What is AWS KMS?

Unencrypted data can be read and seen by anyone who has access to it and is known as plain text or clear text data. The data is plain to see and could be seen and understood by any recipient. There is no problem with this as long as the data is not sensitive in any way and doesn't need to be restricted. However, on the other hand, if you have data that is sensitive and you need to ensure that the contents of that data is only viewable by a particular recipient or recipients, then you need to add a level of encryption to that data. But what is data encryption?

Data encryption is the mechanism in which information is altered, rendering the plain text data unreadable through the use of mathematical algorithms and encryption keys. When encrypted, the original plain text is now known as cipher text which is unreadable. To decrypt the data, an encryption key is required to revert the cipher text back into a readable format of plain text. A key is simply a string of characters used in conjunction with the encryption algorithm and the longer the key the more robust the encryption

The Key Management Service (KMS) is a managed service used to store and generate encryption keys that can be used by other AWS services and applications to encrypt and decrypt your data. So it’s a fundamental security service offered by AWS to help you manage your cryptographic operations

Due to the nature of this service, the contents of which contains highly sensitive information administrators at AWS do NOT have access to your keys within KMS and they cannot recover your keys for you should you delete them. AWS simply administers the underlying operating system and application. As AWS has no access to your keys, it's our responsibility as the customer and users of the KMS service to administer our own encryption keys and administer and restrict how those keys are deployed and used within our own environment against the data that we want to protect.

It is important to understand that the KMS service is for encryption at rest only which can include for example S3 Object Storage, RDS, and EBS Encryption to name a few. KMS does not perform encryption for data in transit or in motion. If you want to encrypt data while in transit, then you would need to use a different method such as SSL.

When working with encrypted data, compliance and regulations are often tightly integrated. As a result, KMS works seamlessly with AWS CloudTrail to audit and track how your encryption keys are being used and by whom in addition to other metadata captured by the APIs used such as the source IP address, etc.

When architecting your environment with regards to data encryption, you need to be aware that AWS KMS is not a multi-region service like IAM is for example. It is region specific. Therefore, if you are working in a multi-region system with multi-region failover, you need to establish a Key Management Service in each region that you want to encrypt data.

Key Components of KMS

There are 4 key components of AWS KSM:

1. Customer Master Keys (CMK)
2. Data Encryption Keys (DEKs)
3. Key Policies
4. Grants

Customer Master Keys (CMKs)

This is the main key type within KMS and it contains the key material that is used to encrypt your data.

There are 3 different types of CMK:

1. Customer Managed
2. AWS Managed
3. AWS Owned

What is AWS KMS?

Customer Managed CMKS: These keys offer the greatest level of flexibility and control. You are able to create, disable or delete the key, configure the key policies associated with your key, configure Grants, and also alter and adjust the key rotation periods and view full usage history of the key. These keys can be used by other AWS services that integrate with KMS. Customer managed keys include an additional charge for creating your customer CMKs

AWS Managed CMKs: These are managed by AWS, however you are still able to view these keys within the Management Console, and also audit and track their usage and view their key policies. However, because they are managed by AWS you are not able to modify them. These keys are created and used by AWS services that integrate with KMS directly, but they can only be used by the service that creates them.

AWS Owned CMKs: These are not visible within the KMS console or anywhere within your account, neither do you have the ability to audit and track their usage, they are essentially abstracted from your AWS account. But of course, some services use this key type to encrypt your data within your account. Examples of AWS Owned CMKs include:

- The S3 Master key uses SSE-S3 encryption
- The default encryption option used on all Amazon DynamoDB tables uses AWS owned keys

Data Encryption Keys (DEKs)

Data keys are created by CMKs however they are used outside of KMS to perform encryption against your data, either in your own applications or by other AWS services.

When a request to generate a data key is received by KMS, the associated CMK in the request will create 2 identical data encryption keys, one will be a plaintext key, and the other will be an encrypted key

During the encryption process, it's the plaintext data key that will be used to perform the encryption of your data using an encryption algorithm. Then once the encryption has taken place, this plaintext data key will then be deleted and the encrypted data key will be stored and associated with the newly encrypted data.

Key Policies:

Key policies determine who can do what with the key, for example, defining who can use the key to perform cryptographic operations, in addition to those who can administer the CMK to perform functions such as deleting and revoking the key. Controlling access to CMKs can't be done using IAM alone. In ALL cases, to manage access to your CMKs, you MUST use a key policy.

Grants:

Grants allow you to programmatically delegate your permissions to another principal or user, and so the grant consists of 2 parties, the user who creates the Grant, and the Grantee who then uses that grant to perform cryptographic operations.

For more information on how to implement and configure KSM, please see our existing course:
How to Use KMS Key Encryption to Protect Your Data:
https://cloudacademy.com/course/amazon-web-services-key-management-service-kms/

Stuart Scott
AWS Content & Security Lead

@Stuart_A_Scott
https://uk.linkedin.com/in/stuartanthonyscott