REPORT

Introduction:

Using the well-known open-source cryptography library OpenSSL, the supplied code provides a method for encrypting and decrypting text. AES is used for secure message transmission in this system, while RSA is used for key exchange in an asymmetric encryption system. Clarifying the primary elements and features of the system is the aim of this report.

2. Components:

2.1. Encryption (project2_encryption.cpp):

Main Function: The primary task coordinates the encryption procedure.
Validates input to make sure the right amount of command-line parameters are supplied.
RSA Encryption: Using the supplied public key, a symmetric key is encrypted using OpenSSL's rsautl command.
Using AES-256-CBC symmetric encryption, the message is encrypted using OpenSSL's enc command. From the previously encrypted file, the symmetric key is read.
Creates a digital signature by employing the private key to sign the encrypted communication.

2.2. Decryption (project2_decryption.cpp):

Main Purpose: Manages the decryption procedure.
Input validation verifies that the right number of arguments are supplied.
Verification of the Digital Signature: Using the public key, confirm the encrypted message's digital signature.
Symmetric Decryption: This method uses the symmetric key that is extracted from the decrypted file along with AES-256-CBC to decrypt the encrypted message.
-The project2_encryption.cpp and project2_decryption.cpp are compiled using this syntax.

3. Workflow:

3.1. Encryption Workflow:

Make a key that is symmetric.
Using the public key and asymmetric encryption (RSA), encrypt the symmetric key.
Using the created symmetric key, encrypt the message using symmetric encryption (AES-256-CBC).

Using the private key, create a digital signature for the encrypted message.

3.2. Decryption Workflow:

Use the public key to validate the encrypted message's digital signature.
With the private key, decrypt the symmetric key.
Utilizing the decrypted symmetric key, decrypt the message.

4. Dependencies:

OpenSSL Library: To perform cryptographic functions including RSA and AES encryption and decryption, as well as the creation and validation of digital signatures, the system depends on OpenSSL.
C++ Standard Library: The standard C++ libraries are utilized for system command execution, file input/output, and string manipulation.

5. Command Line Usage:

Encryption: ./project2_encryption <encrypted_message_file> <public.pem> <privatekey.pem>
Decryption: ./project2_decryption <signedcipher> <public.pem> <symmetric.txt>

6. Security Considerations:

Key Security: The safeguarding of the private key is crucial to the system's security.
Secure Transmission: The message is transmitted securely thanks to AES symmetric encryption.
Verifies the integrity and authenticity of the encrypted message with a digital signature.

7. Conclusion:

The solution that is offered provides a strong method of sending messages in a safe manner by utilizing both symmetric and asymmetric encryption techniques. To preserve the system's security posture, it is imperative to guarantee the safe transfer and storage of keys as well as to adhere to recommended practices for cryptographic activities.

g++ -o project2_encryption project2_encryption.cpp -I/opt/homebrew/opt/openssl@3/include -L/opt/homebrew/opt/openssl@3/lib -lssl -lcrypto
g++ -o project2_decryption project2_decryption.cpp -I/opt/homebrew/opt/openssl@3/include -L/opt/homebrew/opt/openssl@3/lib -lssl -lcrypto

Then after a Successful compile, we execute project2_encryption.cpp using ./project2_encryption symm_key.bin pubkey.pem privatekey.pem and it asks for password.
The password for the encryption is Vivek.

```
vivekreddysuram@Viveks-MacBook-Pro prog assing 3 % g++ -o project2_encryption project2_encryption.cpp -I/opt/homebrew/opt/openssl@3/include -L/opt/homebrew/opt/openssl@3/lib -lssl -lcrypto
vivekreddysuram@Viveks-MacBook-Pro prog assing 3 % g++ -o project2_decryption project2_decryption.cpp -I/opt/homebrew/opt/openssl@3/include -L/opt/homebrew/opt/openssl@3/lib -lssl -lcrypto
vivekreddysuram@Viveks-MacBook-Pro prog assing 3 % ./project2_encryption symm_key.bin pubkey.pem privatekey.pem
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Enter pass phrase for privatekey.pem:
```

After entering the password successfully we can execute project2_decryption.cpp using ./project2_decryption signedcipher public.pem symmetric.txt

```
vivekreddysuram@Viveks-MacBook-Pro prog assing 3 % ./project2_decryption signedcipher public.pem symmetric.txt

Verified OK
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

Once the Verified OK is displayed our code worked successfully.