

# Implementing Secure Data Exchange Using Cryptographic Techniques

## Algorithm: Secure Data Exchange System.

1. Initialize project environment on Kali Linux.
2. Generate cryptographic keys using symmetric and asymmetric algorithms.
3. Implement Public Key Infrastructure (PKI) using a Certificate Authority.
4. Encrypt and decrypt sensitive data using symmetric encryption.
5. Generate and verify digital signatures for integrity and authentication.
6. Establish secure communication using SSL/TLS protocol.
7. Perform security testing and validation.
8. Document results and demonstrate secure data exchange.

### 1. Symmetric Encryption (AES):

#### ◆ Purpose

Ensure confidentiality of data during transmission.

#### ◆ Commands:

```
echo "This is confidential data for secure exchange" > plaintext.txt  
openssl enc -aes-256-cbc -pbkdf2 -salt -in plaintext.txt -out encrypted.txt  
openssl enc -aes-256-cbc -pbkdf2 -d -in encrypted.txt -out decrypted.txt
```

#### ◆ Output

encrypted.txt → Encrypted data

decrypted.txt → Original data restored

### 2. Asymmetric Encryption & PKI:

#### ◆ Purpose:

Key management, authentication, trust establishment.

#### ◆ Commands (PKI Setup):

##### 1. CA Creation:

```
openssl genrsa -out ca.key 4096  
openssl req -x509 -new -nodes -key ca.key -sha256 -days 365 -out ca.crt
```

##### 2. Server Certificate:

```
openssl genrsa -out server.key 2048  
openssl req -new -key server.key -out server.csr  
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -out server.crt -days 365 -sha256
```

##### 3. Client Certificate:

```
openssl genrsa -out client.key 2048
```

```
openssl req -new -key client.key -out client.csr  
openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -out client.crt -days 365 -sha256
```

### **3. Digital Signature:**

◆ Purpose:

- Ensure integrity, authentication, non-repudiation.

◆ Commands:

```
echo "This message must be authenticated" > message.txt  
sha256sum message.txt > hash.txt  
openssl dgst -sha256 -sign client.key -out signature.sig message.txt  
openssl x509 -in client.crt -pubkey -noout > client_public.key  
openssl dgst -sha256 -verify client_public.key -signature signature.sig message.txt
```

◆ Output

Verified OK.

### **4. SSL / TLS Secure Communication:**

◆ Purpose:

- Secure data exchange over untrusted networks.

◆ Commands:

Server:

```
openssl s_server -cert server.crt -key server.key -CAfile ca.crt -accept 8443
```

Client:

```
openssl s_client -connect localhost:8443 -cert client.crt -key client.key -CAfile ca.crt
```

◆ Output:

Verify return code: 0 (ok).

### **4. Cryptographic Techniques Used:**

#### **4.1 Symmetric Encryption (AES):**

- AES-256-CBC algorithm was used to encrypt sensitive data. The same secret key was used for encryption and decryption to ensure confidentiality.

#### 4.2 Asymmetric Encryption (RSA):

- RSA was used for key generation and certificate management in PKI.

#### 5. PKI Setup and Configuration:

- A Public Key Infrastructure was implemented using OpenSSL. A Certificate Authority (CA) issued digital certificates to server and client. Certificates were used for authentication and trust establishment.

#### 6. Digital Signature Implementation:

- Digital signatures were generated using SHA-256 and RSA private keys. Signature verification ensured data integrity and non-repudiation.

#### 7. SSL/TLS Implementation:

- SSL/TLS protocol was implemented using OpenSSL s\_server and s\_client commands. Secure handshake verified certificates and established encrypted communication.

#### 8. Security Testing Report:

- Verified encrypted data confidentiality
- Verified digital signature integrity
- Verified SSL/TLS handshake authenticity
- No plaintext exposure observed

#### 9. Expected Outcomes:

- Secure data exchange system implemented.
- PKI-based authentication achieved.
- Improved understanding of cryptographic controls.

#### 10. Demonstration:

- Encrypted data exchange and SSL/TLS secure communication were successfully demonstrated using OpenSSL tools.

#### 11. Conclusion:

- The project successfully implemented secure data exchange using cryptographic techniques, ensuring confidentiality, integrity, and authentication over untrusted networks.

## Screenshots:

```
kali@kali: ~/Secure-Data-Exchange-Crypto/PKI
Session Actions Edit View Help
(kali㉿kali)-[~]
$ mkdir Secure-Data-Exchange-Crypto
(kali㉿kali)-[~]
$ ls
Desktop Downloads Pictures Secure-Data-Exchange-Crypto Videos
Documents Music Public Templates
(kali㉿kali)-[~]
$ cd Secure-Data-Exchange-Crypto
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ mkdir PKI Encryption Digital-Signature SSL-TLS Documentation Documentation/Screenshots
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ ls
Digital-Signature Documentation Encryption PKI SSL-TLS
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ cd PKI
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl genrsa -out ca.key 4096
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl req -x509 -new -nodes -key ca.key -sha256 -days 365 -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

```
Minimize all open windows and show the desktop
Session Actions Edit View Help
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ ls
Digital-Signature Documentation Encryption PKI SSL-TLS
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ cd PKI
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl genrsa -out ca.key 4096
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl req -x509 -new -nodes -key ca.key -sha256 -days 365 -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
_____
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Krishnagiri
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NextTech
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:.
Email Address []:.

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ █
```

```
kali@kali: ~/Secure-Data-Exchange-Crypto/PKI
Session Actions Edit View Help
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Krishnagiri
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NextTech
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:.
Email Address []:.

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl genrsa -out server.key 2048

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Krishnagiri
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:Vasanthan S
Email Address []:vasanthsvs1234@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Vasantha1510
An optional company name []:.
```

```
kali@kali: ~/Secure-Data-Exchange-Crypto/PKI
Session Actions Edit View Help
└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Krishnagiri
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:Vasanthan S
Email Address []:vasanthsvs1234@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Vasantha1510
An optional company name []:.

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 365 -sha256

Certificate request self-signature ok
subject=C=IN, ST=TamilNadu, L=Krishnagiri, CN=Vasanthan S, emailAddress=vasanthsvs1234@gmail.com

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$
```

```
kali@kali: ~/Secure-Data-Exchange-Crypto/PKI
Session Actions Edit View Help

Certificate request self-signature ok
subject=C=IN, ST=TamilNadu, L=Krishnagiri, CN=Vasanthan S, emailAddress=vasanthsvs1234@gmail.com

[(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]]
$ openssl genrsa -out client.key 2048

[(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]]
$ openssl req -new -key client.key -out client.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Krishnagiri
Organization Name (eg, company) [Internet Widgits Pty Ltd]::
Organizational Unit Name (eg, section) []::
Common Name (e.g. server FQDN or YOUR name) []:Vasantha
Email Address []:vasanthsvs1234@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Vasantha@1510
An optional company name []:

[(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]]
$
```

```
kali@kali: ~/Secure-Data-Exchange-Crypto/PKI
Session Actions Edit View Help

[(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]]
$ openssl req -new -key client.key -out client.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Krishnagiri
Organization Name (eg, company) [Internet Widgits Pty Ltd]::
Organizational Unit Name (eg, section) []::
Common Name (e.g. server FQDN or YOUR name) []:Vasantha
Email Address []:vasanthsvs1234@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Vasantha@1510
An optional company name []:

[(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]]
$ openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out client.crt -days 365 -sha256

Certificate request self-signature ok
subject=C=IN, ST=TamilNadu, L=Krishnagiri, CN=Vasantha, emailAddress=vasanthsvs1234@gmail.com

[(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]]
$
```

```
kali | _ 1 2 3 4 | 22:36 | _ X
Session Actions Edit View Help
Certificate request self-signature ok
subject=C-IN, ST=TamilNadu, L=Krishnagiri, CN=Vasanths, emailAddress=vasanthsvs1234@gmail.com
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/PKI]
$ cd .. /Encryption

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ echo "This is confidential data for secure exchange" > plaintext.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ openssl enc -aes-256-cbc -salt -in plaintext.txt -out encrypted.txt

enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ openssl enc -aes-256-cbc -d -in encrypted.txt -out decrypted.txt

enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
bad decrypt
4047CAC2487F0000:error:1C800064:Provider routines:ossl_cipher_unpadblock:bad decrypt:../providers/implementations/ciphers/ciphercommon_block.c:107:

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$
```

```
kali | _ 1 2 3 4 | 22:50 | _ X
Session Actions Edit View Help
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ cd .. /Encryption

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ echo "This is confidential data for secure exchange" > plaintext.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ cat plaintext.txt
This is confidential data for secure exchange

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ openssl enc -aes-256-cbc -salt -in plaintext.txt -out encrypted.txt

enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ openssl enc -aes-256-cbc -d -in encrypted.txt -out decrypted.txt

enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ openssl enc -aes-256-cbc -d -in encrypted.txt -out decrypted.txt
```

```
kali | [ ] 1 2 3 4 | 23:18 | 🔒 G
kali@kali: ~/Secure-Data-Exchange-Crypto/Digital-Signature
Session Actions Edit View Help
└─(kali㉿kali)-[~]
$ cd Secure-Data-Exchange-Crypto
└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ ls
Digital-Signature Documentation Encryption PKI SSL-TLS
└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto]
$ cd Digital-Signature
└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ echo "This message must be authenticated" > message.txt

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ cat message.txt
This message must be authenticated

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ sha256sum message.txt > hash.txt

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ openssl dgst -sha256 -sign .. /PKI/client.key -out signature.sig message.txt

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ openssl dgst -sha256 -verify <(openssl x509 -in .. /PKI/client.crt -pubkey -noout) -signature signature.sig message.txt
Verified OK
```

```
kali | [ ] 1 2 3 4 | 23:18 | 🔒 G
kali@kali: ~/Secure-Data-Exchange-Crypto/Digital-Signature
Session Actions Edit View Help
└─(kali㉿kali)-[~]
$ cd Digital-Signature
└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ echo "This message must be authenticated" > message.txt

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ cat message.txt
This message must be authenticated

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ sha256sum message.txt > hash.txt

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ openssl dgst -sha256 -sign .. /PKI/client.key -out signature.sig message.txt

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ openssl dgst -sha256 -verify <(openssl x509 -in .. /PKI/client.crt -pubkey -noout) -signature signature.sig message.txt
Verified OK

└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ ls
hash.txt message.txt signature.sig
└─(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Digital-Signature]
$ █
```

```
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ cd .. /SSL-TLS

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ nano tls-server-command.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$
```

```
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ cd .. /SSL-TLS

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ nano tls-server-command.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ nano tls-server-command.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ cat tls-server-command.txt
openssl s_server -cert .. /PKI/server.crt -key .. /PKI/server.key -CAfile .. /PKI/ca.crt -accept
8443

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ nano tls-client-command.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ cat tls-client-command.txt
openssl s_client -connect localhost:8443 -cert .. /PKI/client.crt -key .. /PKI/client.key -CAfi
le .. /PKI/ca.crt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$ ls
tls-client-command.txt  tls-server-command.txt

(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/SSL-TLS]
$
```

A screenshot of a terminal window titled "kali@kali: ~/Secure-Data-Exchange-Crypto/Encryption". The window has four tabs at the top, with the first tab labeled "1" and the second tab labeled "2" being visible. The terminal menu bar includes "Session", "Actions", "Edit", "View", and "Help". The main pane displays the following text:

```
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$ cat decrypted.txt
This is confidential data for secure exchange
(kali㉿kali)-[~/Secure-Data-Exchange-Crypto/Encryption]
$
```