KMS & Envelope Encryption

Encryption is used for protecting your data at rest and during transit. Here, data security hinges on properly protecting the keys used for encryption and decryption.

In a traditional setup, developers use a single key and encrypt/decrypt data with it. In order to protect the key itself, they have to go through hassle of developing variety of techniques to safeguard the key. However, with these schemes, it is difficult to provide perfect forward secrecy. i.e. if your key is compromised in the future, how do you protect all your previously encrypted data?

To address this, AWS uses what is known as envelope encryption. With envelope encryption, multiple keys are involved in keeping your data secure. One key is used for encrypting the data (let's call it data_key), and another master key is used for encrypting the data_key itself. Master key never leaves the KMS Service. Any encryption/decryption operation that involves the master key is performed directly inside the KMS. Master key is referred using an Alias or by using key ARN. You can control who can use these master keys using IAM and Resource level policies. The authorized users cannot access the key material itself; they are simply allowed to use the key by referring to the alias or ARN.

For example, when you enable encryption in S3, S3 uses a data_key for encrypting an object. This data_key is unique for a particular S3 object. If there are 10 objects to be encrypted, each object will get its own unique data_key. Further, the data_key itself is encrypted with a KMS master key. This encrypted data_key is stored along with encrypted object. You can visualize a mailing envelope containing encrypted object and encrypted data key; hence the name Envelope encryption.

To decrypt the object data, S3 has to first decrypt the encrypted data_key; it does so by calling KMS service. If the caller is authorized to access the specific master key, KMS would decrypt and return the data_key. S3 uses the data_key to decrypt the object and return the object back to the caller.

To summarize, envelope encryption scheme has:

1. A unique data_key for service specific data granularity.
    1. For S3, it is every object.
    2. For EBS, it is at volume level.
2. Master key never leaves KMS. Any encryption or decryption that uses master key is performed directly inside the KMS Service
3. Master key itself can be optionally configured for rotation. AWS would automatically rotate the master keys for you while keeping all the previous master key versions. AWS automatically uses correct version of the master key to decrypt your data. For encryption,

it always uses the latest master key version. (https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html)

4. Pricing: Each master key that you create in KMS (referred to as CMK – Customer Master Key) incurs USD 1 per month until you delete it. If you enable key rotation, each newly rotated version will increase cost by USD 1 per month. There is no charge for keys that are managed by AWS. (https://aws.amazon.com/kms/pricing/)

Demo (Do it yourself)

Note: For help/question regarding this demo, please post your questions in this thread:

https://www.udemy.com/aws-certified-solutions-architect-guide-question-bank-i/learn/v4/questions/3461936

1. From IAM service, create a new customer managed key in your region. Give it an alias `DemoKey`. Grant allow access to your myadmin account.
2. From S3 console, upload a file to your S3 bucket. When uploading specify the "encryption" option and select "DemoKey" for encryption
3. From AWS Command Line tool, try to access the encrypted file that you uploaded and copy the file to your local folder. `aws s3 cp s3://<yourbucket>/filename filename --profile myadmin`
4. Verify if you are able to read the file content that was copied to your local folder from S3
5. Now create another IAM user "S3DemoUser" with programmatic access and attach `AmazonS3ReadOnlyAccess` policy.
6. Configure "S3DemoUser" using aws configure. `aws configure --profile S3DemoUser`. Provide the access key and secret access key
7. "S3DemoUser" now has access to read all your buckets and objects; however, this user does not have privileges to access the "DemoKey". So, any object that is encrypted with "DemoKey" is not accessible for this user. Try to read any unencrypted object that is stored in your S3 bucket, the user should be able to read the object. However, when the user attempts to read the encrypted object created previously, user will not be able to access it. `aws s3 cp s3://<yourbucket>/filename filename --profile S3DemoUser`
8. Cleanup:
    1. Delete the objects created in this demo
    2. Disable "DemoKey" from KMS from actions
    3. Schedule "DemoKey" deletion from KMS from actions. Delete this key to avoid incurring monthly key charges.