**Lab: Symmetric Encryption Using AES**

**Overview:**

In this lab, you will learn how to use OpenSSL to encrypt files using the AES (Advanced Encryption Standard) symmetric encryption algorithm and decrypt them back to their original form. AES provides fast and secure encryption, widely used for protecting sensitive data.

**Learning Objectives:**

After completing this lab, you will be able to:

- Generate a secret key for AES encryption
- Encrypt files using AES encryption
- Decrypt files encrypted using AES

Gh

**Initializing the Lab Environment**

Click Terminal to Open New Terminal

Click New Terminal

Copy and paste the commands into the terminal for the remaining steps.

**Step 1: Create a Test File**

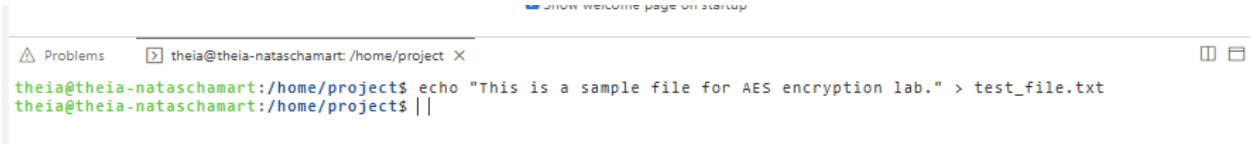
1. 1

1. echo "This is a sample file for AES encryption lab." > test_file.txt

Copied!Wrap Toggled!Executed!

**Command Description**

| Command/Option | Description |
|---|---|
| echo | Outputs a string of text. |
| > test_file.txt | Redirects the text into a file named test_file.txt. |




**Step 2: Generate a Secret Key for AES**

Generate a random 256-bit (32-byte) key for AES encryption:

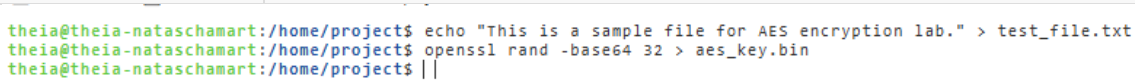
1. 1

1. openssl rand -base64 32 > aes_key.bin

Copied!Wrap Toggled!Executed!

**Command Description**

| Command/Option | Description |
|---|---|
| openssl rand | Generates random data. |
| -base64 | Encodes the random data in Base64 format, making it easier to store. |
| 32 | Specifies the number of bytes (32 bytes = 256 bits). |
| > aes_key.bin | Redirects the generated key into a file named aes_key.bin. |



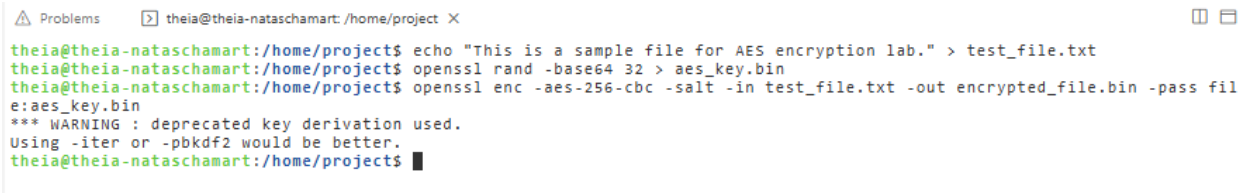

# Step 3: Encrypt the File Using AES

Encrypt the file using the generated AES key:

```
                                                                    1. 1
1. openssl enc -aes-256-cbc -salt -in test_file.txt -out encrypted_file.bin -pass
   file:aes_key.bin
Copied!Wrap Toggled!Executed!
```

## Command Description

| Command/Option | Description |
| --- | --- |
| openssl enc | Uses OpenSSL’s symmetric encryption utility. |
| -aes-256-cbc | Specifies the AES algorithm with 256-bit encryption in CBC mode. |
| -salt | Adds a random salt to increase encryption strength and prevent attacks. |
| -in test_file.txt | Specifies the input file to be encrypted. |
| -out encrypted_file.bin | Specifies the output file for the encrypted data. |
| -pass file:aes_key.bin | Uses the generated key from aes_key.bin for encryption. |




# Step 4: Verify the Encrypted File

Check the contents of the encrypted file to confirm that it is no longer in a readable format:

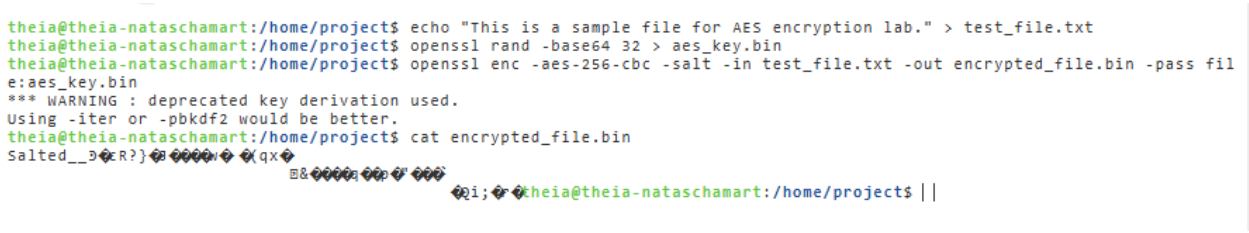
```
cat encrypted_file.bin
```

Copied!Wrap Toggled!Executed!

### Command Description

| Command/Option | Description |
| --- | --- |
| cat | Displays the contents of a file. |
| encrypted_file.bin | The file with the encrypted binary data. |

You should see unreadable binary data, confirming that the file has been encrypted.



# Step 5: Decrypt the Encrypted File

Decrypt the encrypted file back to its original content using the same key:

```
1. 1
1. openssl enc -d -aes-256-cbc -in encrypted_file.bin -out decrypted_file.txt -
   pass file:aes_key.bin
Copied!Wrap Toggled!Executed!
```

### Command Description

| Command/Option | Description |
|---|---|
| openssl enc -d | Specifies that the operation is decryption. |
| -aes-256-cbc | Uses the AES algorithm in the same mode as encryption. |
| -in encrypted_file.bin | Specifies the encrypted file to be decrypted. |
| -out decrypted_file.txt | Specifies the output file for the decrypted text. |
| -pass file:aes_key.bin | Uses the same key from aes_key.bin to decrypt the file. |

```
theia@theia-nataschamart:/home/project$ echo "This is a sample file for AES encryption lab." > test_file.txt
theia@theia-nataschamart:/home/project$ openssl rand -base64 32 > aes_key.bin
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in test_file.txt -out encrypted_file.bin -pass fil
e:aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat encrypted_file.bin
Salted__9R?}
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in encrypted_file.bin -out decrypted_file.txt -pass f
ile:aes_key.bin.bin -out decrypted_file.txt -pass fi
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$
```

# Step 6: Verify the Decrypted File

Finally, check the contents of the decrypted file to confirm that it matches the original:

```
1. 1
1. cat decrypted_file.txt
Copied!Wrap Toggled!Executed!
```

### Command Description

| Command/Option | Description |
| --- | --- |
| cat | Displays the contents of a file. |
| decrypted_file.txt | TThe file with the decrypted data. |

You should see the original text: “This is a sample file for AES encryption lab.”

```
theia@theia-nataschamart:/home/project$ echo "This is a sample file for AES encryption lab." > test_file.txt
theia@theia-nataschamart:/home/project$ openssl rand -base64 32 > aes_key.bin
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in test_file.txt -out encrypted_file.bin -pass fil
e:aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat encrypted_file.bin
Salted__Ɔ◆cR?}◆◆◆
```

### Step 3: Encrypt the File Using AES:

```
theia@theia-nataschamart:/home/project$ openssl rand -base64 32 > aes_key.bin
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in test_file.txt -out encrypted_file.bin -pass fil
e:aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat encrypted_file.bin
Salted__Ɔ◆R?}◆◆◆◆◆◆ ◆(qx◆
                         ⊡&◆◆◆◆◆◆◆◆◆ ◆◆◆
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in encrypted_file.bin -out decrypted_file.txt -pass f
ile:aes_key.bin.bin -out decrypted_file.txt -pass fi
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat decrypted_file.txt
This is a sample file for AES encryption lab.
theia@theia-nataschamart:/home/project$ echo "This is a secret message for Exercise 1." > secret_message.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in secret_message.txt -out secret_message.enc -pas
s file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ ||
```

### Step 4: Decrypt the Encrypted File:

```
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat encrypted_file.bin
Salted__Ɔ◆R?}◆◆◆◆◆◆ ◆(qx◆
                         ⊡&◆◆◆◆◆◆◆◆◆ ◆◆◆
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in encrypted_file.bin -out decrypted_file.txt -pass f
ile:aes_key.bin.bin -out decrypted_file.txt -pass fi
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat decrypted_file.txt
This is a sample file for AES encryption lab.
theia@theia-nataschamart:/home/project$ echo "This is a secret message for Exercise 1." > secret_message.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in secret_message.txt -out secret_message.enc -pas
s file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in secret_message.enc -out decrypted_message.txt -pas
s file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ ||
```

### Exercise 2: Encrypt and Decrypt a Credentials File Using AES

**Objective:** Encrypt a file containing sensitive credentials using AES and decrypt it.

### Task Details:

### Step 1: Create a Credentials File:

```
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ cat decrypted_file.txt
This is a sample file for AES encryption lab.
theia@theia-nataschamart:/home/project$ echo "This is a secret message for Exercise 1." > secret_message.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in secret_message.txt -out secret_message.enc -pass
 file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in secret_message.enc -out decrypted_message.txt -pass
 file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@thei
a-natascha
mart:/home
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$
```

## Step 2: Generate an AES Key and IV (Initialization Vector):

```
theia@theia-nataschamart:/home/project$ cat decrypted_file.txt
This is a sample file for AES encryption lab.
theia@theia-nataschamart:/home/project$ echo "This is a secret message for Exercise 1." > secret_message.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in secret_message.txt -out secret_message.enc -pass
 file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in secret_message.enc -out decrypted_message.txt -pass
 file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@thei
a-natascha
mart:/home
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
openssl rand -out aes_iv.bin 16
theia@theia-nataschamart:/home/project$ ||
```

```
mart:/home
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
openssl rand -out aes_iv.bin 16
theia@theia-nataschamart:/home/project$ IV=$(openssl rand -hex 16)
echo $IV > aes_iv_hex.txt
theia@theia-nataschamart:/home/project$ KEY=$(xxd -p aes_key.bin | tr -d '\n')
IV=$(xxd -p aes_iv.bin | tr -d '\n')
bash: xxd: command not found
bash: xxd: command not found
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -in credentials.txt -out credentials.enc -K $KEY -iv $IV
iv undefined
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in credentials.enc -out decrypted_credentials.txt -K $
KEY -iv $IV
iv undefined
theia@theia-nataschamart:/home/project$ cat credentials.txt
Username: admin
Password: SuperSecret123!
theia@theia-nataschamart:/home/project$
```

## Step 3: Encrypt the Credentials File Using AES:

```
:heia@theia-nataschamart:/home/project$ echo "This is a secret message for Exercise 1." > secret_message.txt
:heia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
:heia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -salt -in secret_message.txt -out secret_message.enc -pass
 file:./aes_key.bin
'** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
:heia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in secret_message.enc -out decrypted_message.txt -pass
 file:./aes_key.bin
'** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
:heia@thei
a-natascha
art:/home
:heia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
:heia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
:heia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
:heia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
penssl rand -out aes_iv.bin 16
:heia@theia-nataschamart:/home/project$ IV=$(openssl rand -hex 16)
echo $IV > aes_iv_hex.txt
:heia@theia-nataschamart:/home/project$ ||
```

```
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in secret_message.enc -out decrypted_message.txt -pass
 file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@thei
a-natascha
mart:/home
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
openssl rand -out aes_iv.bin 16
theia@theia-nataschamart:/home/project$ IV=$(openssl rand -hex 16)
echo $IV > aes_iv_hex.txt
theia@theia-nataschamart:/home/project$ KEY=$(xxd -p aes_key.bin | tr -d '\n')
IV=$(xxd -p aes_iv.bin | tr -d '\n')
bash: xxd: command not found
bash: xxd: command not found
theia@theia-nataschamart:/home/project$ ||
```

```
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in secret_message.enc -out decrypted_message.txt -pas
 file:./aes_key.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
theia@thei
a-natascha
mart:/home
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
openssl rand -out aes_iv.bin 16
theia@theia-nataschamart:/home/project$ IV=$(openssl rand -hex 16)
echo $IV > aes_iv_hex.txt
theia@theia-nataschamart:/home/project$ KEY=$(xxd -p aes_key.bin | tr -d '\n')
IV=$(xxd -p aes_iv.bin | tr -d '\n')
bash: xxd: command not found
bash: xxd: command not found
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -in credentials.txt -out credentials.enc -K $KEY -iv $IV
iv undefined
theia@theia-nataschamart:/home/project$ ||
```

## Step 4: Decrypt the Encrypted File:

```
Using -iter or -pbkdf2 would be better.
theia@thei
a-natascha
mart:/home
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
theia@theia-nataschamart:/home/project$ echo -e "Username: admin\nPassword: SuperSecret123!" > credentials.txt
theia@theia-nataschamart:/home/project$ openssl rand -out aes_key.bin 32
openssl rand -out aes_iv.bin 16
theia@theia-nataschamart:/home/project$ IV=$(openssl rand -hex 16)
echo $IV > aes_iv_hex.txt
theia@theia-nataschamart:/home/project$ KEY=$(xxd -p aes_key.bin | tr -d '\n')
IV=$(xxd -p aes_iv.bin | tr -d '\n')
bash: xxd: command not found
bash: xxd: command not found
theia@theia-nataschamart:/home/project$ openssl enc -aes-256-cbc -in credentials.txt -out credentials.enc -K $KEY -iv $IV
iv undefined
theia@theia-nataschamart:/home/project$ openssl enc -d -aes-256-cbc -in credentials.enc -out decrypted_credentials.txt -K $
KEY -iv $IV
iv undefined
theia@theia-nataschamart:/home/project$
```