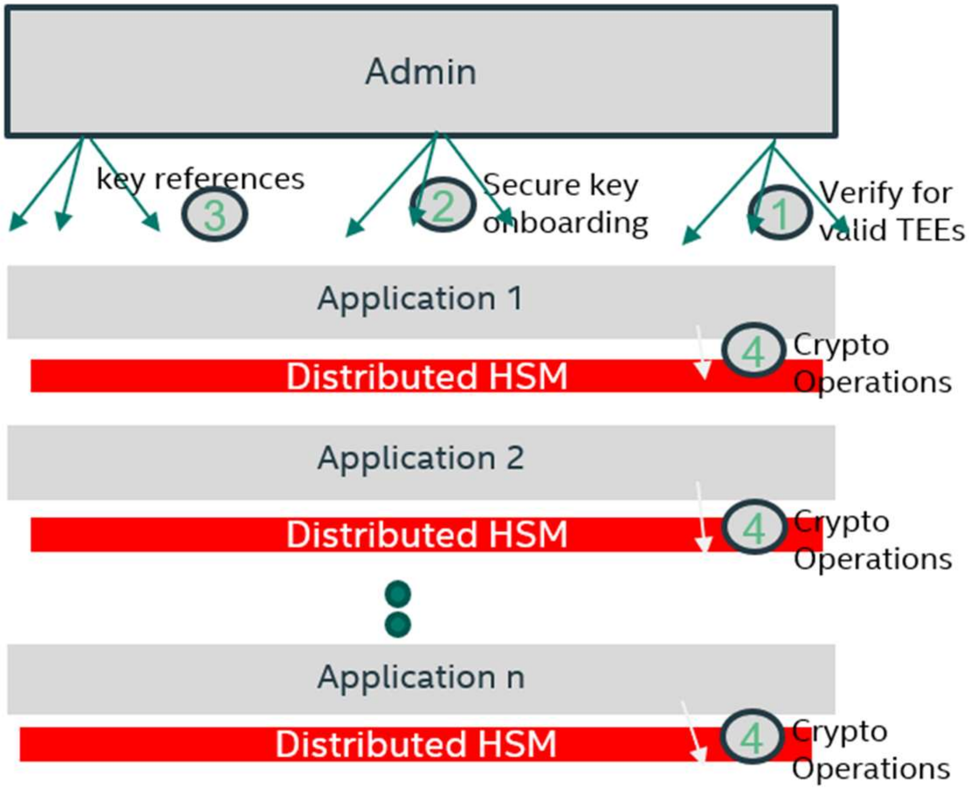
- Lower cost from sharing
- Flexibility and simplicity

Cons:

- Higher latency crypto operations
- Lower transaction rate (TPS)
- Migration difficulty
- No substitutes on edge

Distributed HSM

Where you need it, sized to your needs



- Highly Secure, even at the Edge
- Lower Latency and Greater Throughput
- Lower Cost

How?

Using
Trusted Execution Environments!

Trusted Execution Environments (TEEs)

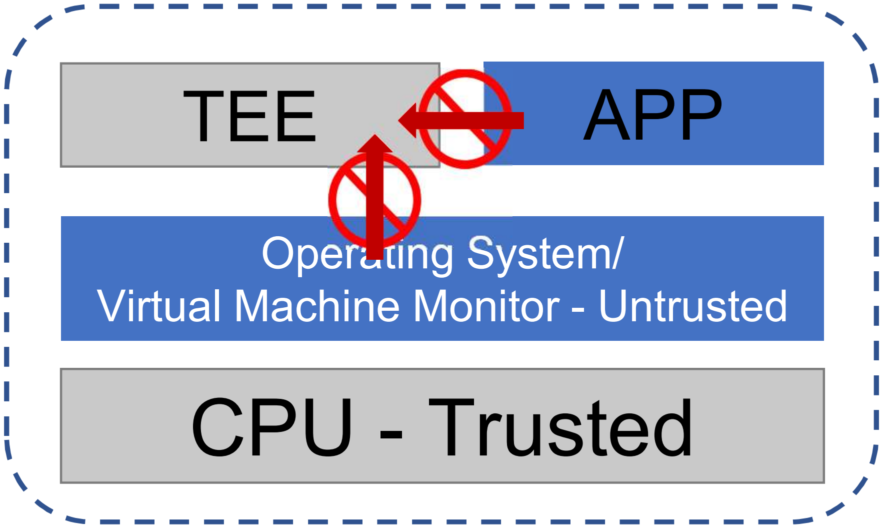
SECURE

Data at Rest

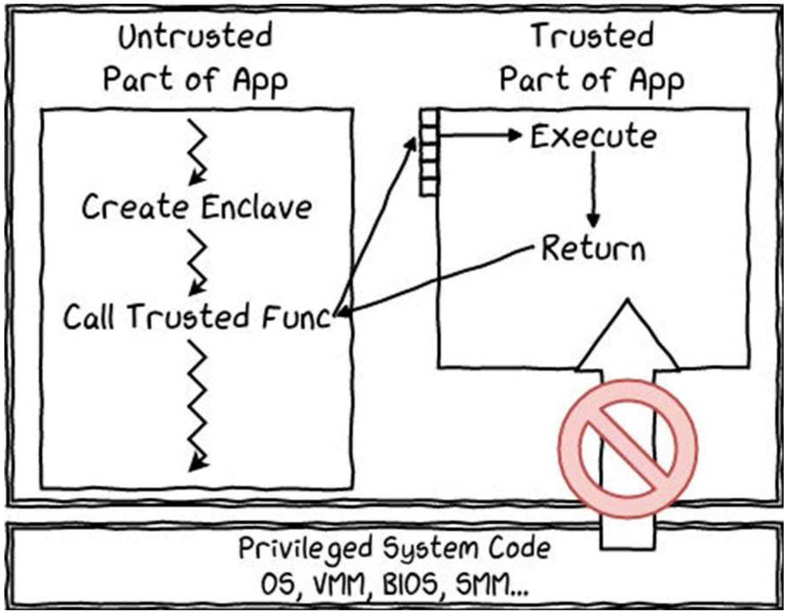
Data in Motion

Data in Use

- Hardware and firmware supported confidentiality and integrity of code and data
- Protect even from privileged processes (OS, Hypervisor..)
- Demonstrate trust - quotes and attestation



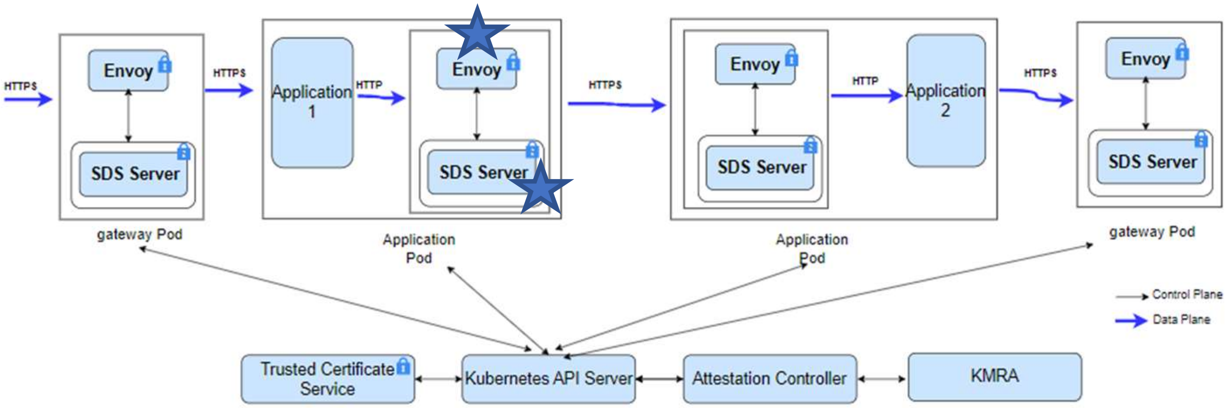
Intel SGX: a Process-based TEE



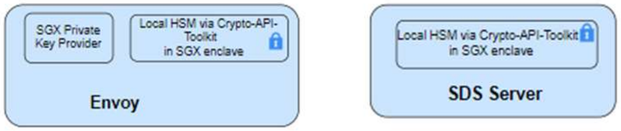
- ❑ **Memory Encryption**
- ❑ **Remote Attestation**
- ❑ **Access Control**
- ❑ **Sealing**

https://www.intel.com/content/www/us/en/support/articles/000058764/software/intel-security-products.html

Use Case 1 – Istio Service Mesh(mTLS & Gateways)



Architecture of Service Mesh data plane private key protection


SGX Private Key Provider
Local HSM via Crypto-API-Toolkit in SGX enclave
Envoy
Local HSM via Crypto-API-Toolkit in SGX enclave
SDS Server

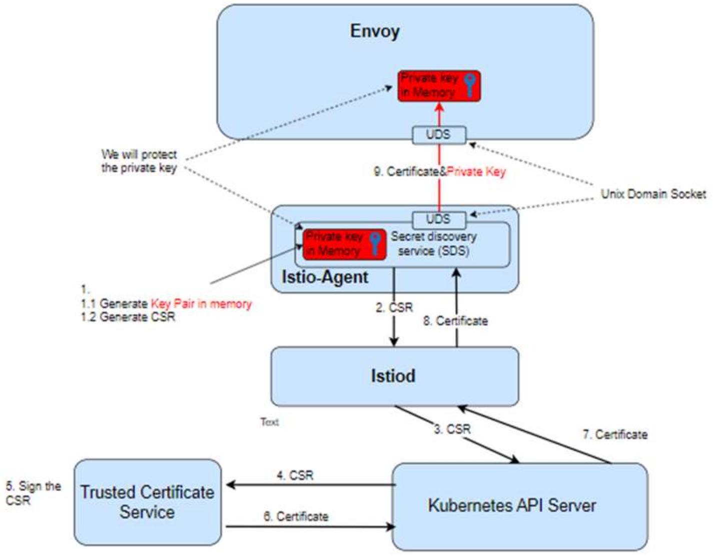
- Local HSM via SGX enclave
- Local Crypto operations
- Credentials can be synced from remote HSM or locally generated

https://github.com/intel/istio
https://github.com/intel/envoy/
https://github.com/istio-ecosystem/hsm-sds-server

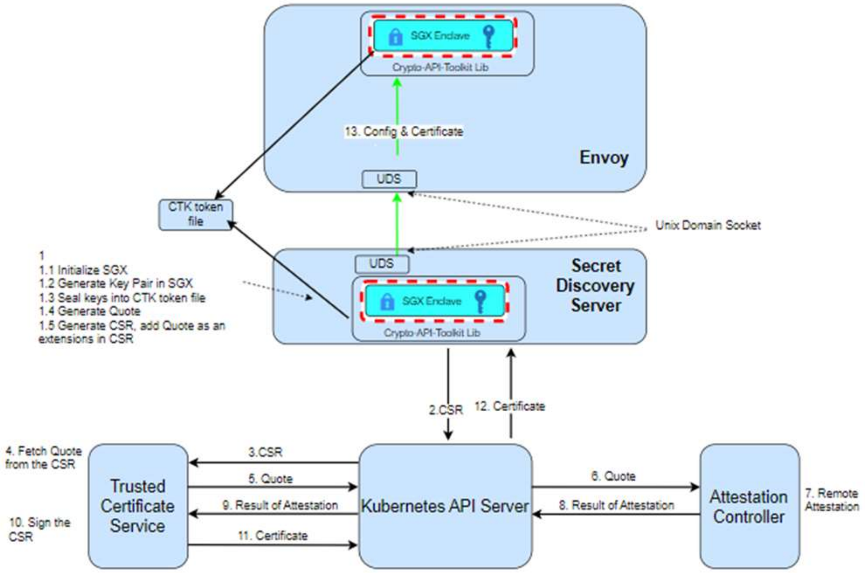
Use Case 1 - Istio Service Mesh(mTLS control plane)

BEFORE

AFTER



Control Plane for mTLS keys - Original flow from upstream(integrated with Intel Trusted Certificate Service)



Control Plane for mTLS for Envoy SGX Enhancement

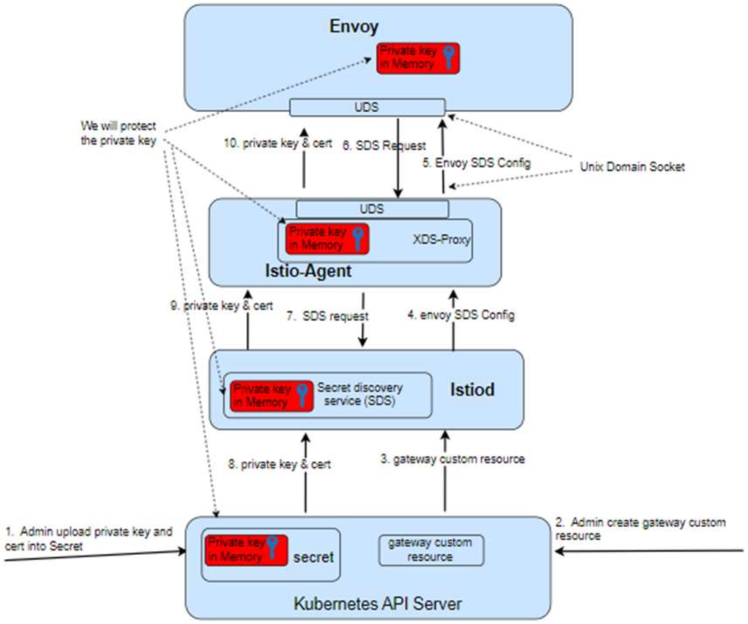
- Leverage external CA
- Private keys are in clear text
- Private keys are generated locally

- Leverage external certificate authority
- Private keys never exposed in clear text
- Signed cert issued on enclave verification
- Crypto Operations locally

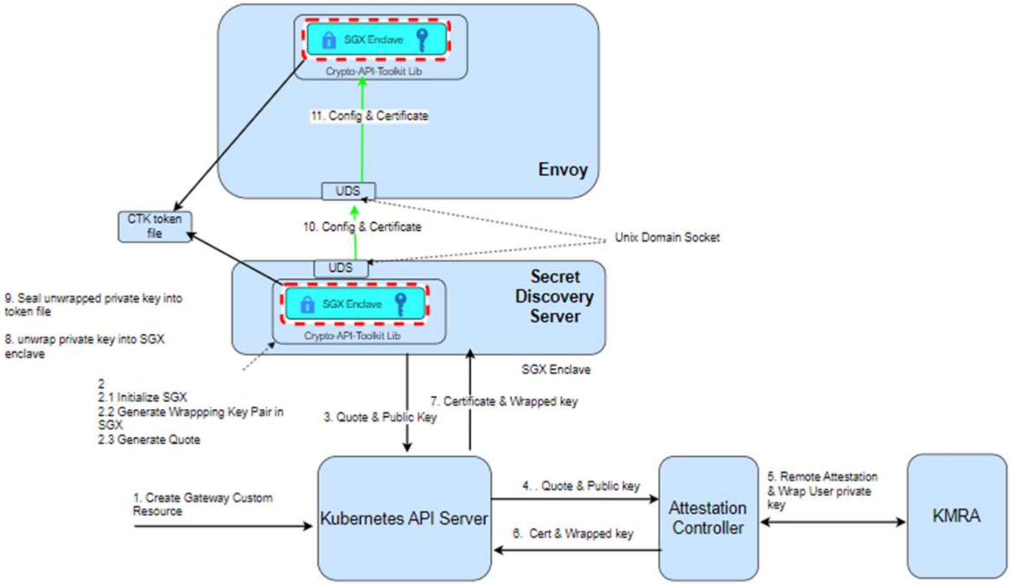
Use Case 1 - Istio Service Mesh(gateway control plane)

BEFORE

AFTER



Control Plane for gateway private keys - Original flow from upstream



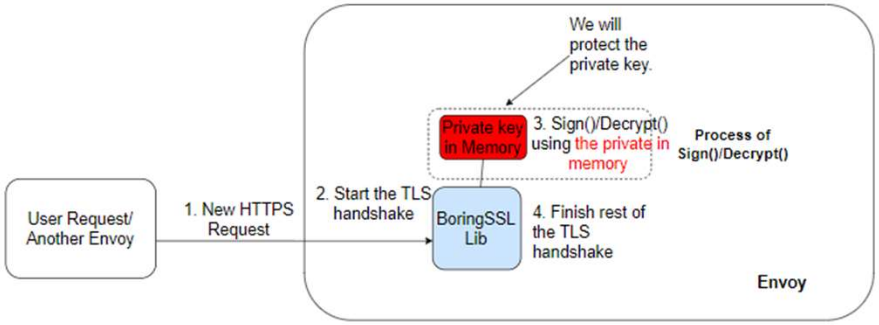
Control Plane for Gateway SGX Enhancement

- Private keys are in clear text
- Private keys are uploaded externally

- Private keys never exposed in clear text
- Got keys uploaded only if enclave attestation verified
- Crypto Operation happened locally

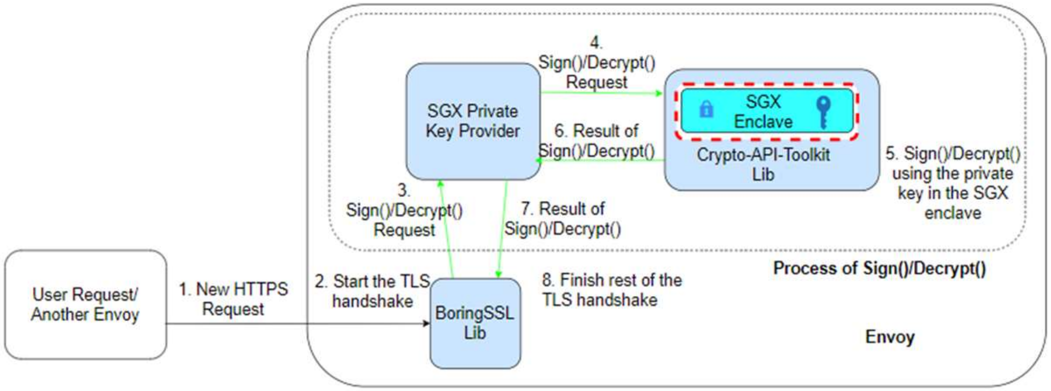
Use Case 1 - Istio Service Mesh(data plane)

BEFORE → AFTER


We will protect the private key.
Private key in Memory
3. Sign()/Decrypt() using the private in memory
Process of Sign()/Decrypt()
User Request/ Another Envoy
1. New HTTPS Request
2. Start the TLS handshake
BoringSSL Lib
4. Finish rest of the TLS handshake
Envoy

Data Plane - Original flow from upstream

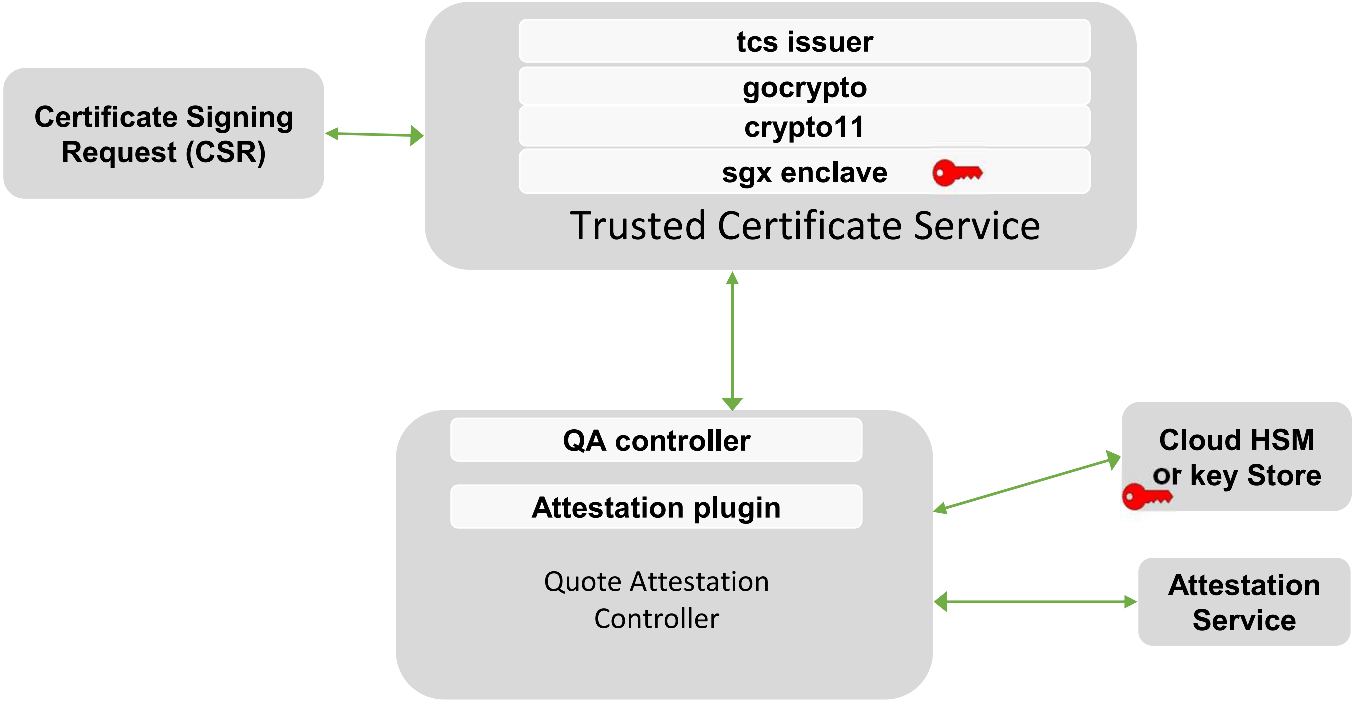
- Crypto operations using the private keys in memory


4. Sign()/Decrypt() Request
SGX Private Key Provider
SGX Enclave
6. Result of Sign()/Decrypt()
Crypto-API-Toolkit Lib
5. Sign()/Decrypt() using the private key in the SGX enclave
3. Sign()/Decrypt() Request
7. Result of Sign()/Decrypt()
Process of Sign()/Decrypt()
User Request/ Another Envoy
1. New HTTPS Request
2. Start the TLS handshake
BoringSSL Lib
8. Finish rest of the TLS handshake
Envoy

Data Plane for Envoy SGX Enhancement

- Private keys never exposed in clear text
- Got keys uploaded only if enclave attestation verified
- Crypto Operation happened locally

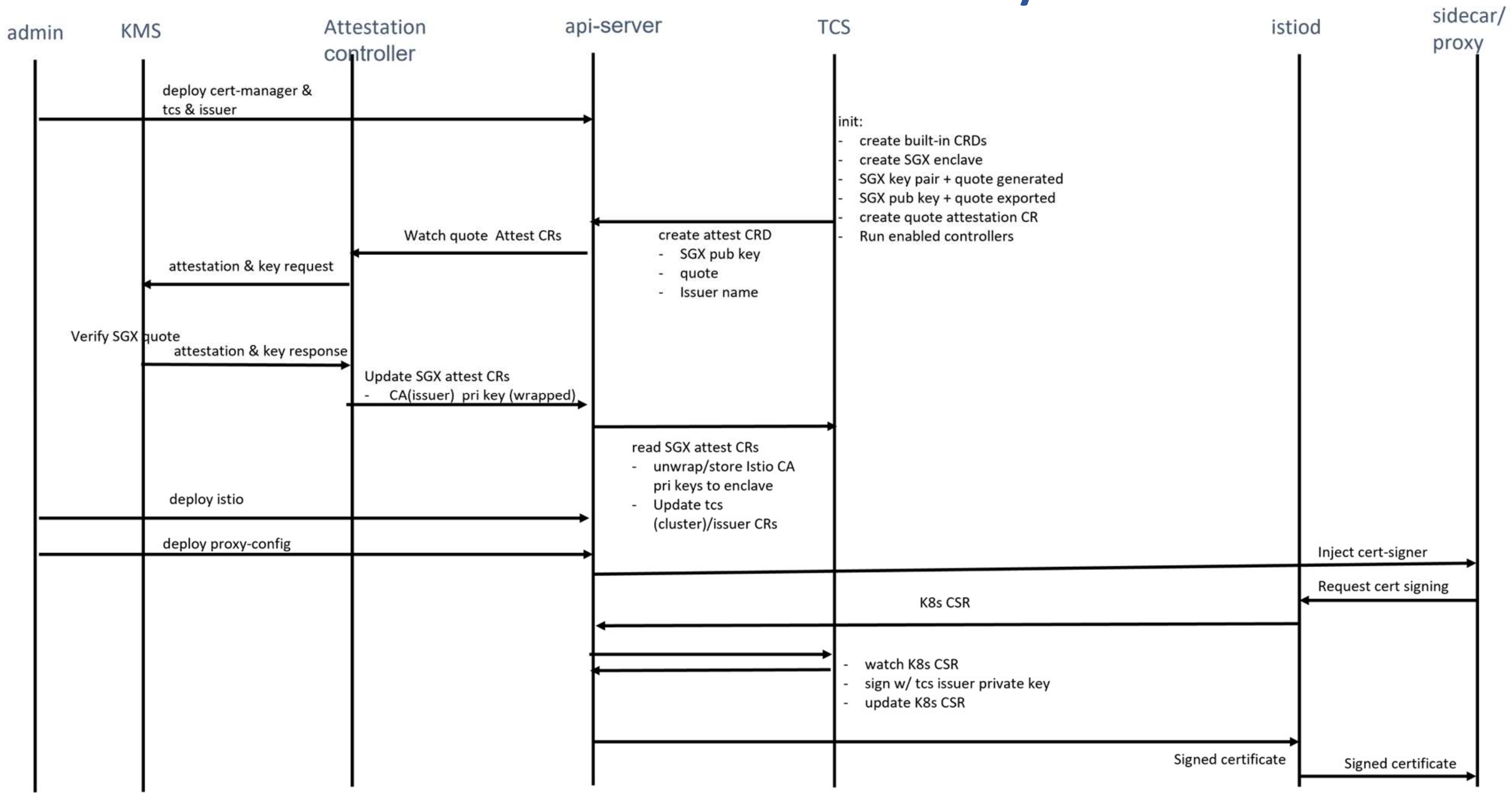
Use Case 2 – Certificate Authority (CA)



- CA Credentials can be synced from remote HSM or locally generated
- Crypto operations happen in local SGX enclave
- Credentials synced only if enclave attestation verified

https://github.com/intel/trusted-certificate-issuer

Use Case 2 - Certificate Authority Flow



admin

KMS

Attestation controller

api-server

TCS

istiod

sidecar/ proxy

deploy cert-manager & tcs & issuer

init:
- create built-in CRDs
- create SGX enclave
- SGX key pair + quote generated
- SGX pub key + quote exported
- create quote attestation CR
- Run enabled controllers

create attest CRD
- SGX pub key
- quote
- Issuer name

Watch quote Attest CRs

attestation & key request

Verify SGX quote

attestation & key response

Update SGX attest CRs
- CA(issuer) pri key (wrapped)

read SGX attest CRs
- unwrap/store Istio CA pri keys to enclave
- Update tcs (cluster)/issuer CRs

deploy istio

deploy proxy-config

Inject cert-signer

Request cert signing

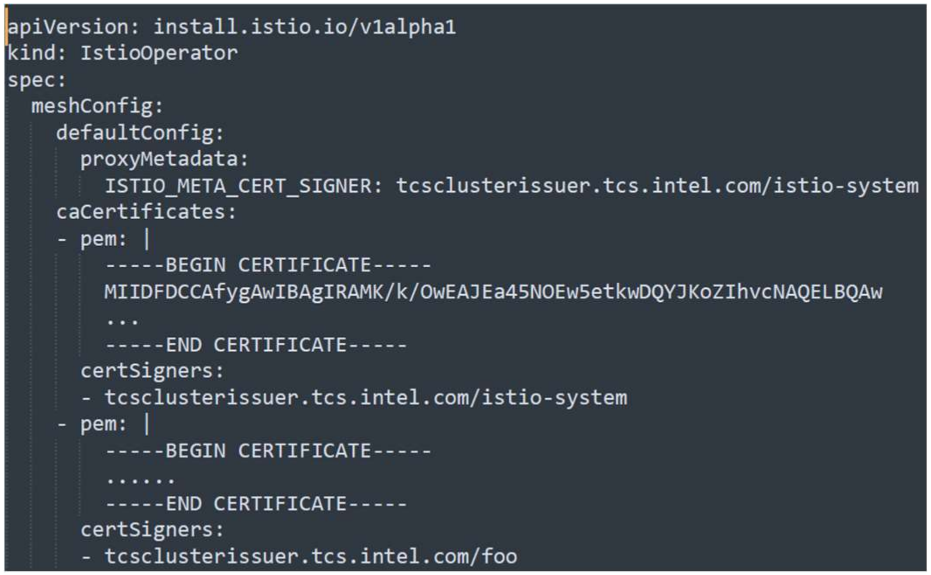
K8s CSR

- watch K8s CSR
- sign w/ tcs issuer private key
- update K8s CSR

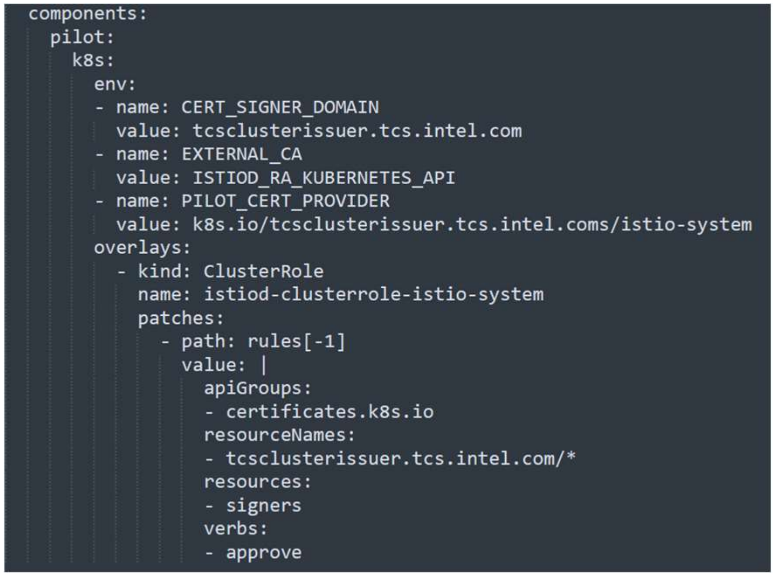
Signed certificate

Signed certificate

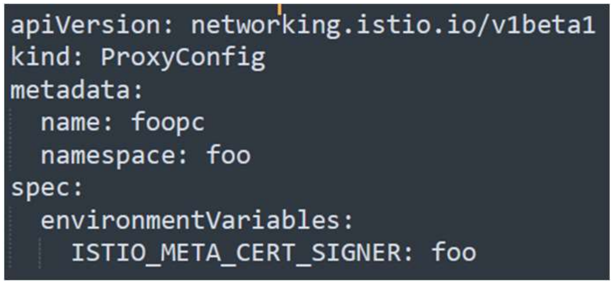
Use Case 2 - Certificate Authority Sample Usage



```yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    defaultConfig:
      proxyMetadata:
        ISTIO_META_CERT_SIGNER: tcsclusterissuer.tcs.intel.com/istio-system
    caCertificates:
    - pem: |
        -----BEGIN CERTIFICATE-----
        MIIDFDCCAfygAwIBAgIRAMK/k/OwEAJEa45NOEw5etkwDQYJKoZIhvcNAQELBQAw
        ...
        -----END CERTIFICATE-----
      certSigners:
      - tcsclusterissuer.tcs.intel.com/istio-system
    - pem: |
        -----BEGIN CERTIFICATE-----
        ......
        -----END CERTIFICATE-----
      certSigners:
      - tcsclusterissuer.tcs.intel.com/foo
```

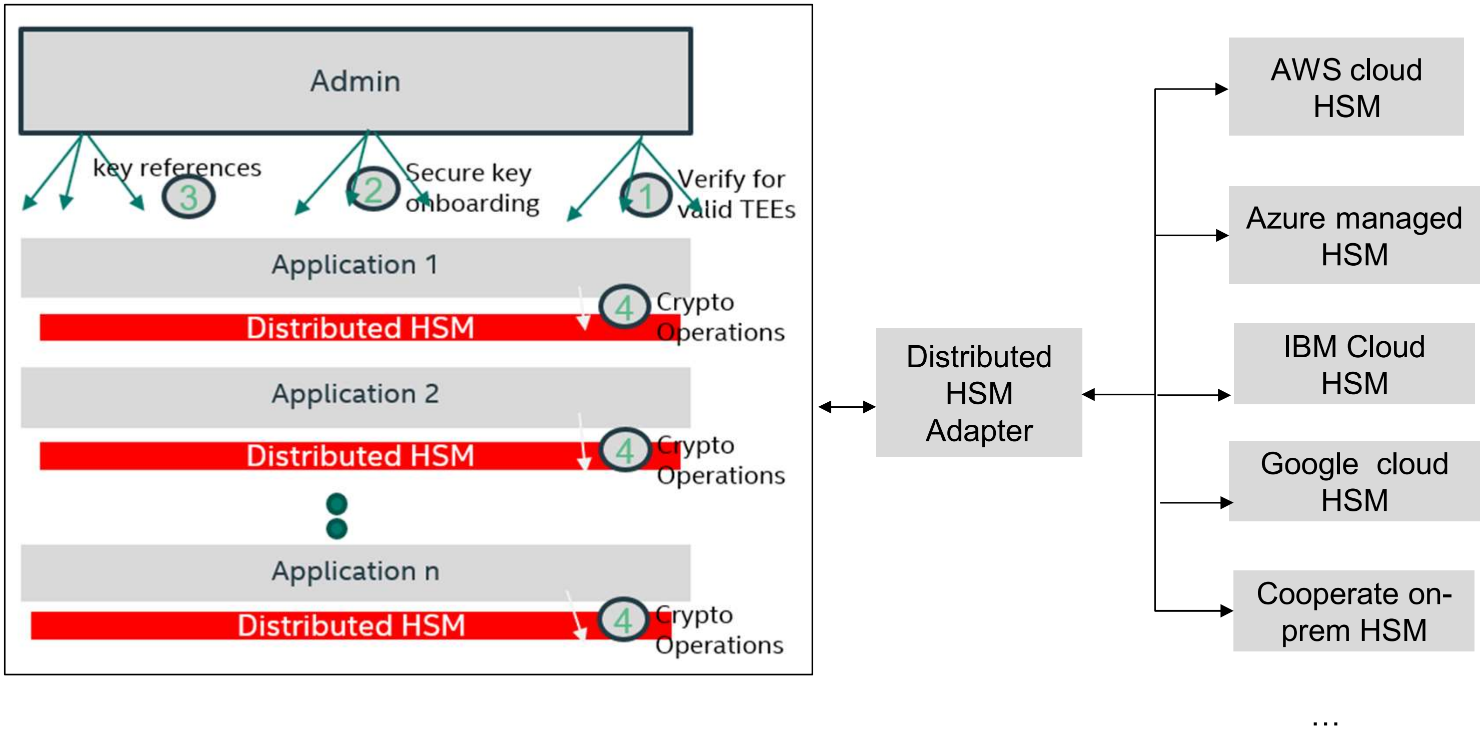


```yaml
  components:
    pilot:
      k8s:
        env:
        - name: CERT_SIGNER_DOMAIN
          value: tcsclusterissuer.tcs.intel.com
        - name: EXTERNAL_CA
          value: ISTIOD_RA_KUBERNETES_API
        - name: PILOT_CERT_PROVIDER
          value: k8s.io/tcsclusterissuer.tcs.intel.coms/istio-system
        overlays:
          - kind: ClusterRole
            name: istiod-clusterrole-istio-system
            patches:
              - path: rules[-1]
                value: |
                  apiGroups:
                  - certificates.k8s.io
                  resourceNames:
                  - tcsclusterissuer.tcs.intel.com/*
                  resources:
                  - signers
                  verbs:
                  - approve
```



```yaml
apiVersion: networking.istio.io/v1beta1
kind: ProxyConfig
metadata:
  name: foopc
  namespace: foo
spec:
  environmentVariables:
    ISTIO_META_CERT_SIGNER: foo
```

Future Steps



- Unified API
- Crypto operation happened in Local distributed HSM

Resources

- https://github.com/intel/istio
- https://github.com/intel/envoy/
- https://github.com/istio-ecosystem/hsm-sds-server
- https://github.com/intel/trusted-certificate-issuer
- https://www.intel.com/content/www/us/en/developer/topic-technology/open/key-management-reference-application/overview.html
- https://github.com/intel/trusted-attestation-controller
- https://github.com/intel/ehsm
- https://istio.io/latest/docs/tasks/security/cert-management/custom-ca-k8s/

…

Explore & Join Us!

Thank you

**Please scan the QR Code above
to leave feedback on this session**