End-to-End Encryption with the Split-Trust Encryption Tool

Jessie Liu

Google Cloud

Overview

- Split Trust Encryption Tool (STET)
  - Open-source CLI tool for encrypting/decrypting data
  - Secure key ingress and egress in/out of GCP
- **Objective:** Ensure the only entities with access to data are the data originator and data consumer

Overview

On-Prem Machine

- Data originator
- Trusted due to being on-premises

KMS

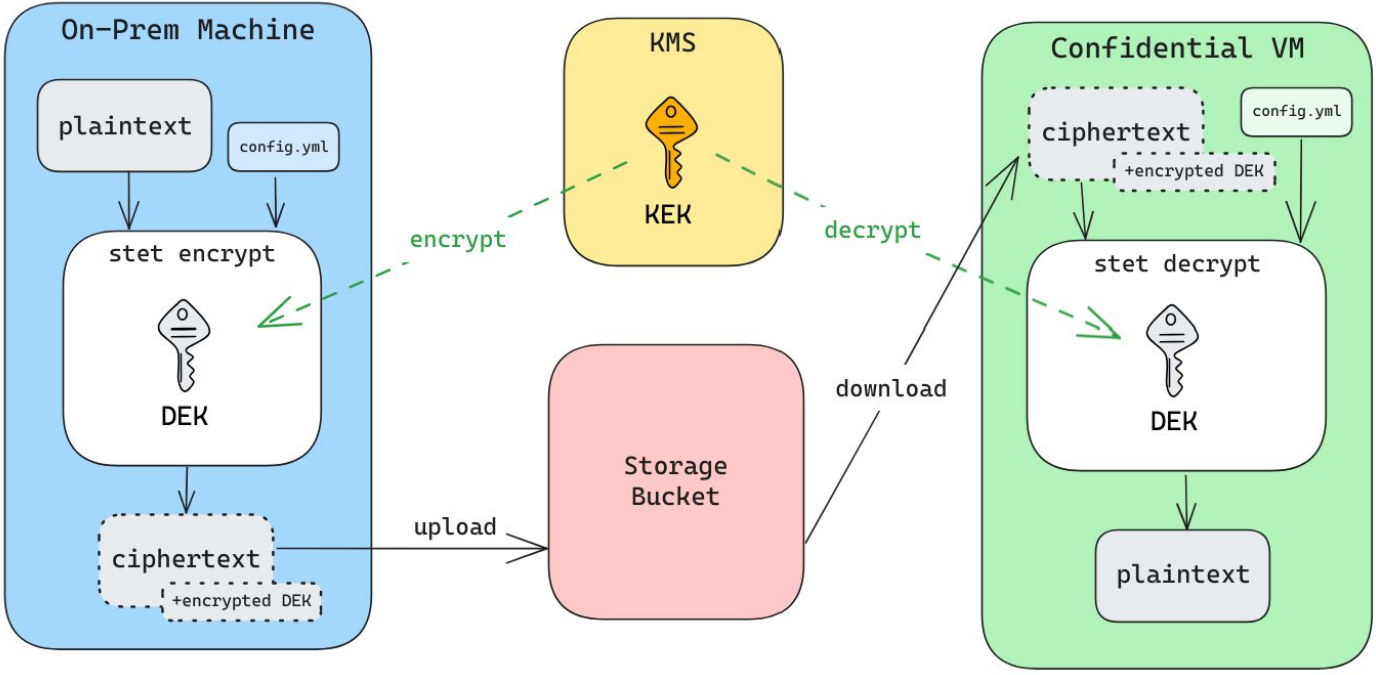
- Holds KEKs
- Required to trust, unless using split trust

Storage Bucket

- Intermediate storage for encrypted data
- Not required to trust due to data encryption

Confidential VM

- Data consumer
- Trusted due to Confidential Computing

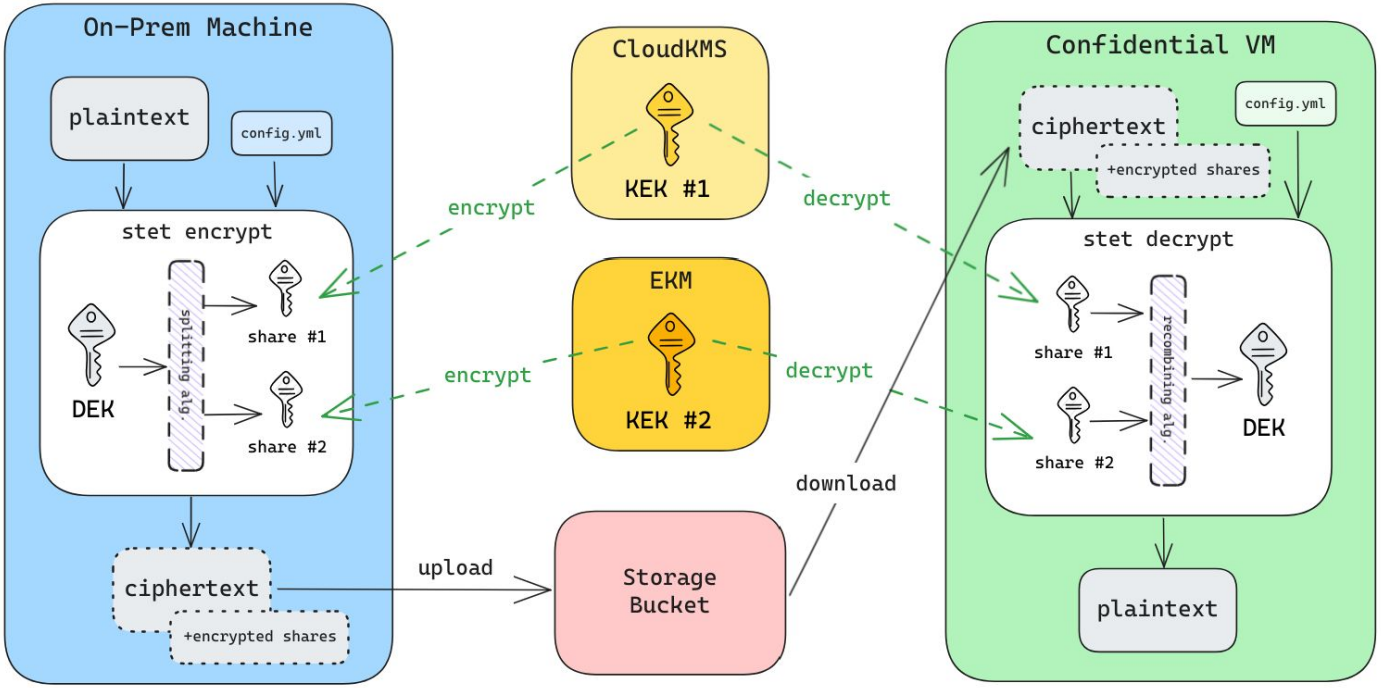

On-Prem Machine
plaintext
config.yml
stet encrypt
DEK
ciphertext
+encrypted DEK
KMS
KEK
encrypt
decrypt
Storage Bucket
upload
download
Confidential VM
ciphertext
+encrypted DEK
config.yml
stet decrypt
DEK
plaintext

Goals

- **Reduce amount of trust needed**
  - Split Trust - using multiple key management systems removes the need to trust a single KMS
- **Fully establish trust when trust is needed**
  - Support attestation in access policies for KEKs
  - Multiple varieties of attestation:
    - Platform-based (TPM attestation)
    - Workload-based (Confidential Space)

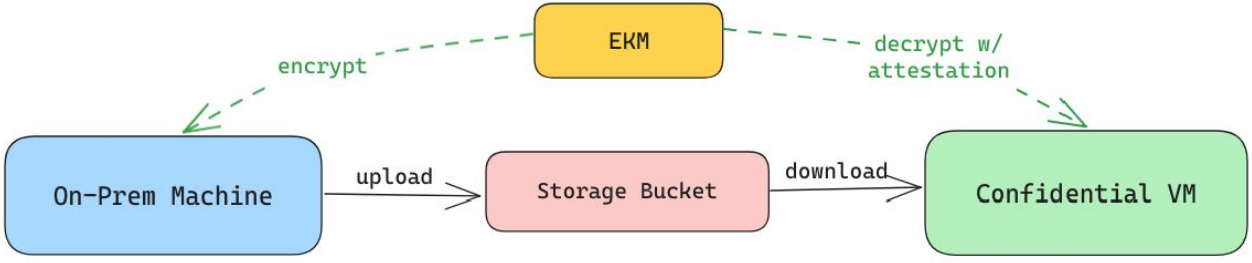
Split Trust

- Purpose: Remove the need to unilaterally trust a KMS
- **During encryption:** split DEK into *n* shares, with each share is encrypted by a different KMS
  - Specify *k*<=*n* shares needed to reconstruct the DEK
    - Having <*k* shares reveals no information about the secret
  - Maliciously accessing data requires collusion between the KMSes
- **During decryption:** each share is decrypted, then the DEK is recombined if at least *k* shares are available


On-Prem Machine
plaintext
config.yml
stet encrypt
DEK
splitting alg.
share #1
share #2
ciphertext
+encrypted shares
CloudKMS
KEK #1
EKM
KEK #2
encrypt
encrypt
upload
Storage Bucket
decrypt
decrypt
download
Confidential VM
ciphertext
+encrypted shares
config.yml
stet decrypt
share #1
share #2
recombining alg.
DEK
plaintext

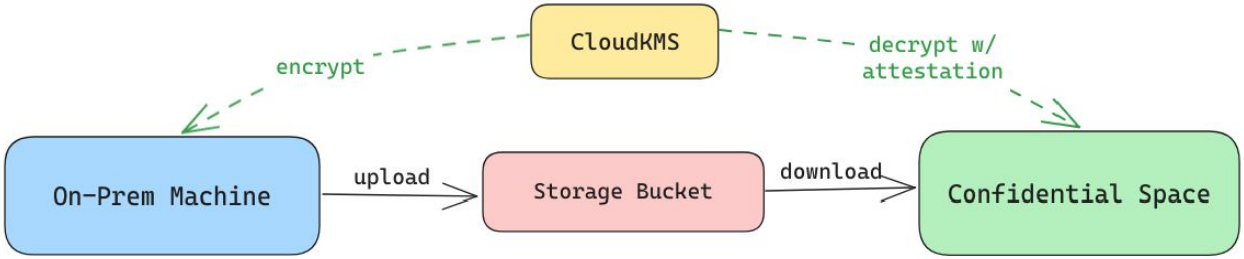
Attestation – EKMs

- TPM attestation is natively supported
  - Configured in KEK settings
  - When STET communicates with EKM, EKM requests an attestation if the key policy requires it

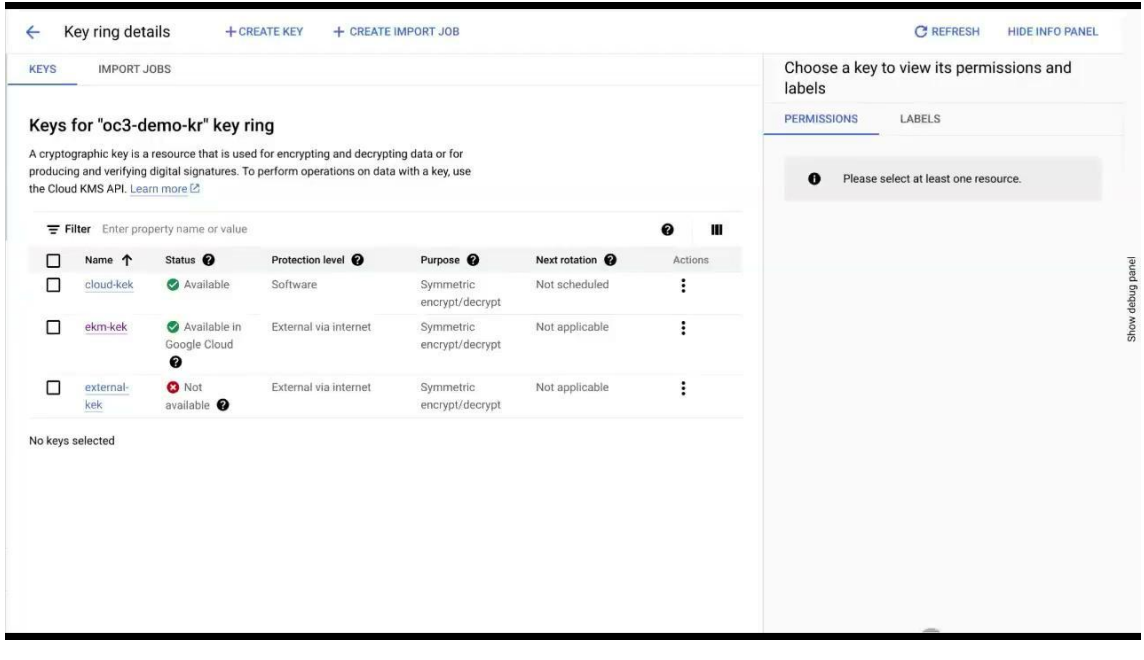


Attestation – CloudKMS

- Supported with Confidential Space attestation token
  - Confidential Space allows confidential data to be shared with a workload while retaining confidentiality & ownership
  - STET sends attestation token to CloudKMS, Cloud IAM verifies access policy



Demo


Key ring details
+ CREATE KEY
+ CREATE IMPORT JOB
REFRESH
HIDE INFO PANEL
KEYS
IMPORT JOBS
Keys for "oc3-demo-kr" key ring
A cryptographic key is a resource that is used for encrypting and decrypting data or for producing and verifying digital signatures. To perform operations on data with a key, use the Cloud KMS API. Learn more
Filter
Enter property name or value
Name
Status
Protection level
Purpose
Next rotation
Actions
cloud-kek
Available
Software
Symmetric encrypt/decrypt
Not scheduled
ekm-kek
Available in Google Cloud
External via internet
Symmetric encrypt/decrypt
Not applicable
external-kek
Not available
External via internet
Symmetric encrypt/decrypt
Not applicable
No keys selected
Choose a key to view its permissions and labels
PERMISSIONS
LABELS
Please select at least one resource.
Show debug panel

To conclude…

- STET provides a secure way of sending data in/out of GCP that is protected from insiders
  - Usability - manages KMS communication & key split

```
$ gsutil cp --stet secrets.txt "gs://my-bucket/my-secrets"
$ gsutil cp --stet "gs://my-bucket/my-secrets" plaintext.txt
```

- Confidential Space supports workload-based attestation
  - STET provides usability & convenience in CS

Next Steps

- More info & try our quickstart guide:
  - https://github.com/GoogleCloudPlatform/stet

- Details on Confidential Space
  - *'Securely collaborating across multiple cloud providers'* talk by Josh Krstic
  - https://cloud.google.com/docs/security/confidential-space