

## Overview:



### How Can S3 Security Threats Be Mitigated?

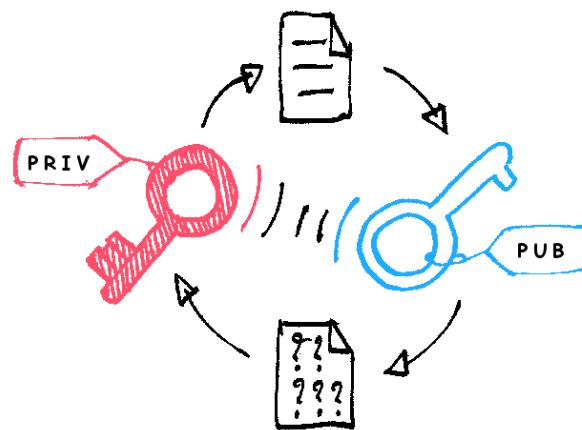
Security threats can be mitigated by encrypting the data stored in the S3 Bucket.

### Why Is This Important?

Creating an unencrypted S3 object is unsafe for the user and for everyone else who uses the information that is being uploaded. This data encapsulated in the S3 object needs to be encrypted with a unique key where this key will then be encrypted by a unique master key. The master key is used to protect all of the data and the information which is stored in the S3 object.

### What Is An S3 Bucket?

Amazon S3 stands for Simple Storage Solution, these S3 buckets are accessible from the Internet. The AWS security controls are used protect all of the resources. Some resources include: security groups, network access control lists. These resources however do not protect the data in the S3 bucket. Although, there are many ways to protect the confidentiality, integrity and availability of this data.



### What Is Encryption?

Encryption is defined as the process of converting plain text data into ciphertext so as to protect the information from being accessed by unauthorized users. There are two main ways for data to be encrypted, first while the data is travelling over the Internet, second when it is stored on the servers. Data is encrypted and stored in the cloud so as to safeguard sensitive information that is travelling through internet networks, the Internet and wireless mobile devices. It is essential for data to be encrypted in the cloud because enterprises and general information security controls do not have access to control data on the cloud. Also, HIPAA and PCI DSS are regulations developed by the industry and require data security to be compliant with their laws.

### Discovery:

A vital part of the security process which protects the data in the S3 is auditing. Amazon S3 buckets can be configured by storage administrators to log details of all the requests that are made to a specific bucket. The log files then can be stored in other Amazon S3 buckets with different access control permissions to reduce chances of tampering.

Services ▾ Resource Groups ▾ ⌵

Amazon S3 > this-is-my-bucket810

Objects Properties Permissions Management

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload + Create folder More ▾ All Deleted objects

US East (N. Virginia) ↻

Viewing 1 to 100 >

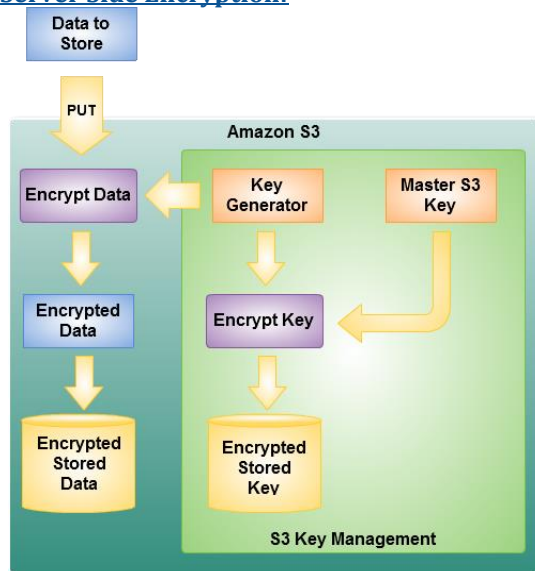
<input type="checkbox"/>	Name ↑ ▾	Last modified ↑ ▾	Size ↑ ▾	Storage class ↑ ▾
<input type="checkbox"/>	2017-03-08-16-20-10-FCB222F95D113038	Mar 8, 2017 11:20:11 AM	570.0 B	Standard
<input type="checkbox"/>	2017-03-08-16-20-36-612D0879CC216CD5	Mar 8, 2017 11:20:37 AM	571.0 B	Standard
<input type="checkbox"/>	2017-03-08-16-20-48-551860BB6602207E	Mar 8, 2017 11:20:49 AM	545.0 B	Standard
<input type="checkbox"/>	2017-03-08-16-23-29-F522F90A5CC0E551	Mar 8, 2017 11:23:30 AM	568.0 B	Standard

It is important to keep an eye on the logs in the S3 Bucket to be sure everything is running as it should.

### Remediation:

Remediation for creation of unencrypted S3 objects would be to use Server Side encryption. Explained in more detail in the prevention section, server-side encryption can encrypt unencrypted data that is already on AWS.

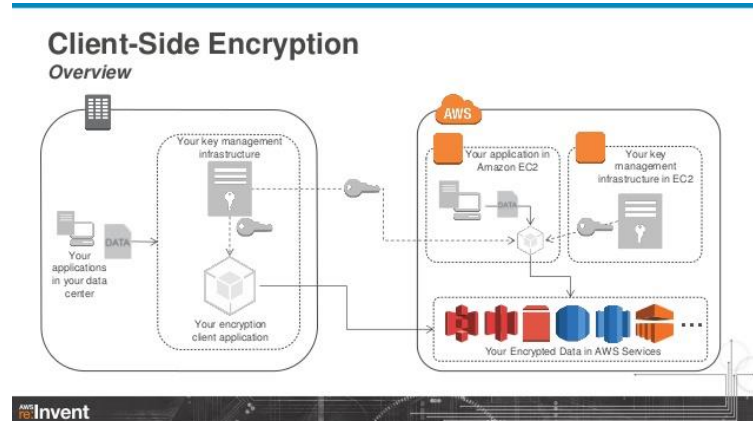
### Server Side Encryption:



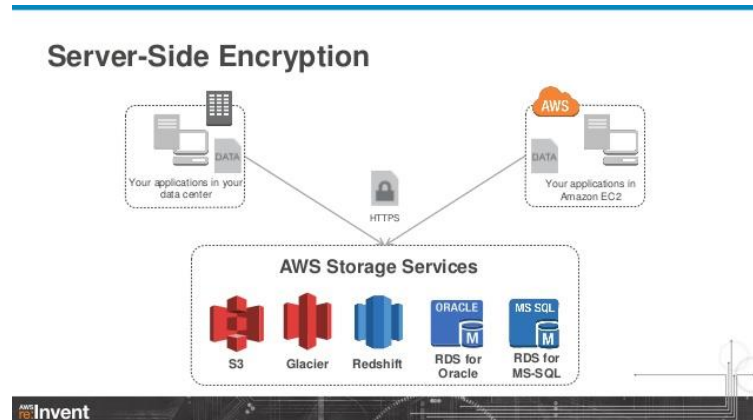
## Prevention:

Two ways to encrypt the data stored on Amazon S3.

- Client-Side Encryption: Encrypt the data before it is sent over the network to AWS



- Server-Side Encryption: Encrypt the data while on AWS



## Client-side encryption:

Client-side encryption is used to encrypt the data before it is sent over the network to AWS.

The encryption process is managed by the client side application, so the plaintext data and the master encryption key. None of the data ever leaves the client application.

There are 2 client side encryption options:

1. Use AWS KMS-managed customer master key
2. Use client-side master key

### Option 1: Use AWS KMS-Managed Customer Master Key (CMK)

When AWS KMS (Key Management Service) is being used, encryption keys do not need to be provided to Amazon S3 encryption client.

AWS KMS manages the customer key for the client-side data encryption.


## Welcome to AWS Key Management Service

AWS Key Management Service (KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS Key Management Service is integrated with other AWS services including Amazon EBS, Amazon S3, and Amazon Redshift, to make it simple to encrypt your data with encryption keys that you manage. AWS Key Management Service is also integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs.

You can create your first encryption key by clicking on the Get Started Now button below.


[Get Started Now](#)

### Key Management Service Concepts




#### Create and Manage Keys

AWS Key Management Service provides a single place to manage your organization's encryption keys. KMS presents a single view for all of the key usage in your organization. Easily implement key creation, rotation, usage policies, and auditing to help keep all of your encryption key management in one place.



#### Use Keys to Encrypt Your Data

AWS Key Management Service makes it easy to use managed encryption in your own applications. KMS provides an SDK for simple integration of encryption into your applications, whether they are run in the cloud, in a private server, or even in a mobile device. KMS provides seamless integration with AWS services like Amazon Simple Storage Solution (S3), Amazon Elastic Block Storage (EBS), and Amazon Redshift.



#### Audit Key Usage

AWS Key Management Service provides audit trail information directly to AWS CloudTrail. These audit trails help you meet compliance and regulatory requirements by providing logs of who used which key to access which data and when that access occurred.

When using this, you only need to provide an AWS KMS customer master key ID (CMK ID) and the client will handle the rest.

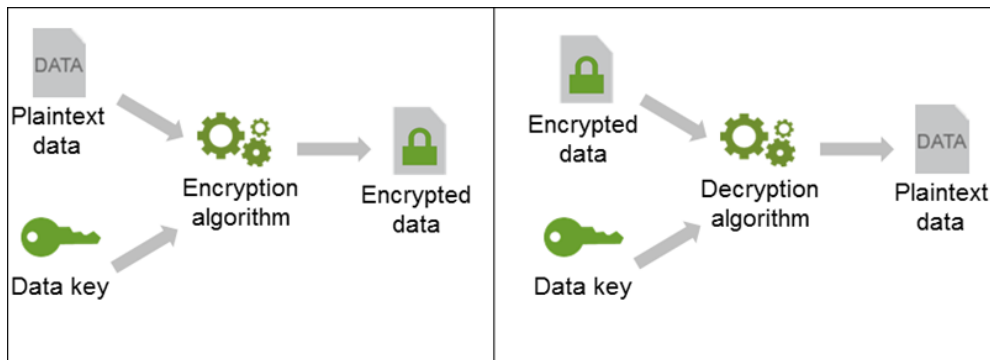
<http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

### When uploading an object

Using CMK ID, the clients can send a request to AWS KMS requesting for a key that it can use to encrypt the object data. In response to the request, AWS KMS returns a randomly generated data encryption key. There are 2 versions of AWS KMS data encryption keys.

- Plain Text Version
  - Client uses the key to encrypt the object data
- Cipher Blob of the same data encryption key
  - Client uploads data to Amazon S3 as object metadata
- **Note:** The client obtains a unique data encryption key for each object uploaded
- Example: Client-Side Encryption (Option 1: Use an AWS KMS – Managed Customer Master Key (AWS SDK for Java)).

Example of how Amazon SDK works:



### When Downloading An Object

First the encrypted data and the cipher blob version of the encrypted data, which is stored as an object metadata, is downloaded from Amazon S3 by the S3 client. The cipher blob is then sent by the client to AWS KMS to get the plain text version and decrypt the object data.

### Option 2: Use Client-Side Master Key

**\*\* Note:** When using the client-side master key, it is vital that you safely manage the encryption key because they were never sent to AWS. If this key is lost, then it will not be possible to decrypt your data.

How to provide client-side master key in the client-side data encryption process:

#### When uploading an object

Provide the client-side master key to the Amazon S3 encryption client i.e. `AmazonS3EncryptionClient` when using the AWS SDK for java. The client then uses the master key to encrypt the data encryption key which is generated randomly.

Process to encrypting the data encryption key:

- Amazon S3 encryption client locally generated a symmetric key which is a one-time key (the data encryption key or data key). This data key is then used to encrypt the data of a single S3 object (the client generated a separate data key for each object).
- The client encrypts the data encryption key by using the master key that was provide.
- As part of the object metadata, the client uploads the encrypted data key and its material description. The material description helps the client to later determine which client-side master key to use for decryption (the client has to decrypt the downloaded object).
- The client then uploads the encrypted data to Amazon S3. The encrypted data is saved as object metadata (`x-amz-meta-x-amz-key`) in Amazon S3 by default.

### When downloading an object

The encrypted object is first downloaded from Amazon S3 along with the metadata. Using the material description in the metadata, the client then determines which master key to use to decrypt the encrypted data key. Finally using the master key, the client decrypts the data key and uses it to decrypt the object.

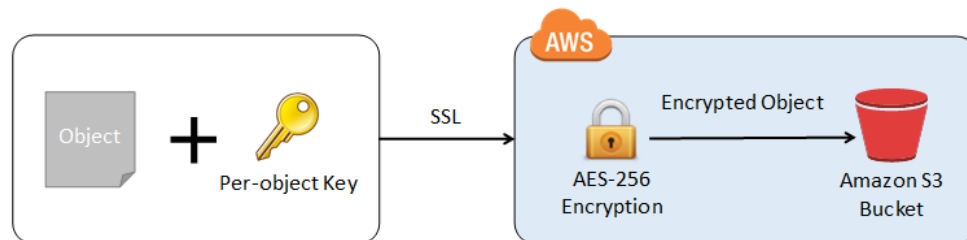
<http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

### Server-side encryption

The server-side encryption is used to protect the data that is at rest. Amazon S3 encrypts the data at the object level as it writes the data to disks in its data centers and decrypts the data for you when you access it. As long as you authenticate your request and you have access to permissions, there is no difference in the way you access encrypted or unencrypted objects because AWS provides three server-side encryption options.

#### Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

Each object is encrypted with a unique key with strong multi-factor encryption. As an additional safeguard, the object encrypts the key itself with a master key which it changes regularly. Amazon S3 server-side encryption uses 256-bit Advanced Encryption Standard (AES-256), which is one of the stronger cipher blocks, to encrypt the data. For more information, see Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).



#### **Step By Step (shown in image above):**

To put the object in the S3 Bucket:

1. Supply your encryption key as part of a PUT and S3 will take care of the rest. It will:
  - a. Use your key to apply AES-256 encryption to your data
  - b. Compute a one-way hash (checksum) of the key
  - c. Expediently remove the key from memory.
  - d. It will return the checksum as part of the response
  - e. Store the checksum with the object. Here's the flow:

To get the object from the S3 Bucket:

1. Supply the same key as part of a GET.
2. S3 will verify the stored checksum matches that of the supplied key
3. S3 will decrypt the object
4. S3 will return the decrypted and remove the key from memory.

#### Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

AWS KMS manages the customer key for the client-side data encryption.

## Welcome to AWS Key Management Service

AWS Key Management Service (KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS Key Management Service is integrated with other AWS services including Amazon EBS, Amazon S3, and Amazon Redshift, to make it simple to encrypt your data with encryption keys that you manage. AWS Key Management Service is also integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs.

You can create your first encryption key by clicking on the Get Started Now button below.

[Get Started Now](#)

### Key Management Service Concepts



#### Create and Manage Keys

AWS Key Management Service provides a single place to manage your organization's encryption keys. KMS presents a single view for all of the key usage in your organization. Easily implement key creation, rotation, usage policies, and auditing to help keep all of your encryption key management in one place.



#### Use Keys to Encrypt Your Data

AWS Key Management Service makes it easy to use managed encryption in your own applications. KMS provides an SDK for simple integration of encryption into your applications, whether they are run in the cloud, in a private server, or even in a mobile device. KMS provides seamless integration with AWS services like Amazon Simple Storage Solution (S3), Amazon Elastic Block Storage (EBS), and Amazon Redshift.



#### Audit Key Usage

AWS Key Management Service provides audit trail information directly to AWS CloudTrail. These audit trails help you meet compliance and regulatory requirements by providing logs of who used which key to access which data and when that access occurred.

Similar to SSE-S3, but there are some additional benefits and with the benefits come additional services. One of the benefits is an envelope key. The envelope key is a key that protects your data encryption key and provides an audit trail of when the key was used and who used it. Another benefit is to create and manage encryption keys yourself, or you can use a default key that is unique to you, the services which you are using and the region from where you are working. For more information, see [Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys \(SSE-KMS\)](#).

### [Use Server-Side Encryption with Customer-Provided Keys \(SSE-C\)](#)

While you manage the encryption key, Amazon S3 manages the encryption, writing data to the disk and decryption, when accessing the data. For more information, see [Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys \(SSE-C\)](#).

***In the website below there are steps shown for client and server-side encryption using AWS SDK In Ruby***

<https://www.concur.com/newsroom/article/using-data-encryption-in-aws>

Other Websites Used:

<http://searchaws.techtarget.com/answer/Whats-the-best-way-to-secure-Amazon-S3-buckets>

<http://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>