25°

Lab - AWS re/Start

Protección de datos mediante encriptación

SWIPE

Tarea 01

Protegiendo datos con AWS KMS

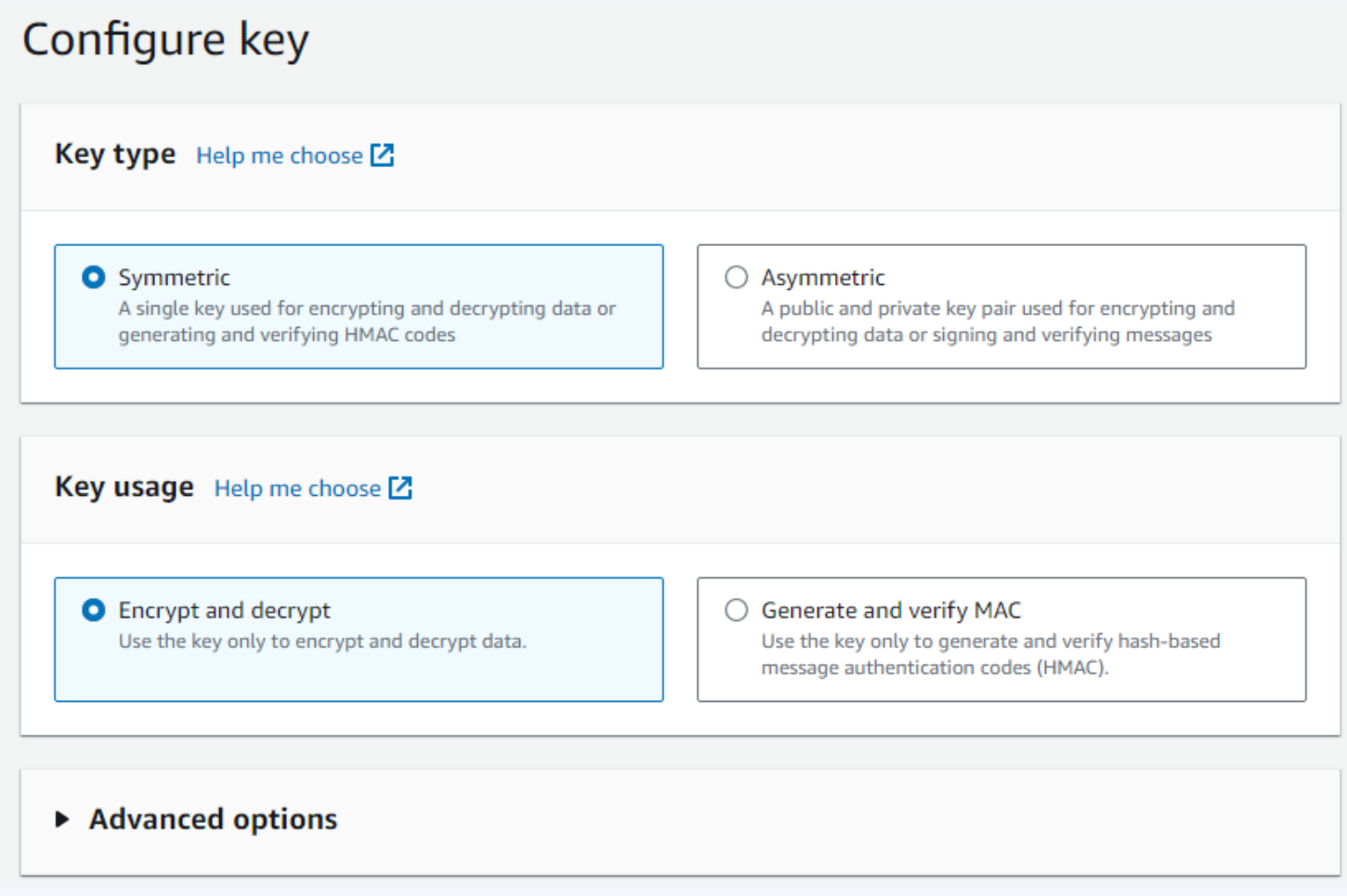
A continuación, se muestra los objetivos del laboratorio:

- Crear una clave de cifrado de AWS KMS
- Instalar la AWS Encryption CLI
- Cifrar texto plano
- Descifrar el texto cifrado

Nota. Es conveniente tener en cuenta que la función central de la criptografía es el cifrado, el cual transforma los datos en una forma ilegible. Esto garantiza la privacidad al mantener la información oculta a personas a las que no está destinada. El caso contrario es la desencriptación, que transforma los datos encriptados de nuevo en datos.

Tarea 01

Empezamos creando la clave en AWS KMS (Key Management Service), que usaremos para encriptar y desencriptar la datos. Con el servicio de AWS KMS podemos crear y administrar claves cifradas, y controlar su uso entre los servicios AWS y sus aplicaciones.

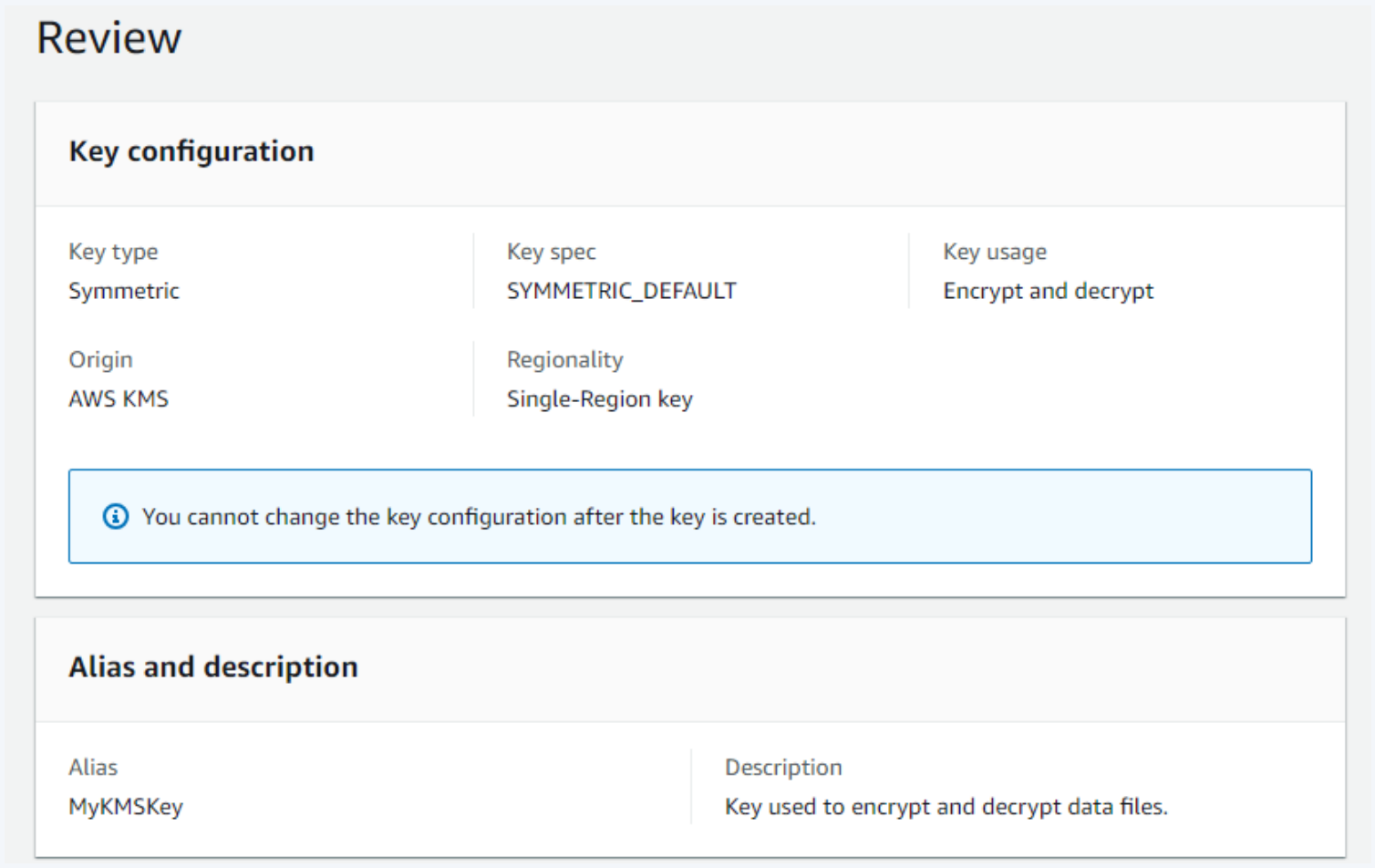


En este caso la clave a crear es simétrica, lo que significa que el cifrado usa la misma clave para encriptar y desencriptar los datos. Mientras que las de tipo asimétrico usan una clave pública para encriptar y una clave privada para desencriptar.

SWIPE

Tarea 01

Luego de añadir las etiquetas, definimos los permisos administrativos para la clave, en este caso los roles: *LabUser* y *AWSLabsUser-pZvTUrLxmfxgzPW8DuyY2J*. Y en los permisos de uso al rol *LabUser*.

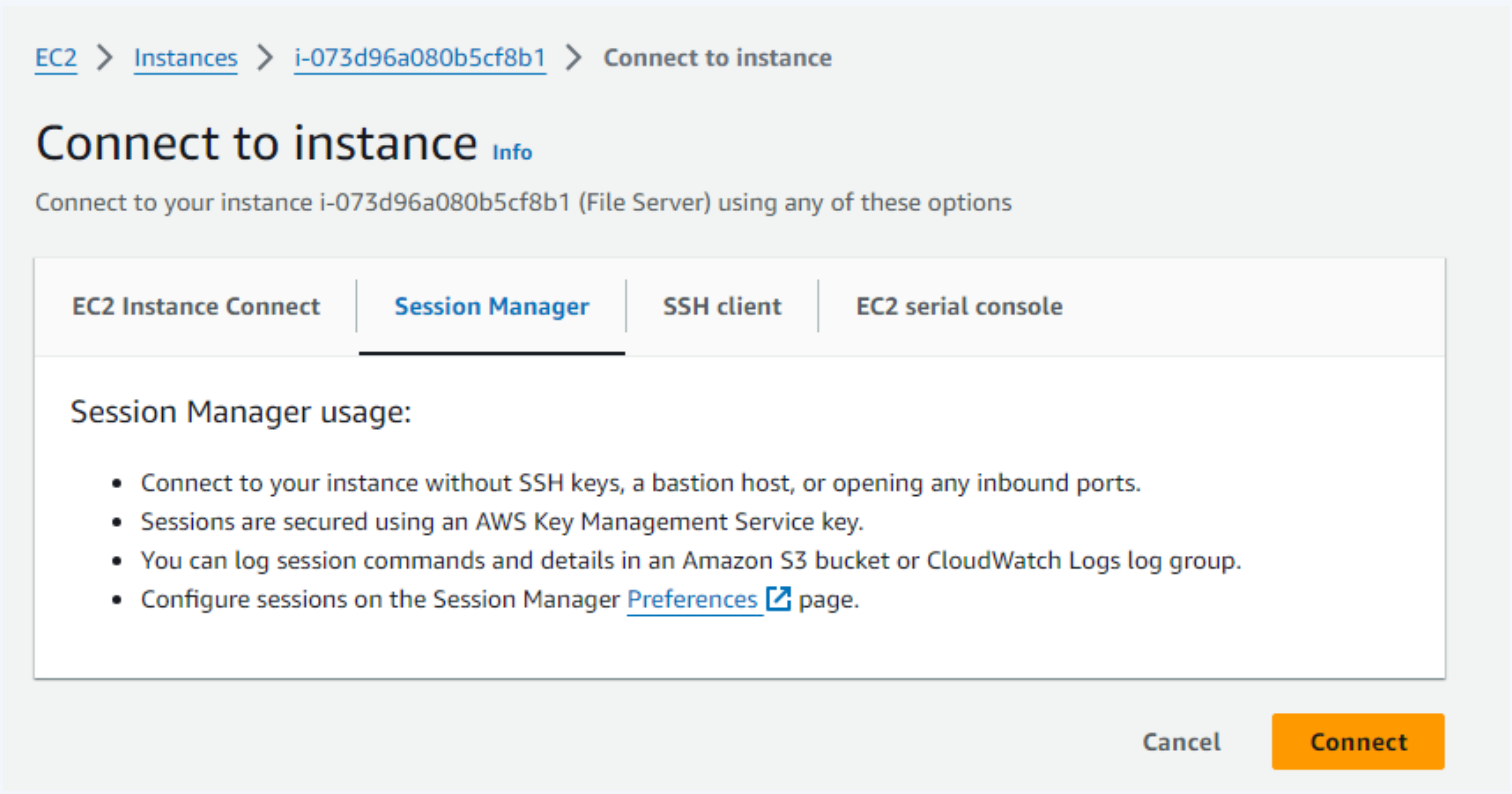


Es conveniente tener a la mano el nombre del recurso Amazon (ARN):
arn:aws:kms:us-west-2:555451576738:key/20ba30b0-3dbd-4dec-9b91-c8d8abc9ef24
Ahora, procedemos a configurar la instancia del servidor de archivos. Para ello debemos configurar las credenciales AWS de la instancia, y después de ello se debe instalar la AWS Encryption CLI.

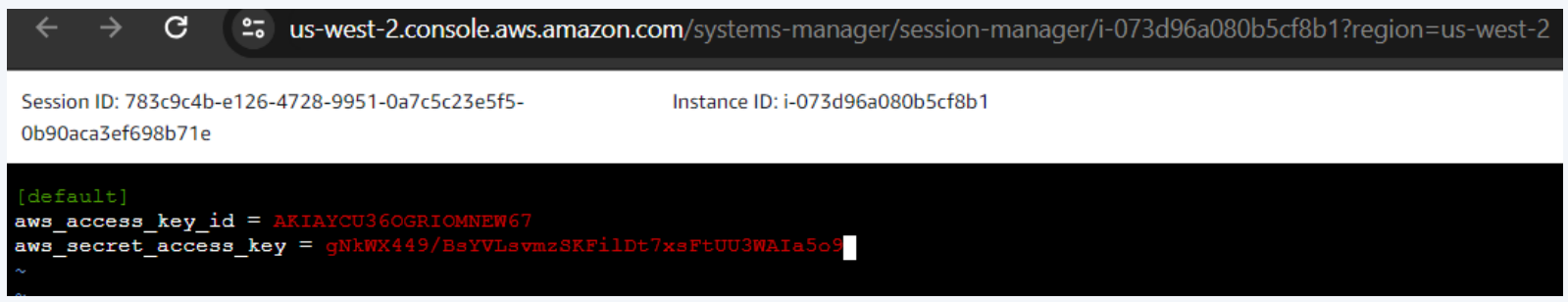
SWIPE

Tarea 01

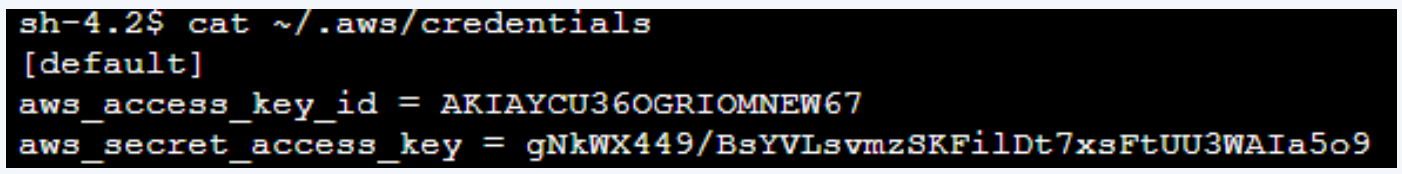
Nos conectamos a la instancia via *Session Manager*:



Y configuramos las credenciales AWS utilizando el comando *vi ~/.aws/credentials*



Y notamos que se ha actualizado el contenido en el archivo de las credenciales



SWIPE

Tarea 01

Ahora instalaremos la AWS Encryption CLI, utilizando los siguientes comandos:
pip3 install aws-encryption-sdk-cli
export PATH=$PATH:/home/ssm-user/.local/bin

Después de esto, vamos a encriptar y desencriptar los datos. Empezaremos creando un archivo con data sensible (privacidad)

```
sh-4.2$ touch secret1.txt secret2.txt secret3.txt
sh-4.2$ echo 'TOP SECRET 1!!!' >  secret1.txt
sh-4.2$
sh-4.2$ cat secret1.txt
TOP SECRET 1!!!
sh-4.2$
```

Luego, necesitamos una carpeta donde se almacenará el archivo cifrado, creamos dicho directorio: *mkdir output*
Y entonces, procedemos a encriptar los datos, para ello creamos la variable KeyArn y corremos el siguiente comando:

```
sh-4.2$ keyArn=arn:aws:kms:us-west-2:555451576738:key/20ba30b0-3dbd-4dec-9b91-c8d8abc9ef24
sh-4.2$ aws-encryption-cli --encrypt \
>                          --input secret1.txt \
>                          --wrapping-keys key=$keyArn \
>                          --metadata-output ~/metadata \
>                          --encryption-context purpose=test \
>                          --commitment-policy require-encrypt-require-decrypt \
>                          --output ~/output/.
```

Tarea 01

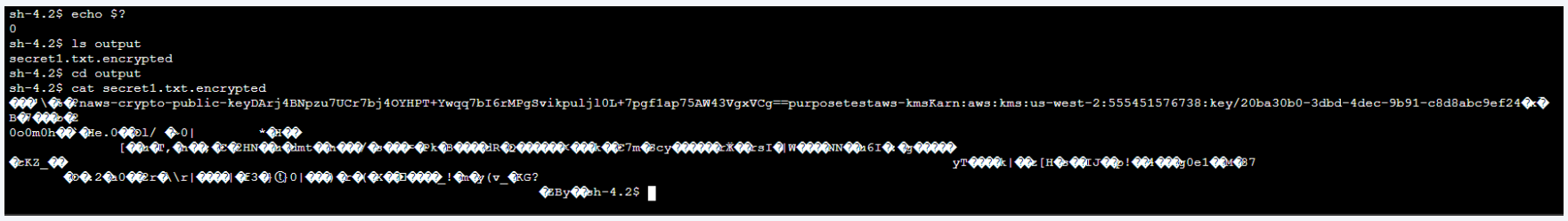
Y la explicación del comando utilizado es la siguiente:

- La primera línea cifra los contenidos del archivo. El comando usa el parámetro **–encrypt** para especificar la operación y el parámetro **–input** para indicar el archivo a cifrar.
- El parámetro **–wrapping-keys**, y su atributo requerido key, le indican al comando que use la clave de AWS KMS que está representada por el ARN de clave.
- El parámetro **–metadata-output** se usa para especificar un archivo de texto para los metadatos acerca de la operación de cifrado.
- Como práctica recomendada, el comando usa el parámetro **–encryption-context** para especificar un contexto de parámetro.
- El parámetro **–commitment-policy** se usa para especificar que la característica de seguridad de la confirmación de claves se debe usar para cifrar y descifrar.
- El valor del parámetro **–output**, ~/output/., indica al comando que escriba el archivo de destino en el directorio de destino.

SWIPE

Tarea 01

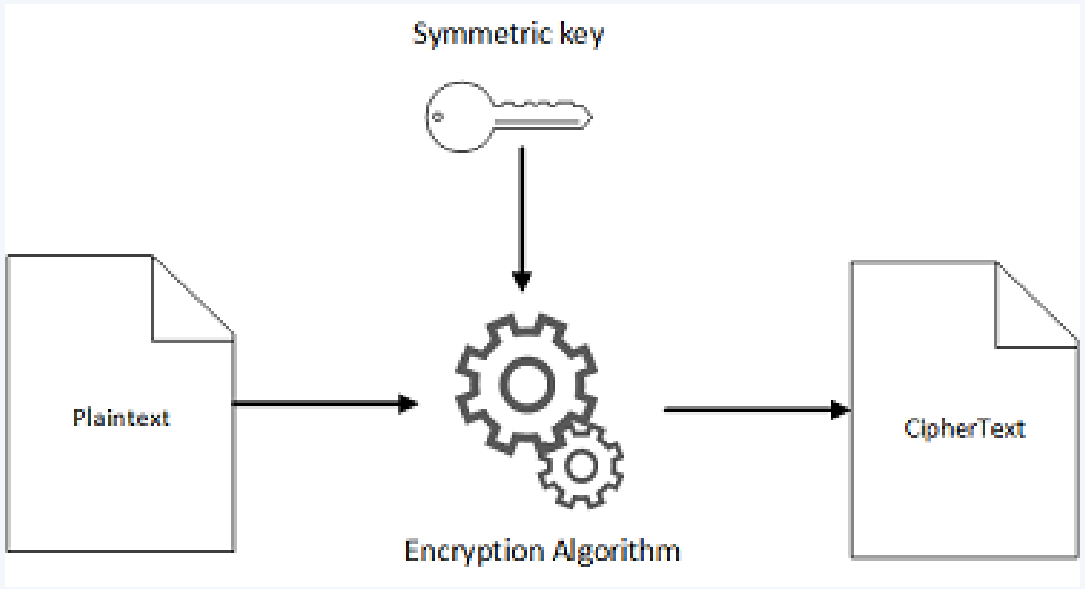
Para verificar que el cifrado ha sido exitoso, el resultado del comando **echo $?** debe ser 0. Así obtenemos lo siguiente:

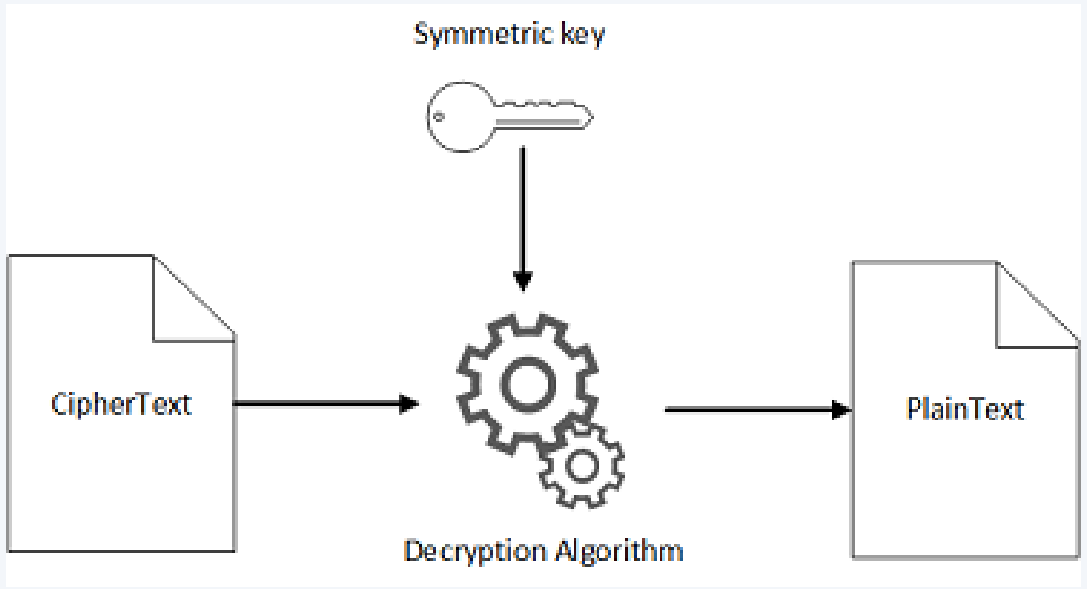


Ahora, haremos el proceso contrario que es decifrar el archivo secret1.txt.encrypted, mediante el siguiente comando

```
sh-4.2$ aws-encryption-cli --decrypt \
>                          --input secret1.txt.encrypted \
>                          --wrapping-keys key=$keyArn \
>                          --commitment-policy require-encrypt-require-decrypt \
>                          --encryption-context purpose=test \
>                          --metadata-output ~/metadata \
>                          --max-encrypted-data-keys 1 \
>                          --buffer \
>                          --output .
sh-4.2$ ls
secret1.txt.encrypted  secret1.txt.encrypted.decrypted
sh-4.2$ cat secret1.txt.encrypted.decrypted
TOP SECRET 1!!!
sh-4.2$
```

Notar que usamos el mismo KeyArn para cifrar y decifrar, debido a que estamos empleando una encriptación del tipo simétrica





SWIPE