What is Google Cloud HSM? How to Protect Data in Google Cloud?

Google Cloud HSM (Hardware Security Module) is a cloud service offered by Google that delivers secure key storage and cryptographic operations within a hardware environment that is set apart from any other instances.

In contrast to HSMs, these hardware devices originated for the purpose of keeping cryptographic keys safe and executing cryptographic operations in a secure, tamper-resistant manner.

The Google Cloud HSM helps users manage and secure cryptographic keys when they are generated and stored remotely, guarding against data leakage and meeting regulations.

While the service provides FIPS 140-2 Level 3 HSMs (validating to the highest security standards of cryptographic modules), it operates under compliance with security, privacy, and quality standards.

Through Google Cloud HSM, users can securely administer keys used in practically every situation: encryption, digital or space signatures, and key management.

The service integrates with other Google Cloud services, enabling users to utilize the scalable and secure solution to set up cloud-based applications and take care of different workloads.

Recommended: Supported Cloud HSMs for Code Signing

Google Cloud HSM is specifically for fintech cases, healthcare and government, and situations that need the highest security and compliance.

The cloud computing environment can be made safer by employing hardware security modules that are purposely dedicated to the issue, simultaneously benefiting the

organization from the flexibility and scalability of cloud computing and the highest security standards of cryptography.

Google Cloud HSM Key Hierarchy

In Google Cloud KMS with the Cloud HSM backend, we implement the key hierarchy, which allows for a secure and robust key management system for governing encryption keys. Here is a detailed breakdown of the key hierarchy.

Root KMS:

Google's key management service within the organization establishes the root of the entire hierarchy of keys.

It plays the role of the overarching managerial and integrity measures of the key management system.

KMS Master Key:

The KMK is just what all the KEKs are encrypted with.

This key, which is concerned with security and stored in the memory, is distributed and protected.

KEK (Key Encryption Key):

KEKs are a mechanism used to encrypt DEKs.

They are in-built and are, therefore, critical in protecting the DEKs, which are managed within Cloud HSM.

DEK (Data Encryption Key):

The DEK is the password that users use to protect their information effectively.

These keys are stored and managed inside the Cloud HSM, a FIPS 140-2 Level 3 Certified Hardware Security Module.

Key Protection:

The private key of the KEK lives within the AWS HSM in such a way that only Google has access to the DEK unwrapping. It means that an attacker cannot enter and retrieve the EK key without Google's involvement.

In this scenario, the user's encryption keys will be one more protective layer.

BYOK (Bring Your Own Key) Feature:

BYOK option allows users to keep and import keys that they are already using into CloudKMS.

Imported keys can be used at HSM or software storage level either, completing the key management options.

Cloud KMS Components

Google's Cloud Key Management Service (Cloud KMS) is a cloud-hosted key management service by Google Cloud Platform (GCP). This tool helps you to manage your keys centrally and at a scale in between your cloud services.

The Cloud KMS Components include:

Key Rings:

Key Rings are the names of containers where cryptographic keys are stored. They facilitate a process for grouping keys right inside Google Cloud KMS. A Key Rings role is to segment keys with respect to security policies and needs, for example, access controls or compliance aspects.

Recommended: How to Configure Google CloudHSM to Sign Windows Executables?

Keys:

Keys are cryptographic entities which are being placed within public or private Key Rings. Cloud KMS permits symmetric encryption keys to be used for the encryption and decryption stages and for the asymmetric keys to be used for the signing and verification process. Keys can be generated, changed, and deleted within a KeyERing.

Key Versions:

The first is the generation of Key Versions, which represent different versions of a cryptographic key. The purpose of a key rotation is to anticipate that the old key may get hacked.

Recommended: How to Configure EV Code Signing Certificate on Google Cloud HSM?

When a key is rotated (i.e., updated), a new version of the key is created, and the previous version is retained for decryption purposes. This operation then enables one to switch keys uninterruptedly, regardless of whether encryption or decryption is in progress.

CryptoKeys:

CryptoKeys are cryptographic keys created by Cloud KMS for crypto operations. They are either symmetric or asymmetric versions. CryptoKeys is for a specific Key Ring followed by many versions.

CryptoKey Versions:

CryptoKey Versions stand for different CryptoKey variants. When CryptoKeys are rotated, new sets of these are created to smoothly carry out the key rotation, which provides backward compatibility for the decryption operations.

Key Policy:

Key Policy is a set of permissions that regulates the use of cryptographic keys by entities in the Cloud KMS. It gives power to these actors to perform activities like key generating, encrypting, decrypting, and deleting the keys. Key Policy is defined at the Key Ring level and remains applicable to all the keys lying within the Key Ring.

Service Accounts and IAM Roles:

Cloud KMS interacts with Google Cloud Identity and Access Management (IAM) to decide who gets access to keys and to which Key Rings. For instance IAM roles may be utilized to

grant access either for individual accounts or services in order to conduct encryption and management operations of applied keys.

Google KMS Purpose & Algorithms

Google Cloud Key Management Service (G Cloud KMS) provides an essential means for key management and performs cryptographic operations in a cloud environment. Its primary purpose includes:

Key Management:

Cloud KMS users are, therefore, able to produce, import, rotate, and destroy the cryptographic keys. It is the most sophisticated approach, allowing the implementation of keys at a centralized level for the encryption, rotating, signing, or verifying operations.

Data Protection:

Cloud KMS helps guard plaintext data in cloud services on Google Cloud Platform (GCP) including Cloud Storage, Big Query, Compute Engine, etc. by following the cryptographic keys handling requirements. It performs encryption on rest data and transit ones so as to make the data confidential and integral.

Recommended: How to Create and Validate Digital Signatures using Google Cloud Key Management Service?

Compliance:

Situated with the cloud KMS, companies meet the regulatory requirements and standards regulations related to data protection and encryption. It has varied features such as key storage, key rotation, and access controls to ensure compliance with data security rules.

Integration:

All service orchestration for Cloud KMS is natively run in the Google Cloud which is what makes it easy to implement and manage encryption and decryption within applications. It also helps in access to developers to integrate the offering with other external systems and applications such as APIs and client libraries.

Scalability and Performance:

Knowing that the cloud KMS can fulfill the cryptographic requirements of both small-scale and large-scale applications broadens the range of applications where such services can be used.

It is inherently fault tolerant, low-latency, and scalable; which are the exact features needed by today's cloud powered applications.

On the basis of cryptographic algorithms, Cloud KMS provides symmetric and asymmetric encryption algorithms for different applications.

Some commonly supported algorithm includes:

Symmetric Encryption:

Cloud KMS incorporated mathematically strong symmetric encryption algorithms, Advanced Encryption Standard (AES) with key sizes of 128, 256 or 512 bits. AES is among the most employed algorithms for encryption of sensitive data because of being both secure and efficient.

Asymmetric Encryption:

In the field of asymmetric encryption, Cloud KMS works with algorithms like RSA and Elliptic-Curve Cryptography (ECC). These algorithms are implemented for key exchange, signature generation and other cryptographic operations where the asymmetric system of public-private key pair is used.

Hashing:

The same is true for cloud KMS, SHA256, and SHA512 hash functions, which could be used to calculate the hash values of the data. Hashing is a ubiquitous tool for verifying data integrity or transforming passwords into hashes.

By being equipped with several symmetrical and asymmetrical encryption algorithms, Cloud KMS shows the necessary adaptability and variability to respond to the multiple security prerequisites of cloud-based applications and services.

Choose the Right Encryption for Your Needs.

The type of encryption algorithm to choose may vary based on your needs taking into account such things as security requirements, speed of performance, compatibility of the systems with existing ones, and regulatory compliance.

Here are some considerations to help you select the appropriate encryption for your needs:

Symmetric vs.Asymmetric Encryption:

Symmetric Encryption: This is useful in situations where the same person provides both encryption and decryption. Symmetric encryption algorithms, notably AES, are often twice as fast and more efficient than public key algorithms, and this is the reason why they are used for encrypting large volumes of data.

Asymmetric Encryption: Perfect for situations where one is engaged in key exchange, digital signature, and secure conversation between different participants. An RSA, ECC, etc., can use the public-private key pair technique to provide more security, but they are quite a bit slower than symmetric encryption.

Security Strength:

Think of the security quality provided by every kind of encryption algorithm. Taking the AES-256 vs. AES-128 as an example, it should be noted that the former uses a larger key size, providing greater security.

Correspondingly, the long key length for the asymmetric cryptographic technique increases the resistance against the brute-force approaches.

Performance and Efficiency:

Assess the functioning and efficacy of encryption techniques, particularly concerning the amount of overhead and resources used. Pick algorithms that give an acceptable balance between efficiency and safety supported by the requirements of your application.

Compatibility and Interoperability:

Ensure the selected encryption algorithm is consistent with the existing systems, operation protocols, and compliance standards.

Consider including options like pairing with cloud services, supporting industry standards, and running smoothly with other crypto libraries and tools.

Regulatory Compliance:

Figure out if your organization has to follow some regulatory requirement or some kind of standards that are related to encryption.

For example, GDPR, HIPAA, and PCI DSS require the use of particular encryption algorithms as well as key management methods that are inherently intrinsic to their ontology.

Future-Proofing:

When choosing encryption algorithms, consider forecasting future needs and technological developments. Figure out algorithms that are able to handle emergent threats and can be enhanced to meet your security needs over time.

How to Protect Data in Google Cloud?

Data in Google Cloud should be safeguarded by applying blended security best practices, computational encryption techniques, user access control methods, and continuous observations.

Here are some key steps to help you protect your data effectively:

Use Encryption:

Encrypt Data at Rest:

Consider the use of Google Cloud's encryption tools namely Cloud KMS (Key Management Service) and Customer-Managed Encryption Keys (CMEK), which serve the purpose of encrypting data stored in Google Cloud Storage, Compute Engine disks, and others.

Encrypt Data in Transit:

Secure data that is sent between applications at the Google Cloud and Google Cloud services, too, as Google Cloud employs TLS (Transport Layer Security).

Implement Access Controls:

IAM experience (Identity and Access Management) is leveraged to manage access authorizations of Google Cloud resources. Place users, service accounts, and groups in roles with the principle of least privilege, hence granting the users, service accounts, and groups the least number of permissions possible.

Using VPC Service Controls, you can configure certain access boundaries for Google Cloud resources within determined network perimeters, which may be used for network segmentation and, thus, help to prevent data exfiltration.

Monitor and Audit:

Enable Cloud Audit Logs capable of tracking and logging access to your Google Cloud resources like a data access facilitator and administrative activities. Stackdriver Logging and Monitoring must be used to analyze logs and to set up often to monitor suspicious activities.

Data Loss Prevention (DLP) API from Google Cloud is used in order to screen and categorize sensitive information, facilitate unauthorized exposure as well as meet regulations.

Secure Identities:

Impart a security best practice of strong authentication like multi-factor authentication (MFA) for user logins trying to access Google Cloud Console and APIs.

Leverage IAM service accounts and service account keys, which are created with appropriate permits for applications and systems running in Google Cloud instead of using users' credentials.

Network Security:

Leverage Google Cloud VPC to form your resources in a separate network environment. You will control and restrict traffic using configurable firewall rules with network segmentation.

Activate Google Cloud Armor as the first line of defense in blocking DDoS attacks based on networking and web-based application vulnerabilities.

Regularly Update and Patch:

You must ensure that your systems and installed software are always updated by implementing security patches or upgrades to prevent known defects and potential backdoors.

Data Residency and Compliance:

Specify your Google Cloud regions and places where data residence rules and compliance commitments are implemented.

Develop regulations for getting and using data to determine standards such as GDPR, HIPAA, and PCI DSS to ensure proper legislation application.

Conclusion

Implement Google Cloud HSM seamlessly and unleash the capabilities of GC to enhance the security of your apps.

SignMyCode offers cloud code signing certificates to ensure the consistency of your cryptographic keys as well as the integrity of your code.