DATA DEFENDERS APA PAPER DELIVERABLE: MARIANA ESPINOZA

TEAM MEMBERS: JAELIN LAZENBERRY, AWA AFO, TAKALA CROOK

- - - -

Abstract:

Encryption is a critical component of a defense-in-depth strategy, which is a security approach with a series of defensive mechanisms designed so that if one mechanism fails, there's at least one more still operating. As organizations look to operate faster and at scale, they need solutions for meeting critical compliance requirements and improving data security. Encryption, when used correctly, can provide an additional layer of protection above basic access control.

This is increasingly important in today's cybersecurity landscape where sensitive data is being targeted by threat actors, particularly as it relates to ransomware groups/attacks. As reported by SentinelOne in their end of year 2023 WatchTower report, the healthcare industry has been subjected to increased ransomware attacks. It is no surprise when considering the vast amounts of sensitive data that the healthcare industry handles, processes, and stores. And even currently, people across the United States were recently affected by the cyber attack on CDK global, where malicious actors who hacked the software company demanded tens of millions in ransom. This attack affected a major software company (CDK global) used by major car dealerships across the U.S. As such the attack likely compromised sensitive personal customer data such as social security numbers, addresses (current and former), employment history, and more.

Our project aims to focus on the secure encryption of sensitive data using AWS KMS and AWS CloudHSM as cloud solutions for data encryption and protection. Keys in AWS KMS or AWS CloudHSM can be used to encrypt data directly, or to protect other keys that are distributed to applications that directly encrypt data. As such, AWS KMS and AWS CloudHSM are both great solutions for organizations looking to safeguard their sensitive data.

- - - -

Introduction:

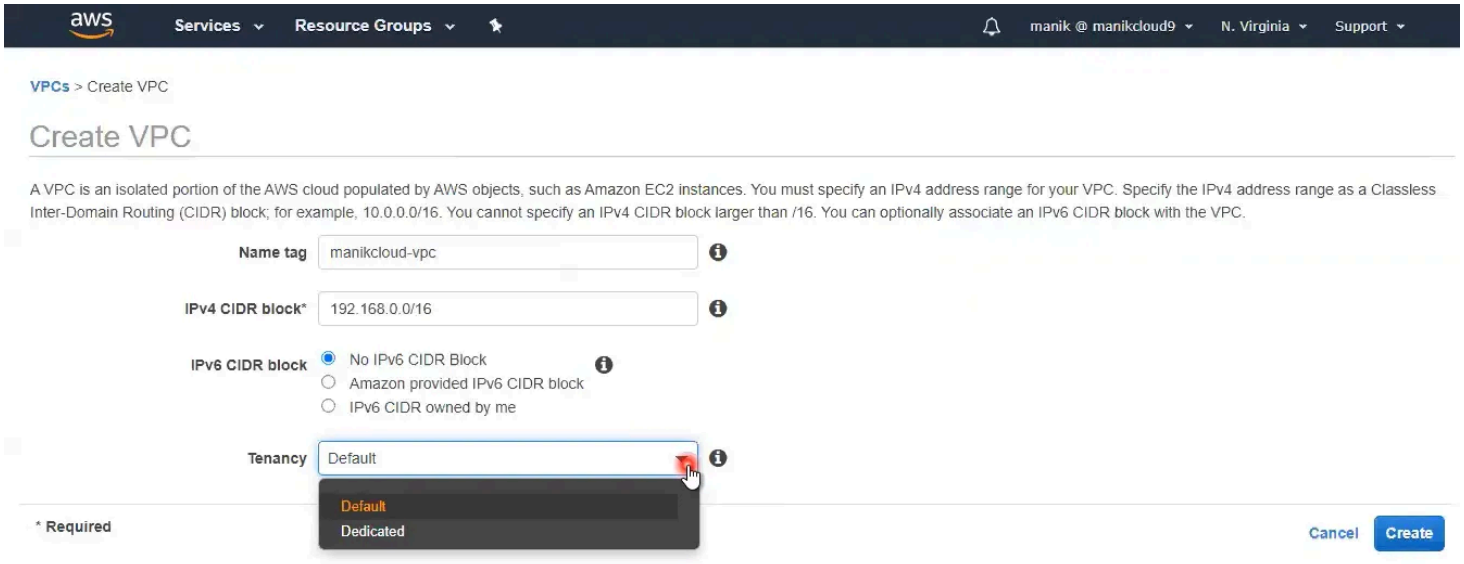
Our group, Data Defenders, encrypted sensitive data by using AWS KMS to create and manage our cryptographic keys for data encryption. We also provided an additional layer of security in compliance with standards like PCI DSS by having AWS CloudHSM clusters to manage and access our keys on FIPS-validated hardware, protected with customer-owned, single-tenant HSM instances that run in our own Virtual Private Cloud (VPC).

Lastly, we plan on using CloudTrail to collect logs of AWS KMS related events/activity which provides us with high level visibility for increased security, monitoring, and detection capability.

- - - -

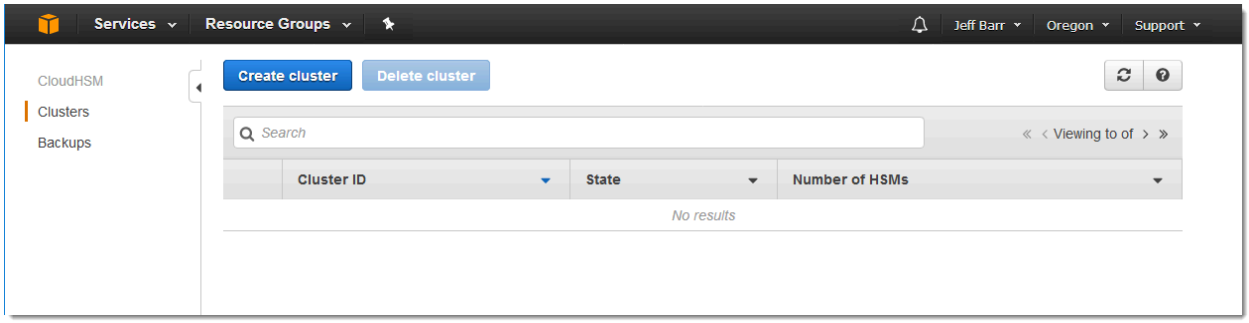
Methodology:

For our project, we first started out by setting up a VPC to logically isolate our cloudHSM clusters and key stores as well as our sensitive data storage, KMS keys, and all our related operations. For VPC creation, you can use the AWS VPC console and configure a name, an IPv4 CIDR block and more.



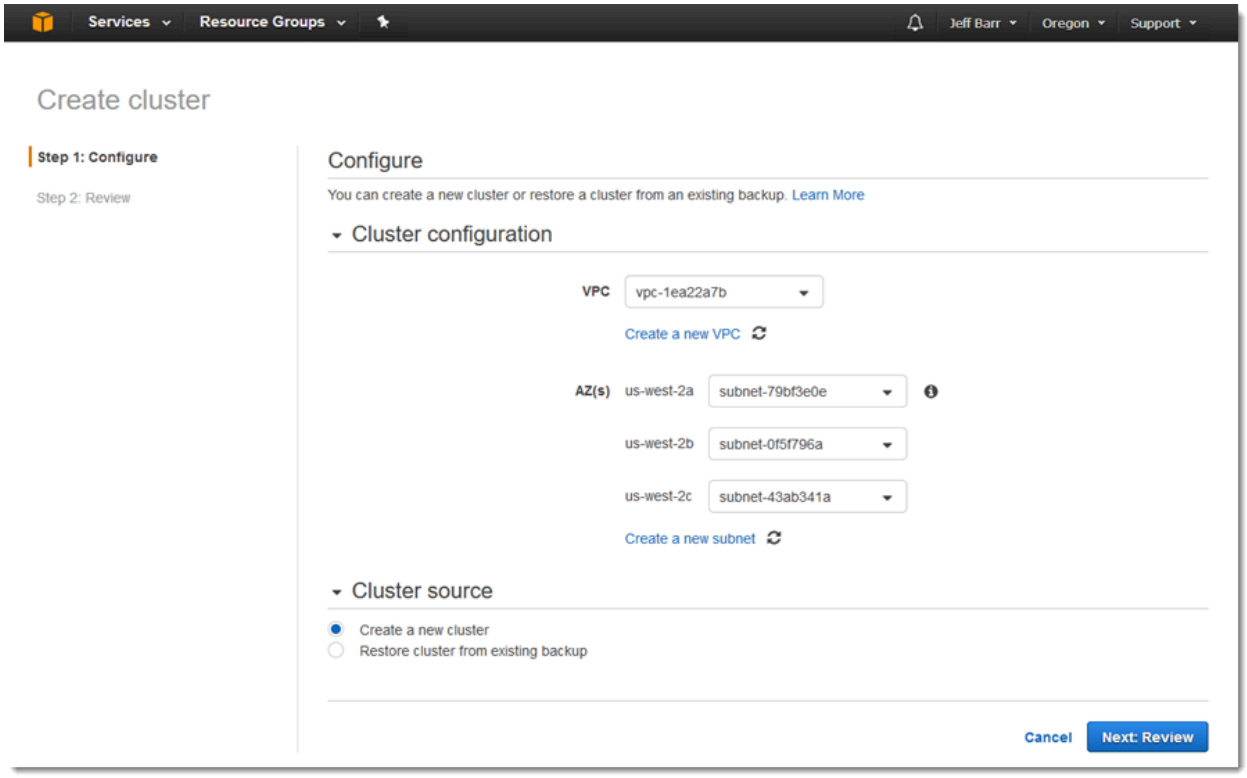
Once our VPC was set up, we were able to focus on setting up our CloudHSM cluster for use and integration with AWS KMS.

First, we went to the AWS CloudHSM console.



To create a CloudHSM cluster, choose create cluster and then configure as needed.

Sample Image of Cluster configuration:



Then we reviewed our settings and hit create.

Afterwards, we had to initialize our cluster and after doing so, our cloudHSM was ready to be used to protect our capstone project data.

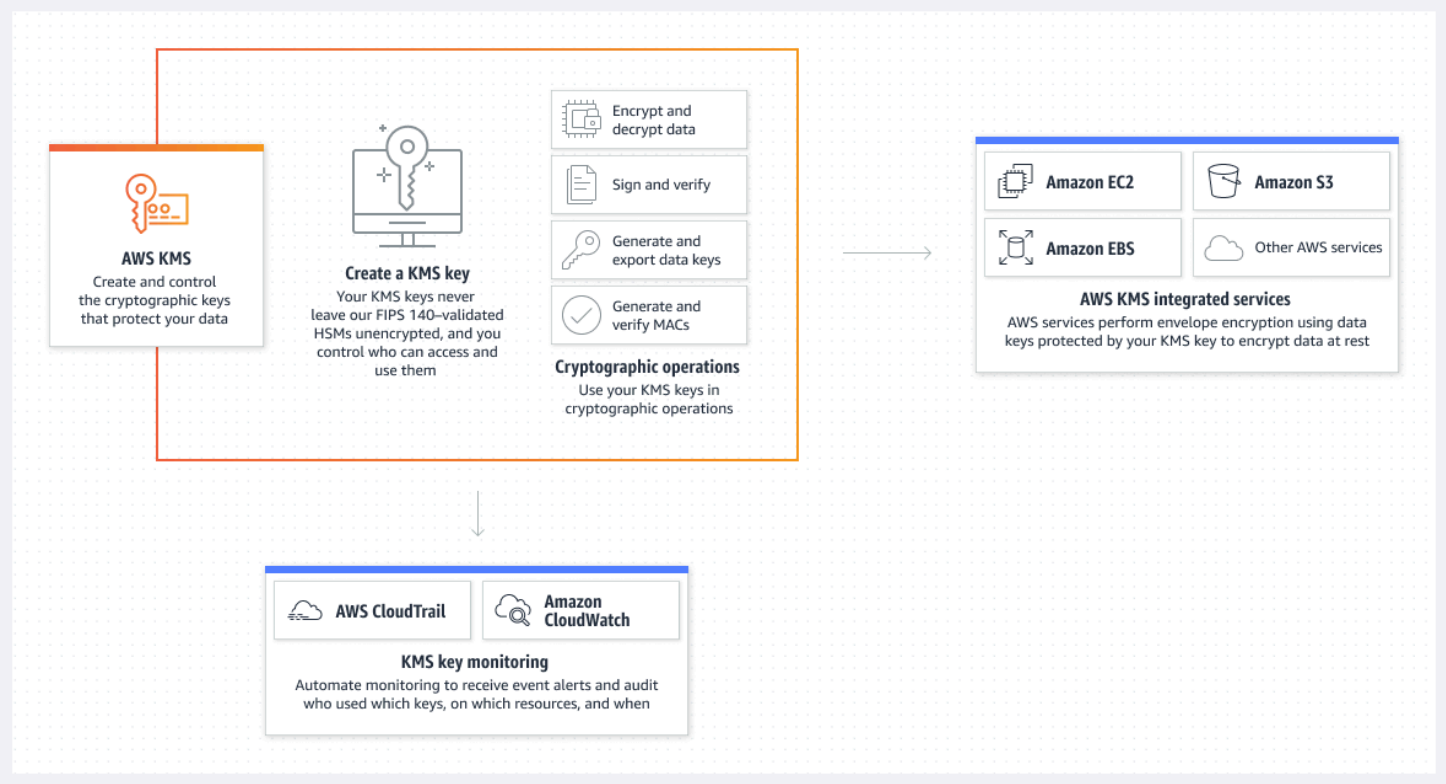
As a best practice for using AWS CloudHSM and CloudHSM clusters, AWS recommends placing at least two HSMs in a cluster, deployed in different AZs, to optimize data loss resilience and improve the uptime in case an individual HSM fails. If workloads grow, you may want to add extra capacity. In that case, it is a best practice to spread any new HSMs across different AZs to keep improving resistance to failure.

We used the AWS KMS console to create and manage our cryptographic keys for data encryption.

- - - -

Technologies:

AWS KMS is a cloud service that allows organizations to create and manage cryptographic keys for data encryption and protection.



AWS KMS uses hardware security modules (HSMs) to protect and validate all its cryptographic keys under the FIPS 140-2 Cryptographic Module Validation Program (CMVP). CMVP is a joint initiative between the Canadian Centre for Cyber Security and National Institute of Standards and Technology (NIST) to advance the use of validated cryptographic modules and provide public agencies with a standardized way to procure equipment containing validated cryptographic modules.

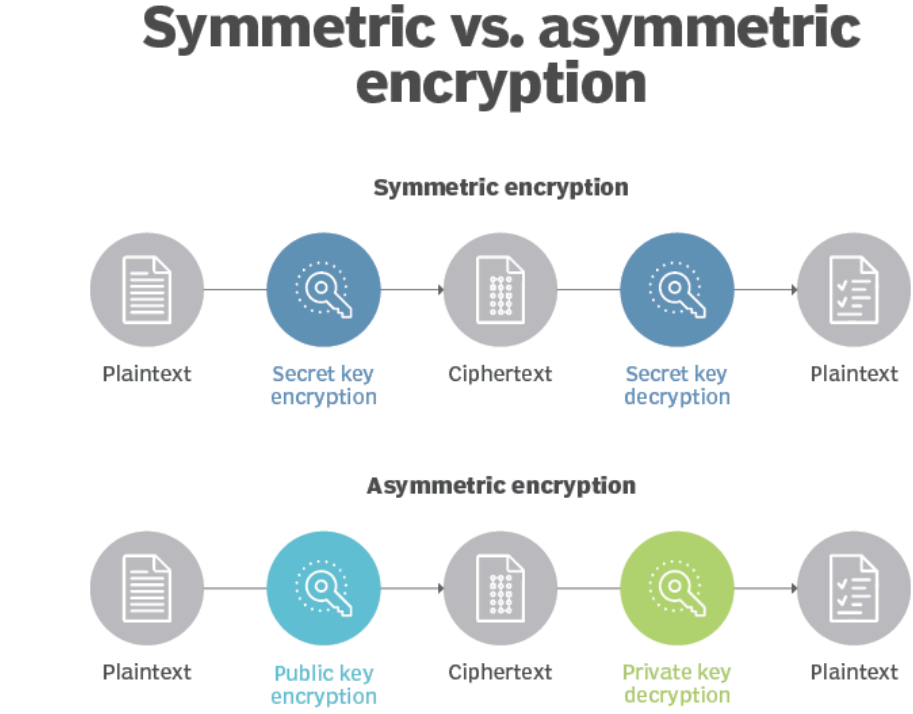
An AWS KMS key is a logical representation of a cryptographic key. There are three types of KMS keys that can be created in AWS KMS:

- Customer managed key. Created by an organization/customer.
- AWS managed key. Created by an AWS service that uses KMS keys to encrypt the AWS account service resources.
- AWS owned key. KMS keys created by AWS in a service account.

A KMS key contains:

- metadata (key ID, key spec, key usage, creation date, description and key state);
- reference to the key material used when cryptographic operations are performed with the KMS key.

A KMS key can be created with the cryptographic key material generated in AWS validated HSMs. KMS keys can be used for multiple cryptographic functions, such as data encryption, message signing and verification, generating exportable symmetric data keys and asymmetric data key pairs, and more.



KMS uses envelope encryption; a type of encryption that has two different keys for protecting data and encryption keys. Envelope encryption involves encrypting plaintext data with a data key, and then encrypting that data key with another key. This top-level encryption key, which is required to decrypt the keys and data, is known as the root key or AWS KMS key. All encryption keys are stored and managed securely in AWS KMS.

Encryption works by using an algorithm with a cryptographic key to convert data into an unreadable format (ciphertext) that can only be read again (aka converted back into plaintext) with the right key. For example, a text phrase like “Welcome!!” can look like “1c28df2b595b4e30b7b07500963dc7c” when encrypted. There are many different types of encryption algorithms, and they all use different keys. A strong encryption algorithm focuses on producing ciphertext that can’t be decrypted using a practically available amount of computing power such that you need to have the required key in order to achieve decryption. As such, protecting and managing the keys becomes a critical part of any encryption solution for organizations that deal with sensitive data.

An effective security strategy begins with stringent access control and continuous work to define the least privilege necessary for persons or systems accessing data. AWS requires that organizations manage their own access control policies, and also supports defense in depth to achieve the best possible data protection.

Even if an organization mistakenly created overly permissive access policies on their data, a well-designed encryption and key management system (such as one using AWS KMS and CloudHSM) could prevent this from becoming an issue, because it separates access to the decryption key from access to data.

AWS KMS and AWS CloudHSM can protect plaintext master keys on an organization’s behalf, but the organization is still responsible for managing access controls to determine who can cause which encryption keys to be used under which conditions.

One advantage of using AWS KMS is that the policy language you use to define access controls on keys is the same one you use to define access to all other AWS resources.

Although the language is the same, the actual authorization controls are not. You need a mechanism for managing access to keys that is different from the one you use for managing access to your data. AWS KMS provides that mechanism by allowing customers to assign one set of administrators who can only manage keys and a different set of administrators who can only manage access to the underlying encrypted data. Configuring the key management process like this helps provide separation of duties needed to avoid accidentally escalating privilege to decrypt data to unauthorized users.

For even further separation of control, AWS CloudHSM offers an independent policy mechanism to define access to keys.

Even with the ability to separate key management from data management, AWS still gives us ways to verify that we have configured access to encryption keys correctly. AWS KMS is integrated with AWS CloudTrail so customers can audit who used which keys, for which resources, and when. This provides granular visibility into encryption management processes, which is typically much more in-depth than on-premises audit mechanisms.

All AWS services integrated with AWS KMS use only symmetric encryption KMS keys to encrypt data. They do not support encryption with asymmetric KMS keys. These services include:

- AWS DynamoDB
- AWS S3
- AWS RDS
- AWS EBS

- - - -

Technologies:

AWS KMS can be used with AWS CloudTrail to log the use of organizational KMS keys. This is used to audit key usage and ensure compliance. Additionally, it can be used to increase visibility of KMS activity to provide additional security and detection capabilities. By default, all AWS KMS actions are logged as CloudTrail events. If a customer wants, they can exclude AWS KMS actions from a CloudTrail trail but that requires further action in changing a trail's configuration.

AWS CloudTrail is a service that records all calls to AWS KMS by users, roles, and other AWS services/resources.

CloudTrail captures all API calls to AWS KMS as events, including calls from the KMS console, AWS KMS APIs, the AWS Command Line Interface (AWS CLI), and more.

CloudTrail logs all AWS KMS actions, including read-only, such as ListAliases and GetKeyRotationStatus, operations that manage KMS keys, such as CreateKey and PutKeyPolicy, and cryptographic operations, such as GenerateDataKey and Decrypt.

CloudTrail also logs processes that AWS KMS calls for you, such as DeleteExpiredKeyMaterial, DeleteKey, SynchronizeMultiRegionKey, and RotateKey.

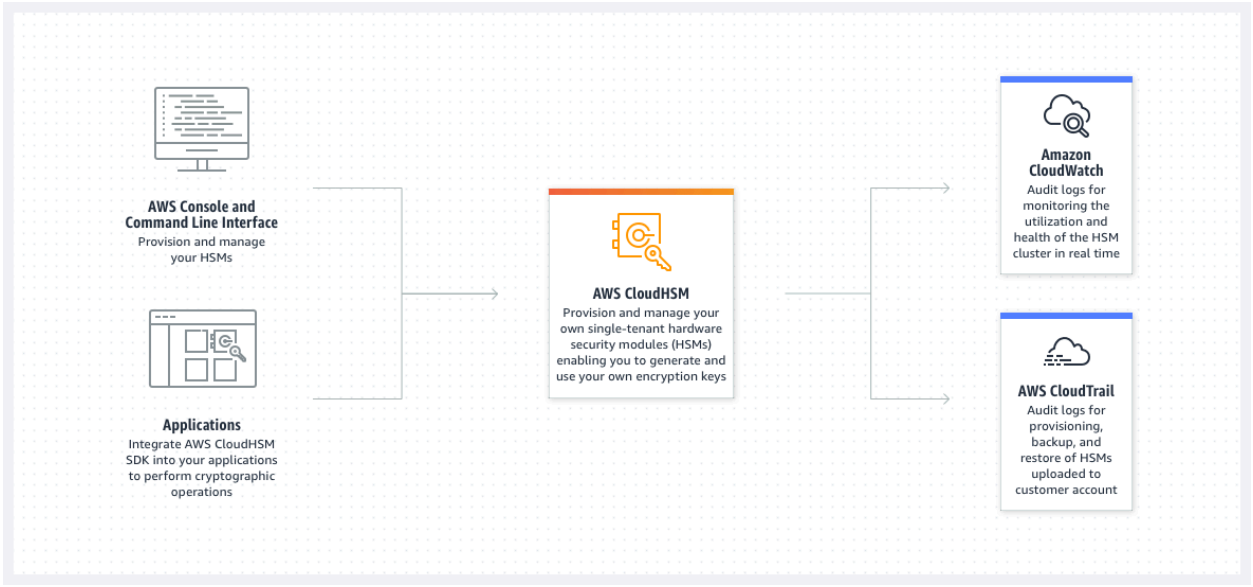
For security reasons, some fields are omitted from AWS KMS log entries, such as the Plaintext parameter of an Encrypt request, and the response to GetKeyPolicy or any cryptographic operation.

AWS KMS adds the key ARN of the affected KMS key to the responseElements field in CloudTrail log entries for some AWS KMS key management operations, even when the API operation doesn't return the key ARN. This key ARN is included by KMS to certain CloudTrail log entries for key management events to make it easier to search for log entries for specific KMS keys.

- - - -

Technologies:

AWS CloudHSM helps organizations meet corporate, contractual, and regulatory compliance requirements for data security.

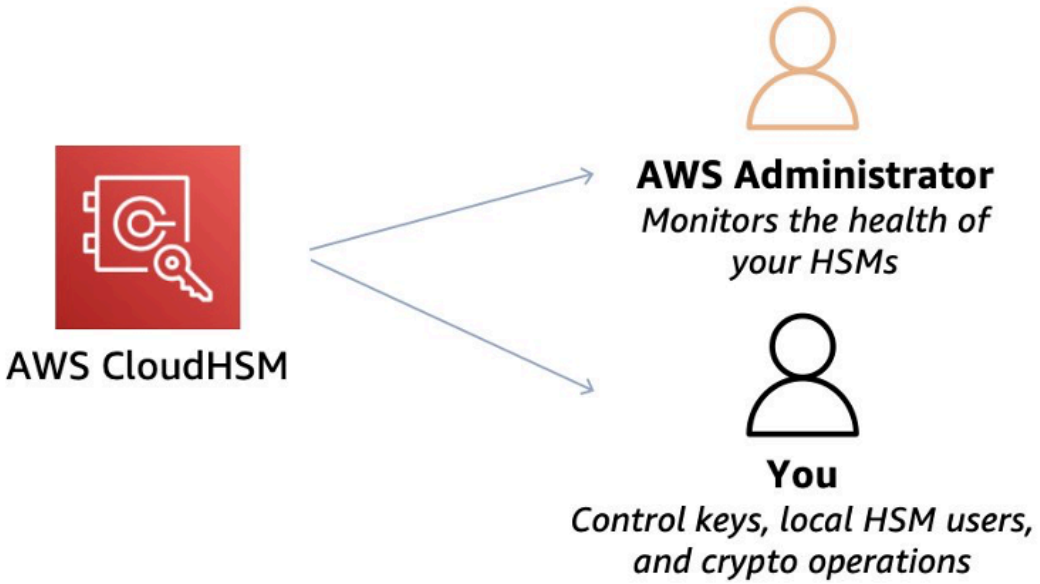


The best practice to maximize cryptographic key security is using a hardware security module (HSM). This is a specialized computing device that has several security controls built into it to prevent encryption keys from leaving the device in a way that could allow threat actors to access and use those keys.

One security control in modern HSMs is tamper response, which is when the device detects physical or logical attempts to access plaintext keys without authorization, and destroys the keys before the attack succeeds. Because organizations can’t install and operate their own hardware in AWS data centers, AWS offers two services using HSMs with tamper response to protect customers’ keys: AWS Key Management Service (KMS), which manages a fleet of HSMs on the customer’s behalf, and AWS CloudHSM, which gives customers the ability to

manage their own HSMs. Each service can create keys on an organization's behalf, or they can import keys from their on-premises systems to be used by each service.

AWS CloudHSM lets organizations manage and access their cryptographic keys on FIPS-validated hardware, protected with customer-owned, single-tenant HSM instances that run in a customer's own Virtual Private Cloud (VPC). This allows organizations to have additional security when protecting and encrypting sensitive data and allows them to have regulatory compliance with industry standards like PCI DSS for storing and handling certain kinds of sensitive/personal data such as credit card data in the case of PCI DSS.



AWS CloudHSM allows organizations to generate and use encryption keys using FIPS 140-2 Level 3 validated hardware security modules (HSMs) on the AWS Cloud. CloudHSM integrates with client applications using industry-standard APIs, such as PKCS#11, Java Cryptography Extensions (JCE), and Microsoft CryptoNG (CNG). It is also standards-compliant and enables organizations to export all of their keys to most other commercially-available HSMs. CloudHSM is a fully-managed service that automates time-consuming administrative tasks for AWS customers, such as hardware provisioning, software patching, high-availability, and backups. With CloudHSM, organizations can add and remove HSM capacity on-demand, with no up-front costs.

All interaction with a hardware security module (HSM) takes place through the AWS CloudHSM client. The AWS CloudHSM client runs on an EC2 instance and uses certificate-based mutual authentication to create secure (TLS) connections to the HSMs.

At the hardware level, each HSM includes hardware-enforced isolation of crypto operations and key storage. Each customer HSM runs on dedicated processor cores to ensure compliance with industry standards and regulations.

Applications can use the APIs in AWS CloudHSM SDKs to manage keys, encrypt & decrypt objects, and more. The SDKs provide access to the CloudHSM client (running on the same instance as the application). The client, in turn, connects to the cluster across an encrypted connection.

Because CloudHSM clusters are deployed inside an Amazon VPC, customers should use the familiar controls of Amazon VPC security groups and network access control lists (network ACLs) to limit what instances are allowed to communicate with their CloudHSM clusters. Even though a CloudHSM cluster itself is protected in depth by AWS account login credentials, Amazon VPC offers a useful first line of defense. Because it's unlikely that an organization will need its communications ports to be reachable from the public internet, it's a best practice to take advantage of the Amazon VPC security features.

- - - -

Conclusion:

AWS KMS and AWS CloudHSM are two great cloud solutions offered by AWS for securing and encrypting sensitive data properly and in line with industry standards and regulations.

When users combine the use of AWS KMS with CloudHSM cluster integration for key stores and more along with strict access controls following the principle of least privilege and enforcing separation of duties, sensitive data can be encrypted securely and cryptographic keys can be managed and stored with multi-faceted defense mechanisms in place.

In our project, we used IAM to control access & permissions to both AWS services in general as well as to control permissions and separate duties regarding specific AWS KMS keys like the key we used to encrypt the trail we created to log KMS events and usage on the AWS CloudTrail console.

We then used AWS KMS to create and manage our cryptographic keys used for encryption operations and used CloudHSM to ensure we met compliance regulations and industry security standards for the management and storage of our keys used to encrypt data. The HSMs provided by AWS CloudHSM are FIPS 140-2 level 3 certified and are recognized as complying with PCI DSS requirements.

Sources:

Amazon Web Services, Inc. (n.d.). *AWS CloudHSM use cases*.

https://docs.aws.amazon.com/cloudhsm/latest/userguide/use-cases.html

Amazon Web Services, Inc. (n.d.). *AWS CloudHSM Documentation*.

https://docs.aws.amazon.com/cloudhsm/

Amazon Web Services, Inc. (n.d.). *AWS Key Management Service Documentation*.

https://docs.aws.amazon.com/kms/

Amazon Web Services, Inc. (n.d.). *AWS Key Management Service features*.

https://aws.amazon.com/kms/features/?pg=ln&sec=c/#AWS_Service_Integration

Amazon Web Services, Inc. (n.d.). *Best practices for AWS CloudHSM*.

https://docs.aws.amazon.com/cloudhsm/latest/userguide/best-practices.html

Amazon Web Services, Inc. (n.d.). *How AWS CloudTrail uses AWS KMS*.

https://docs.aws.amazon.com/kms/latest/developerguide/services-cloudtrail.html

Amazon Web Services, Inc. (n.d.). *Managing Keys - AWS Key Management Service*.

https://docs.aws.amazon.com/kms/latest/developerguide/getting-started.html

Grantham-Philips, W. (2024, June 24). Car dealerships in North America revert to pens and paper after cyberattacks on software provider. *The Associated Press*.

https://apnews.com/article/car-dealerships-cyberattack-cdk-outage-3f7c81f6be0e212172b33cdc9f49feba

Manik, V. K. (2020, August 3). How to Create AWS VPC in 10 steps, less than 10 min. *Medium*.

https://varunmanik1.medium.com/how-to-create-aws-vpc-in-10-steps-less-than-5-min-a49ac12064aa