**NAME: CHUKWU DANIEL NONSO**

**MATRIC NO.: BHU/20/04/05/0010**

**CMP 408 ASSIGNMENT 2**

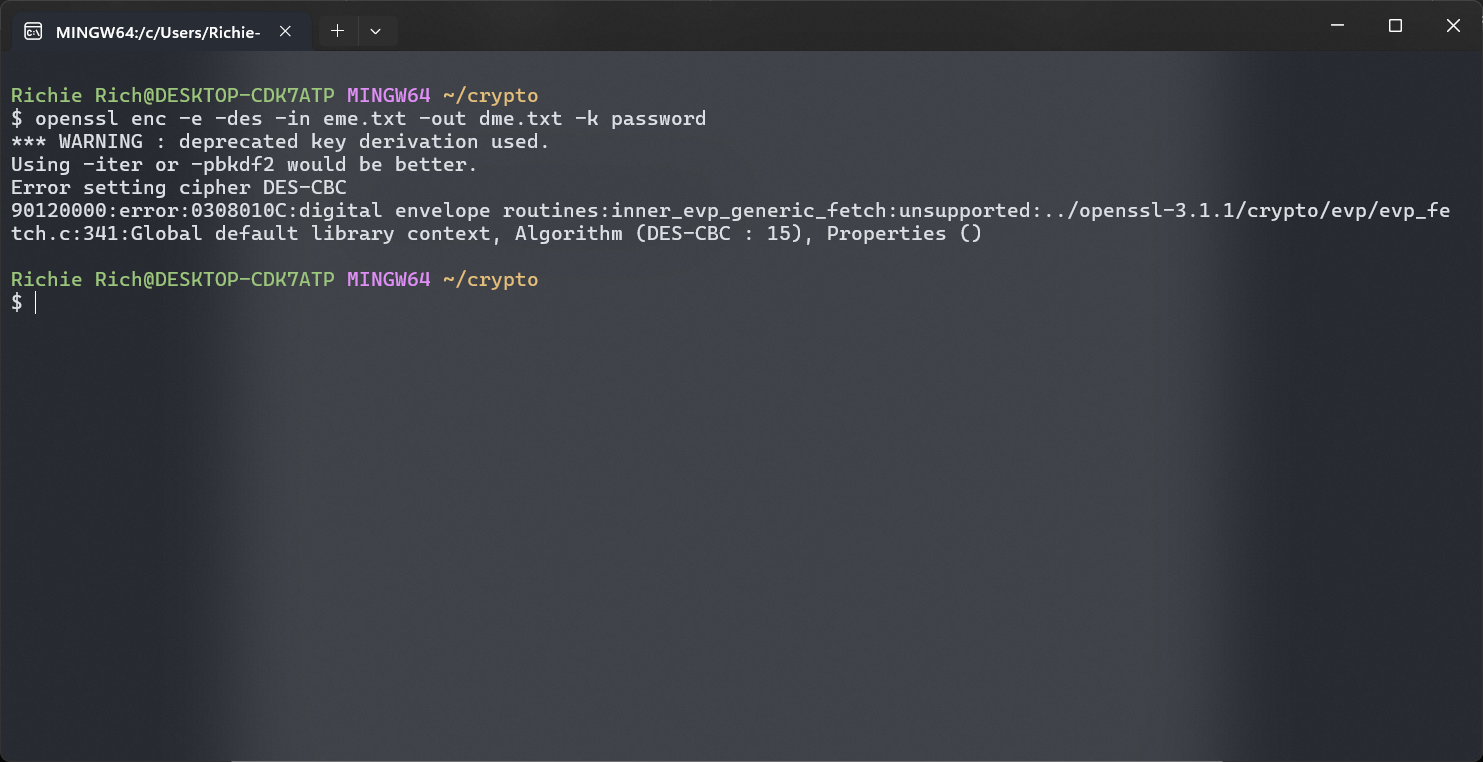
# PART 1: Use OpenSSL for Symmetric Encryption

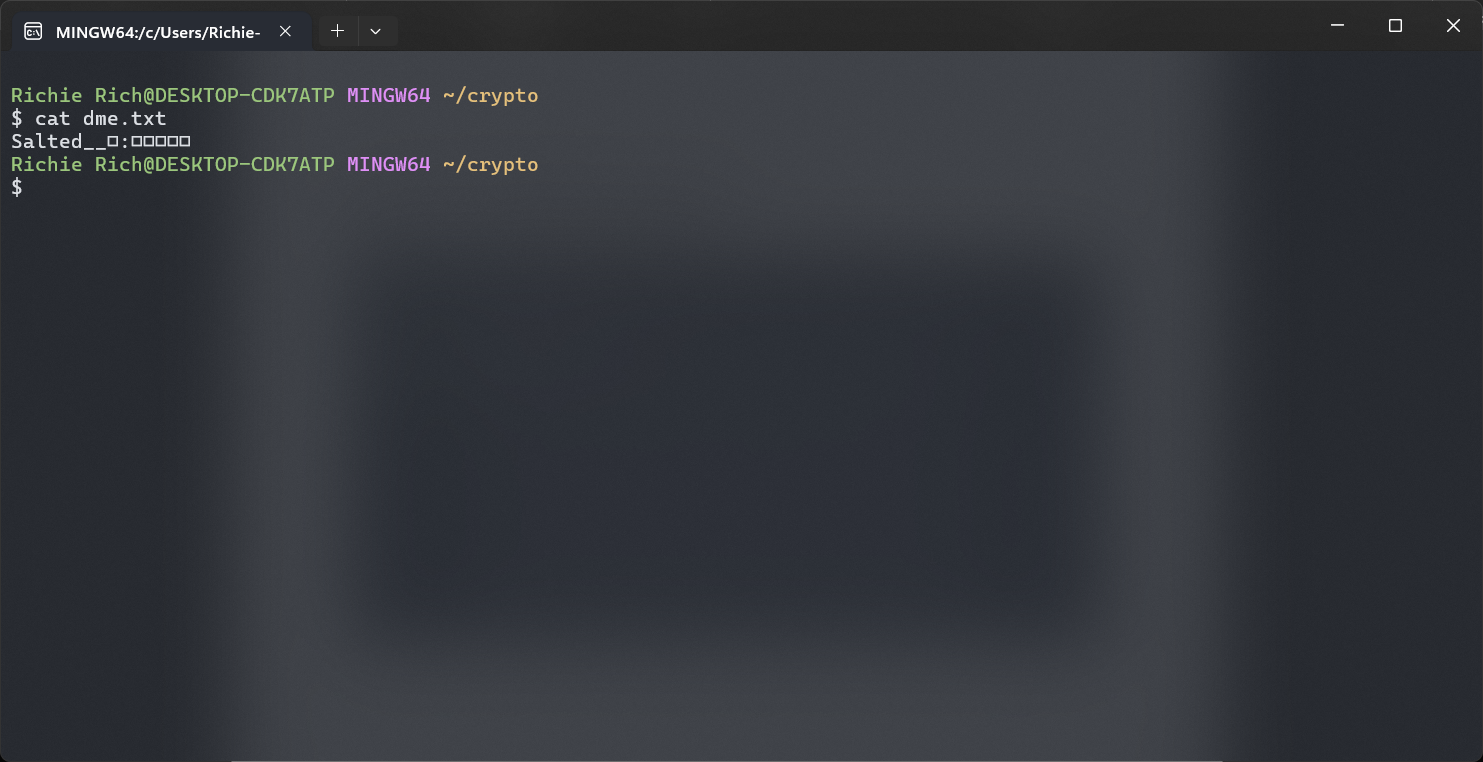
d. What does the ***enc*** command do?

The openssl ***enc*** command is used for symmetric encryption and decryption of data. It allows users to choose from various cipher algorithms, derive keys from passwords, handle input/output from files or streams, and apply options like base64 encoding or salting. This command is primarily used for encrypting files or data streams using specified algorithms, with options to control the encryption process, key derivation, and output format.

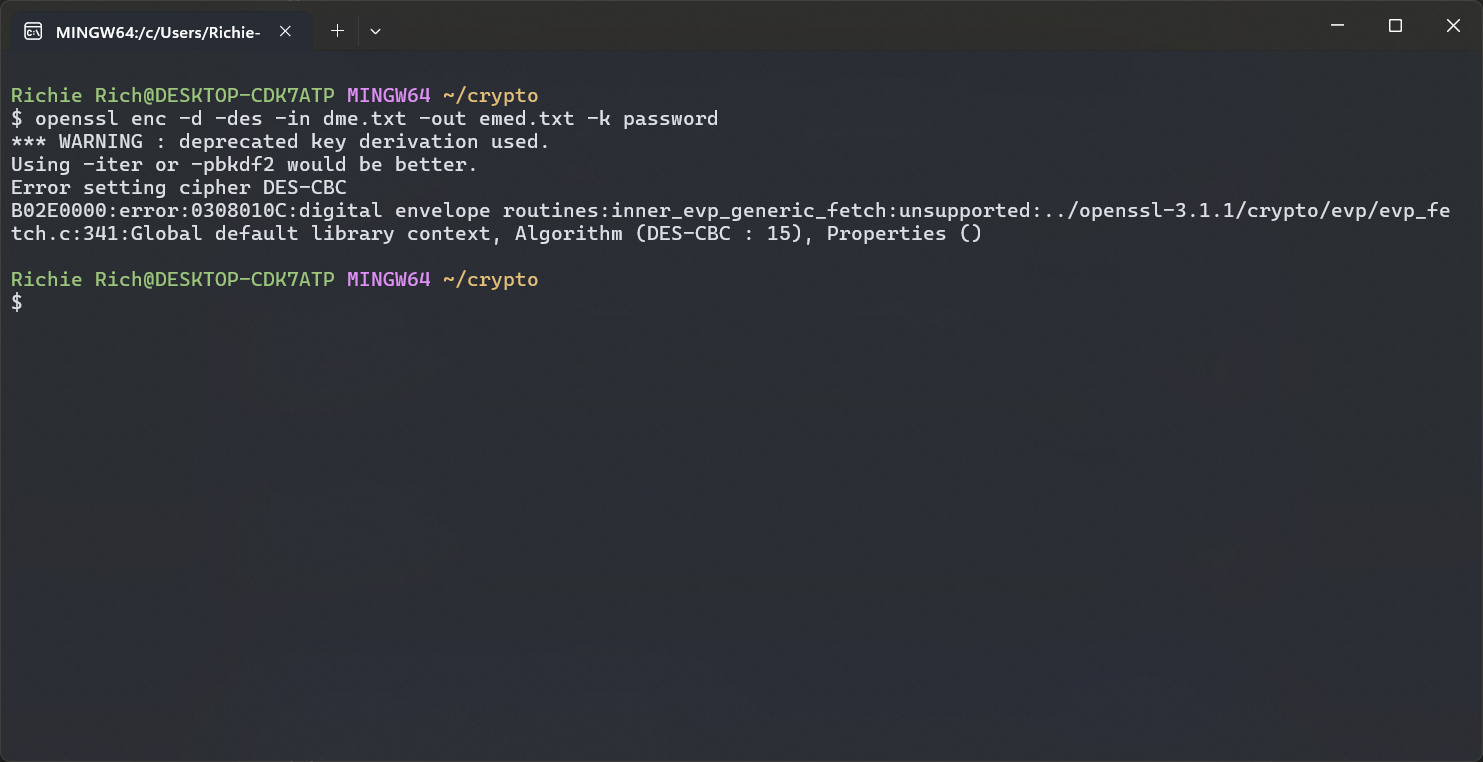
# PART 2: Use OpenSSL for Symmetric Encryption

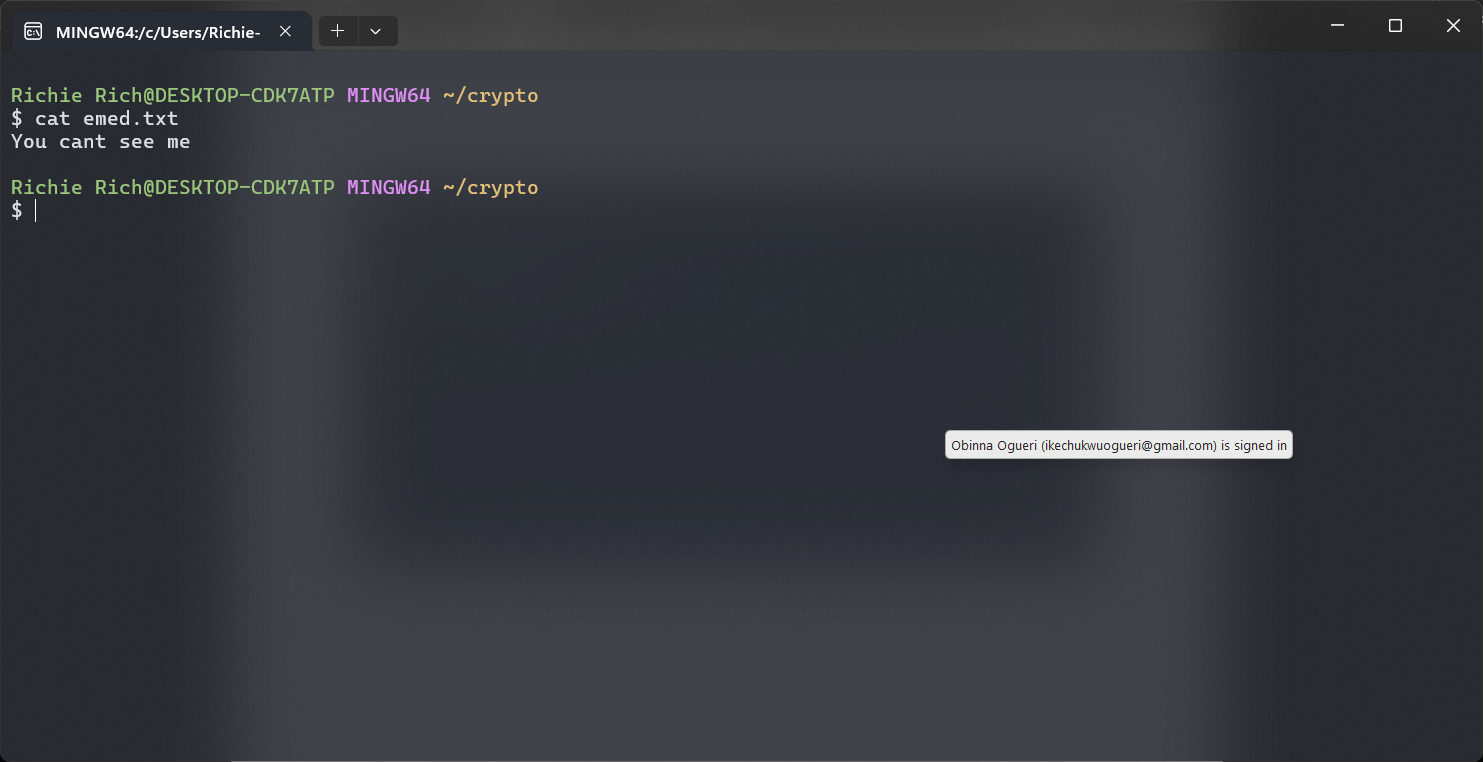
1. Encrypt “eme.txt” using DES to produce “dme.txt”; the key is “password”.

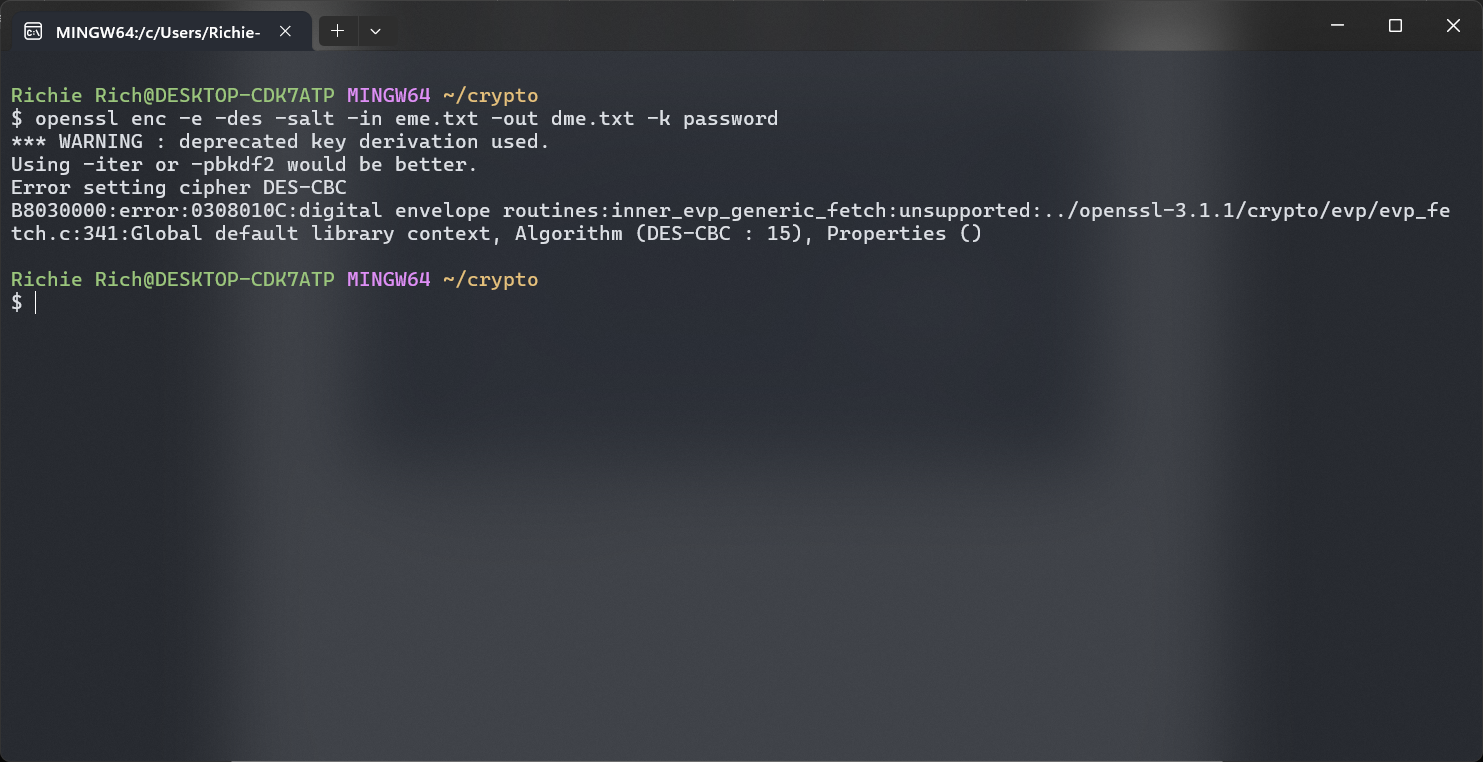
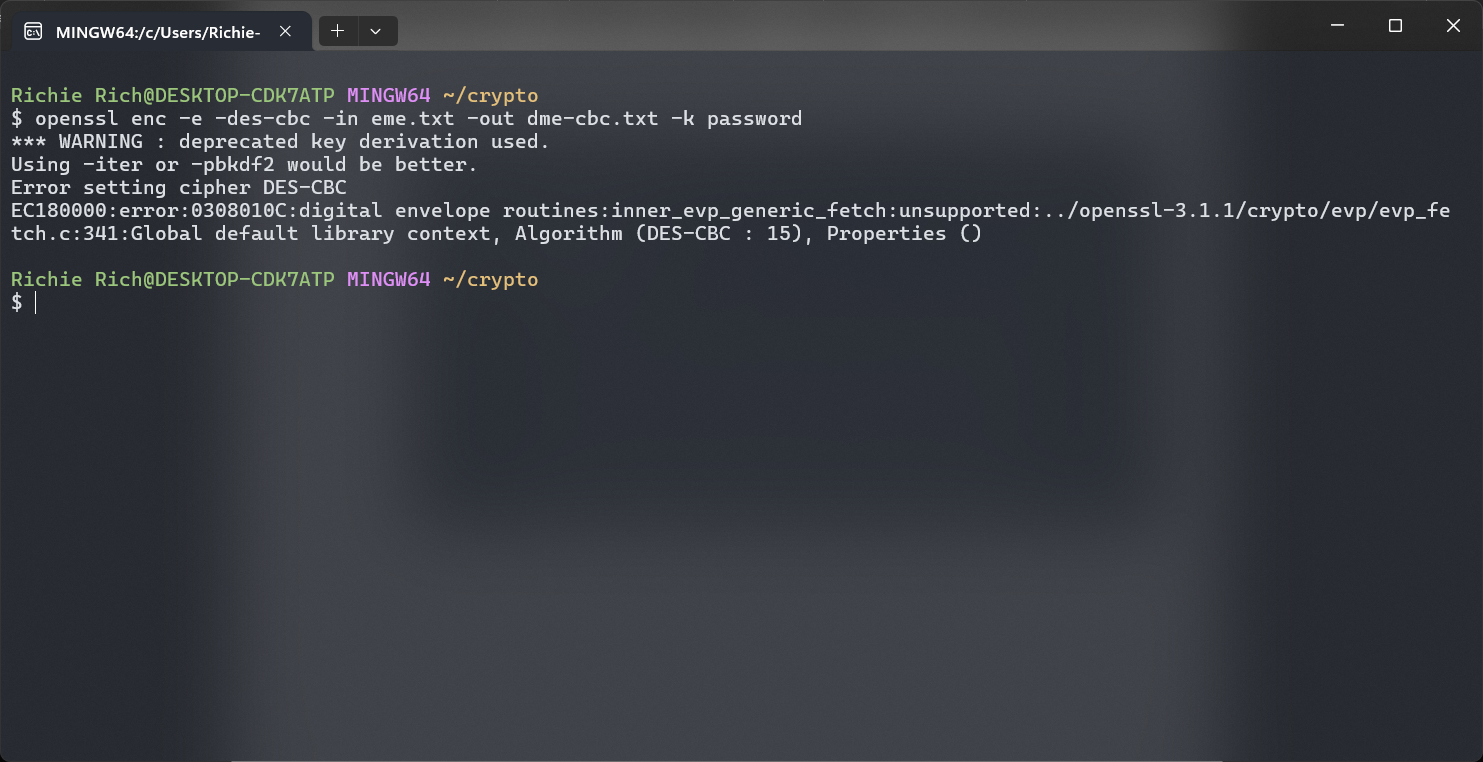
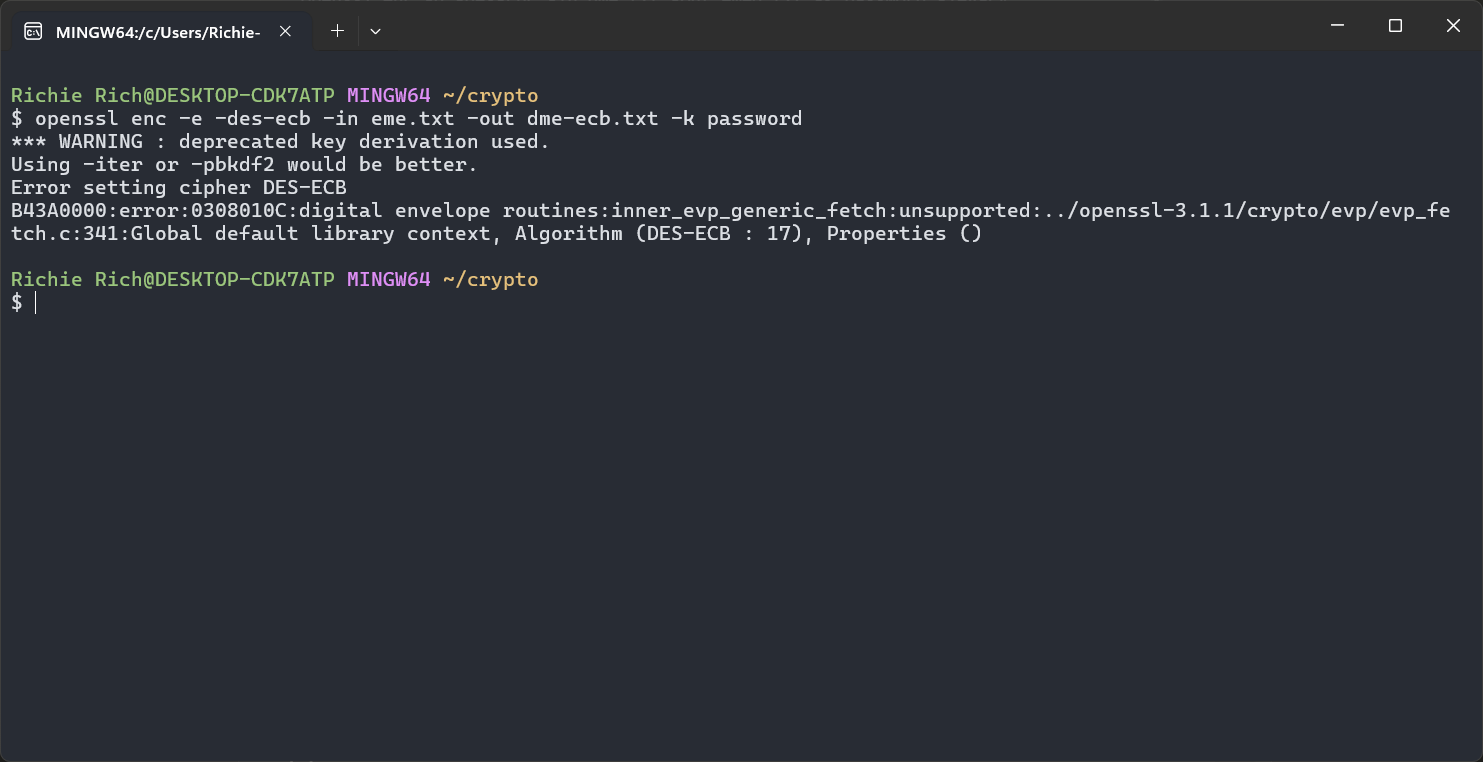
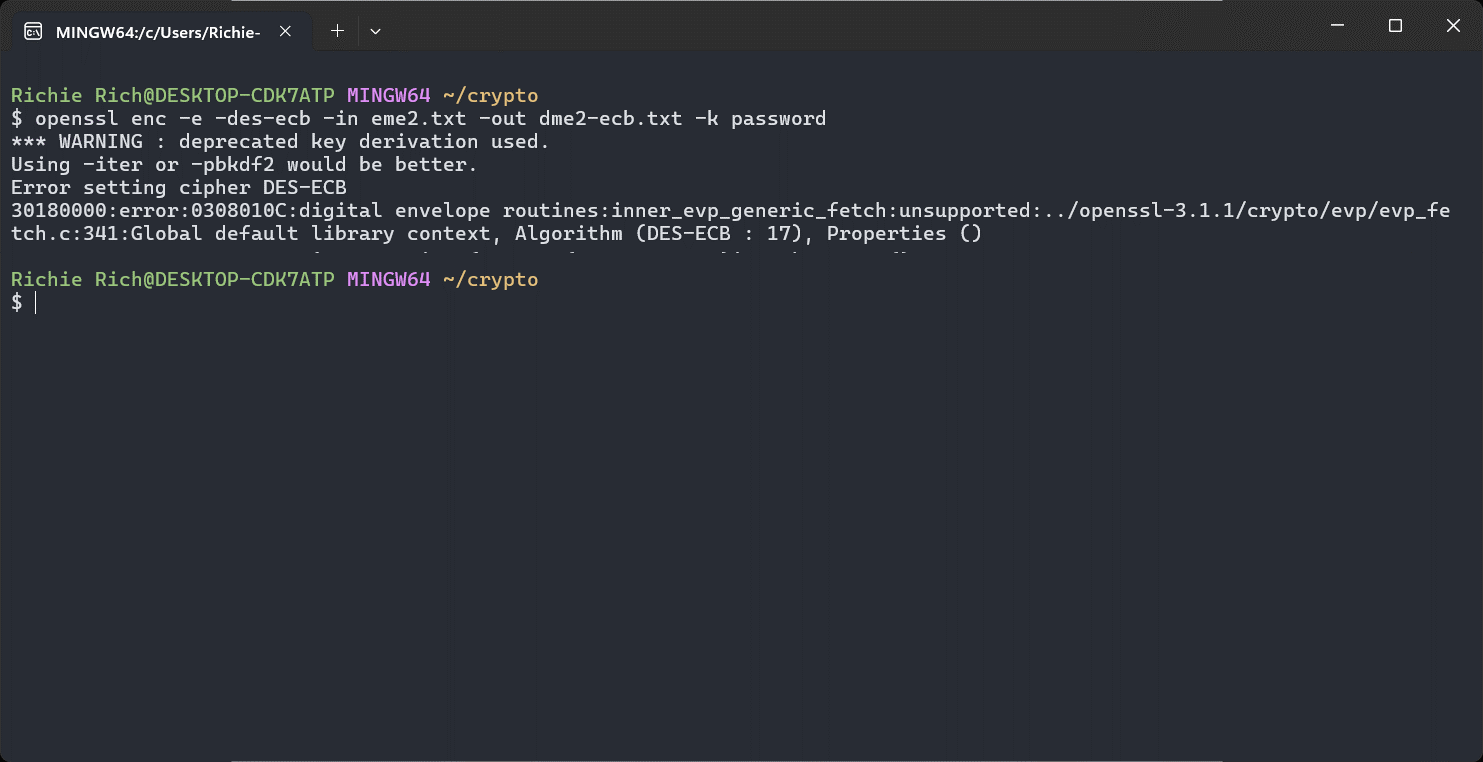
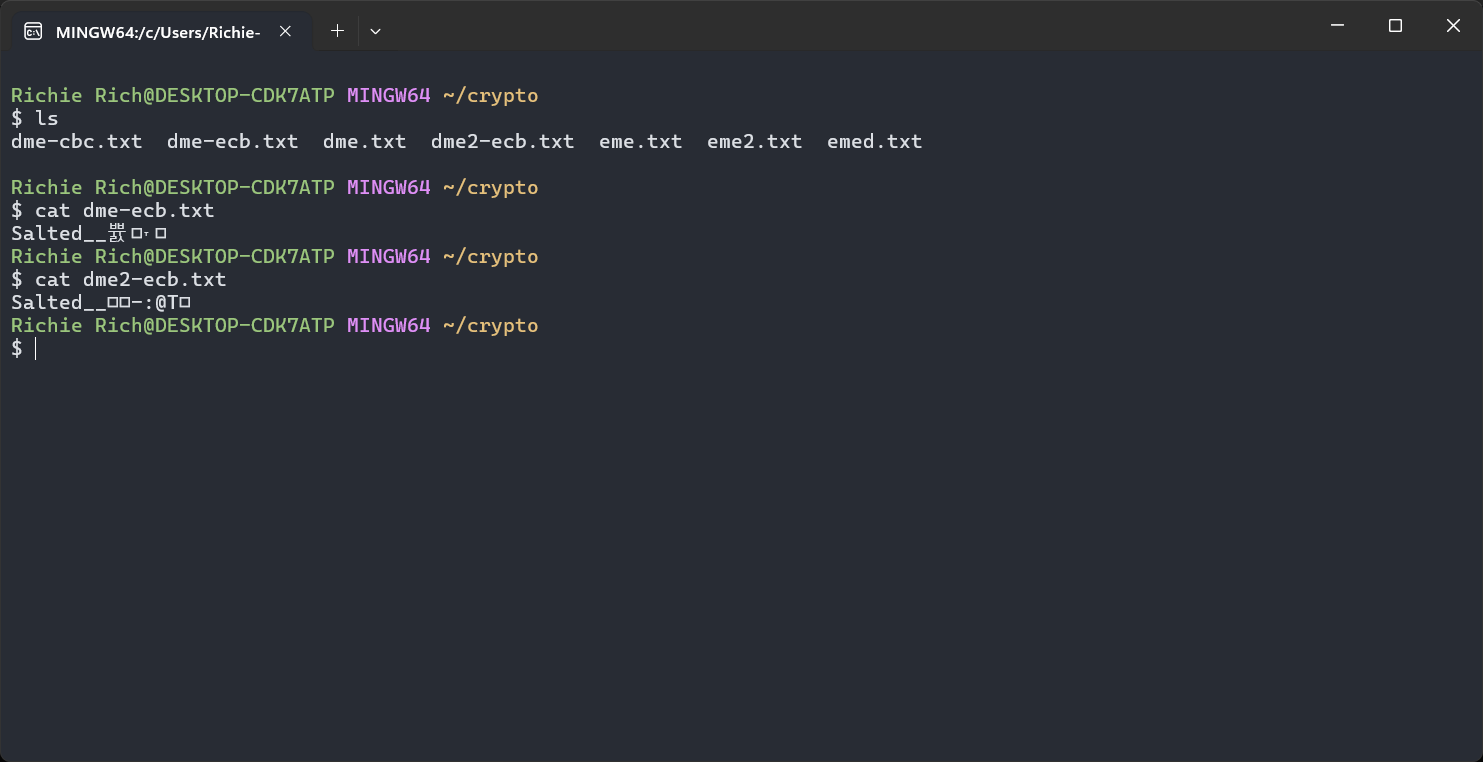
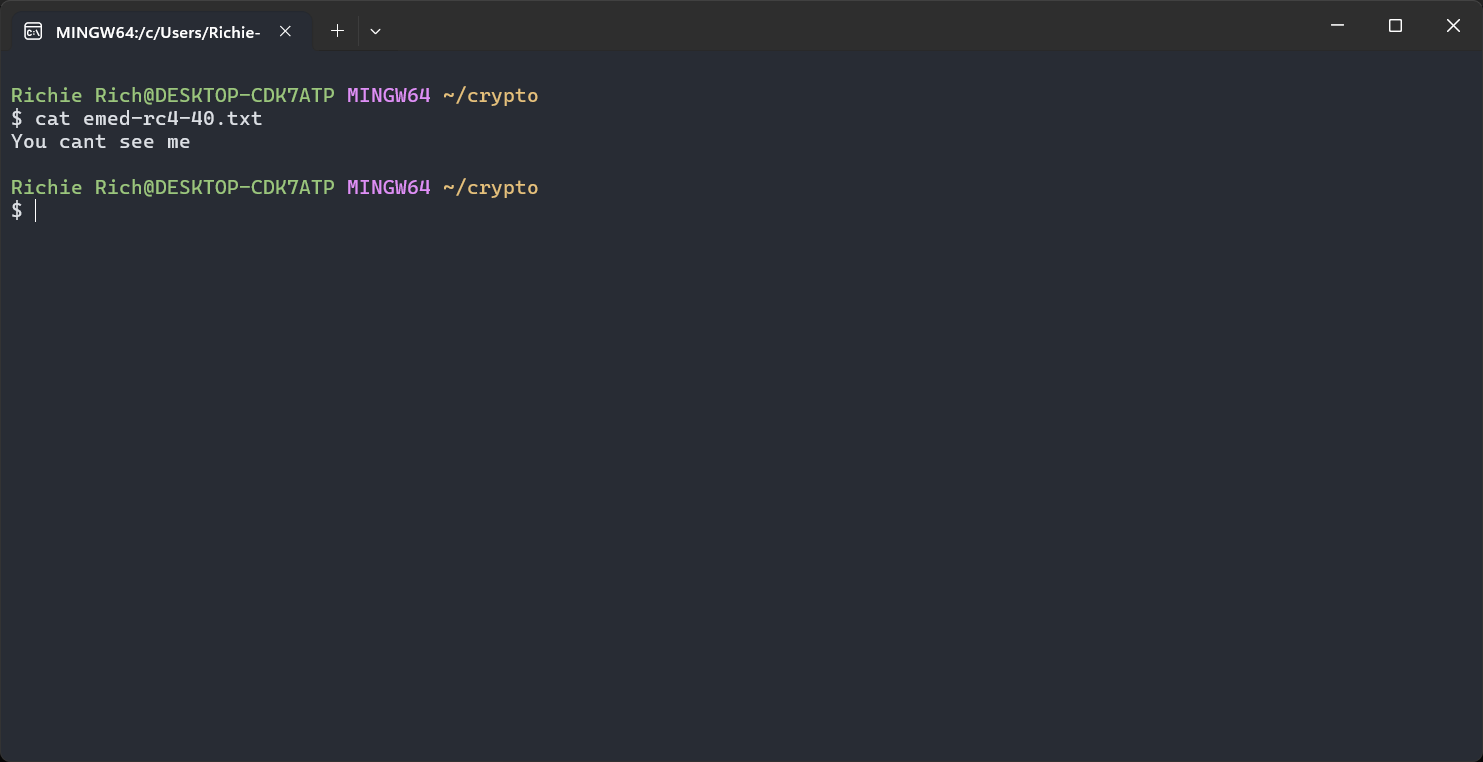
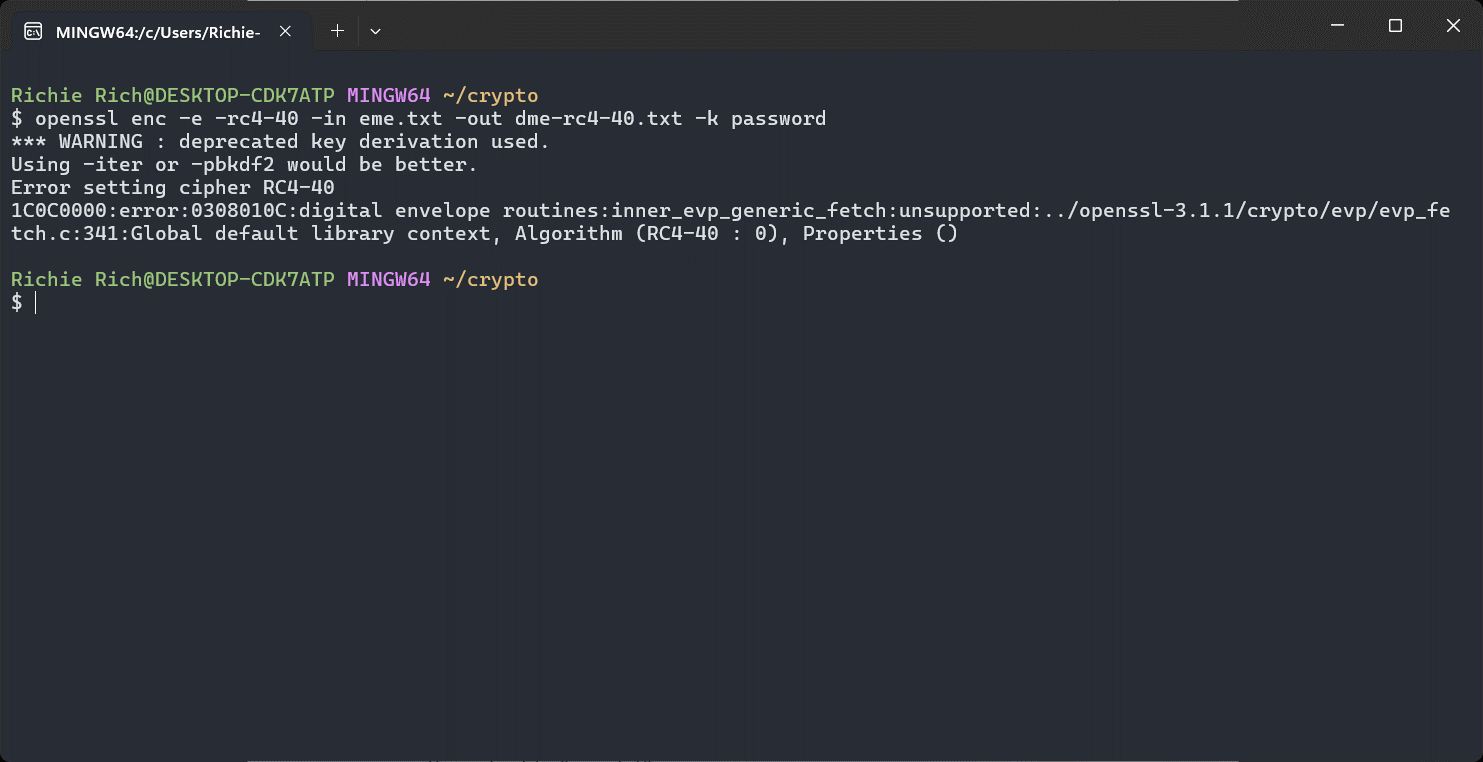
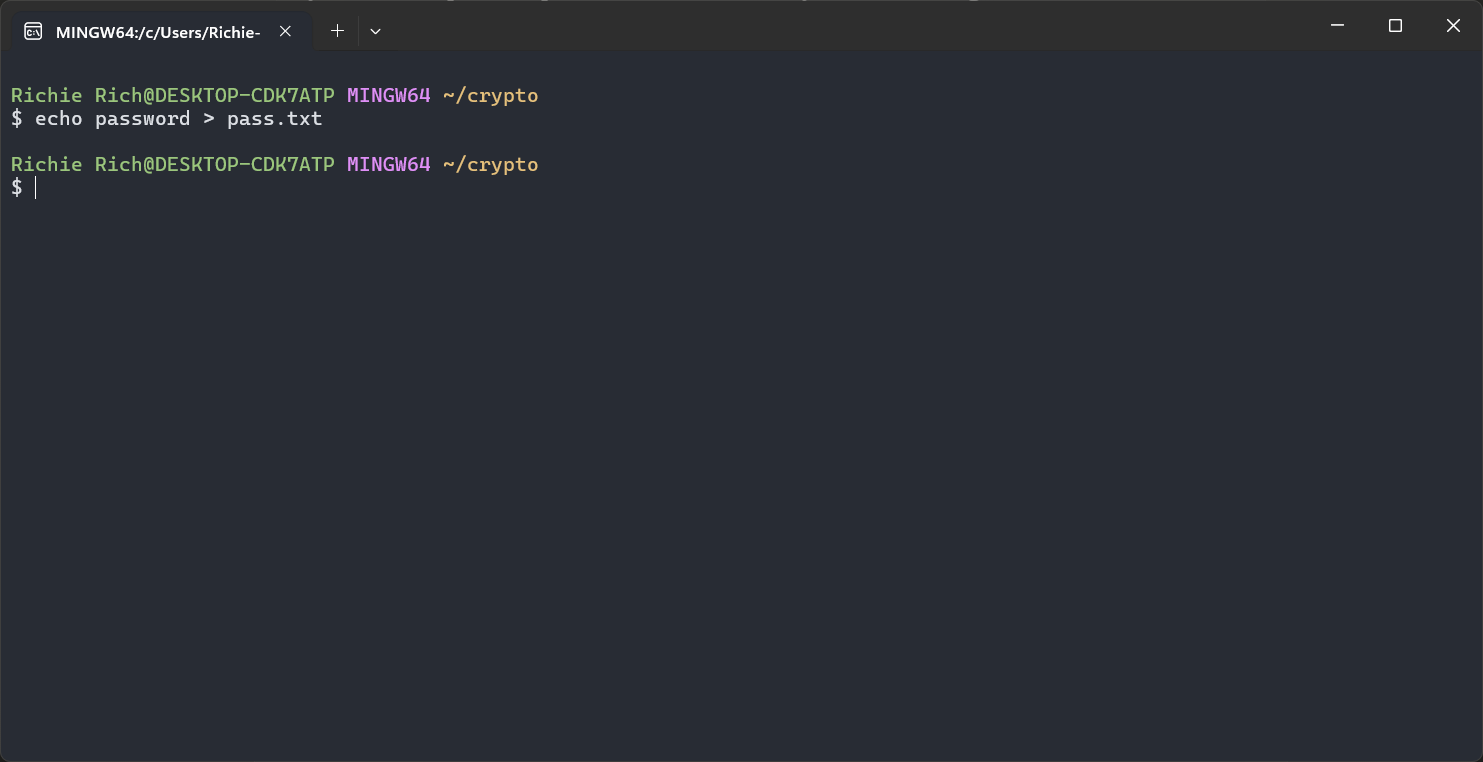




1. Decrypt “dme.txt” to “emed.txt”, compare “eme.txt” to “emed.txt”.



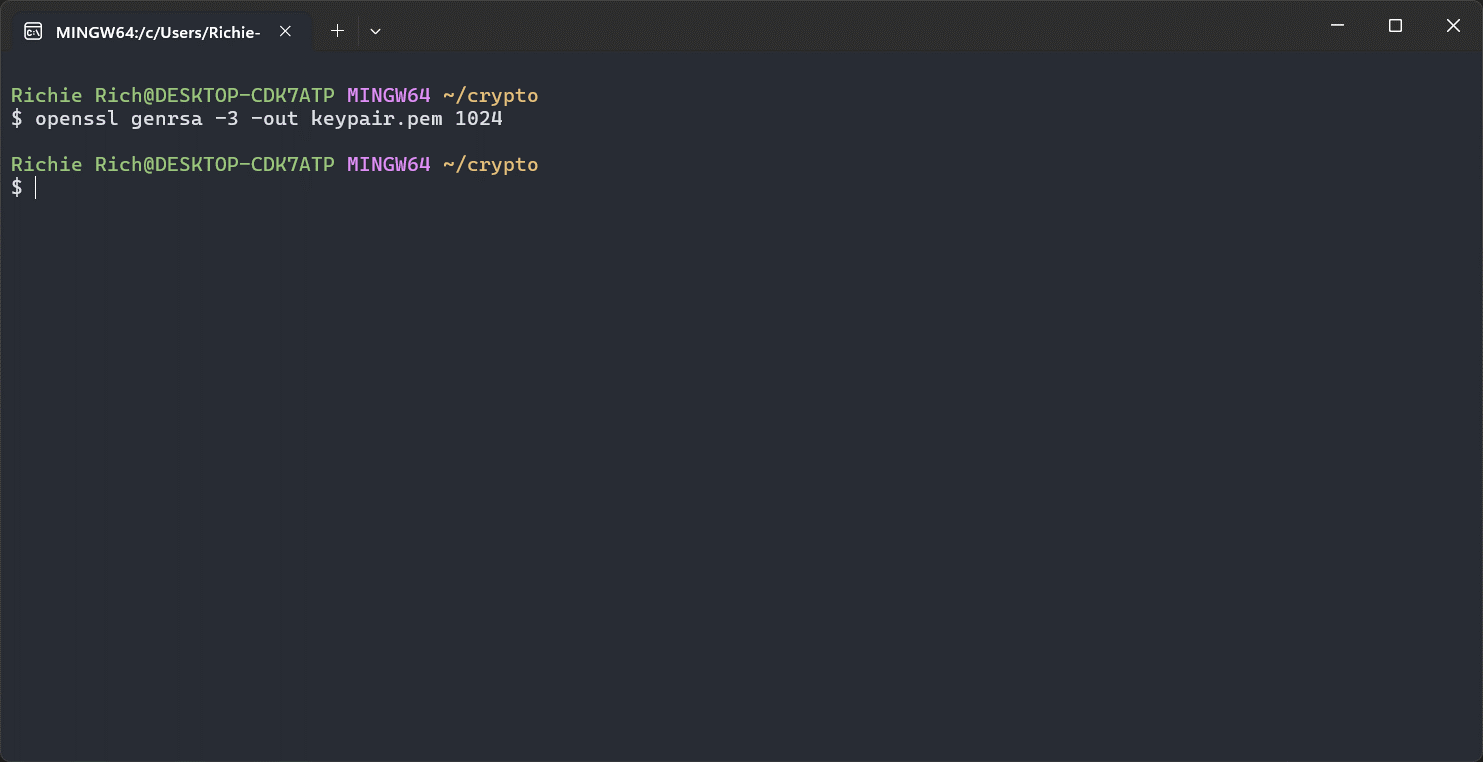


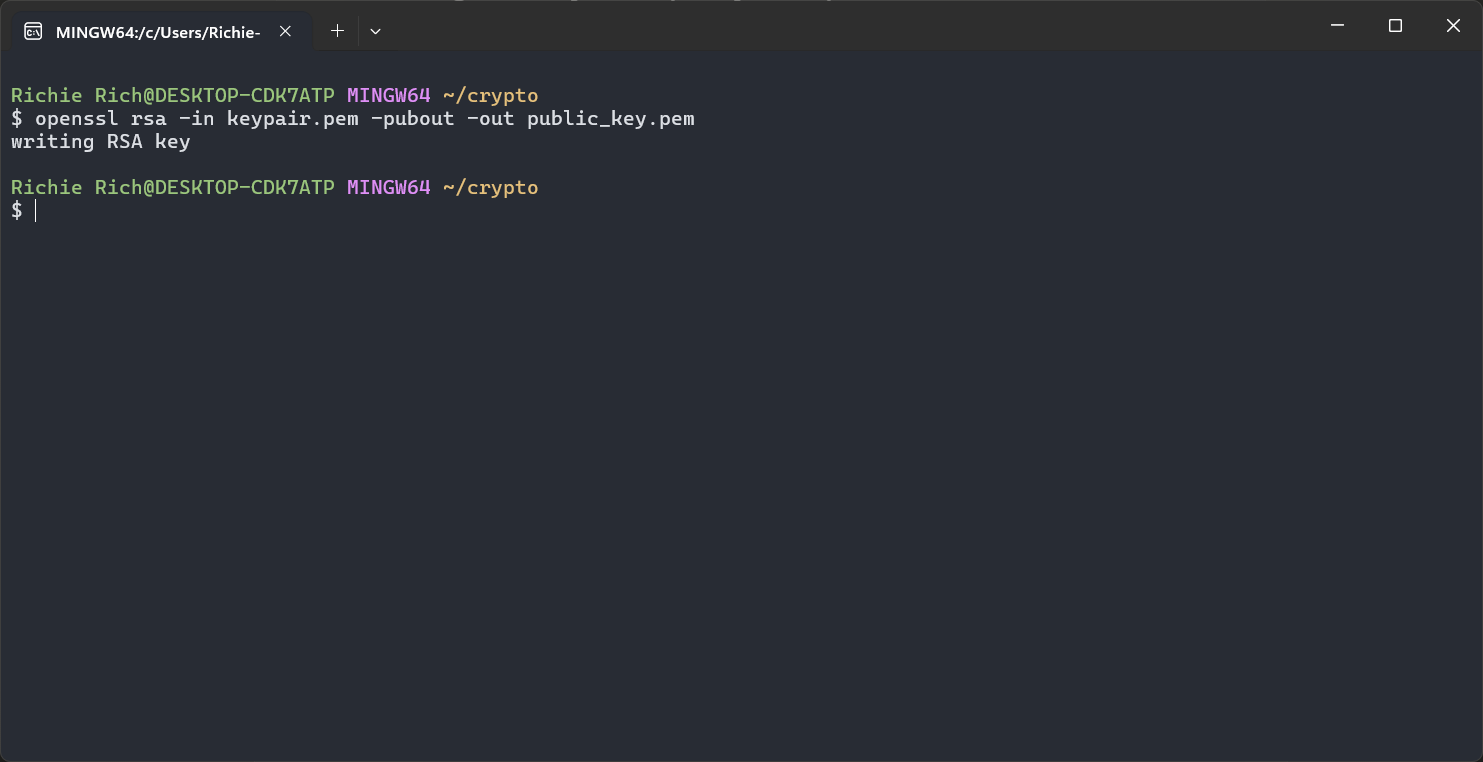
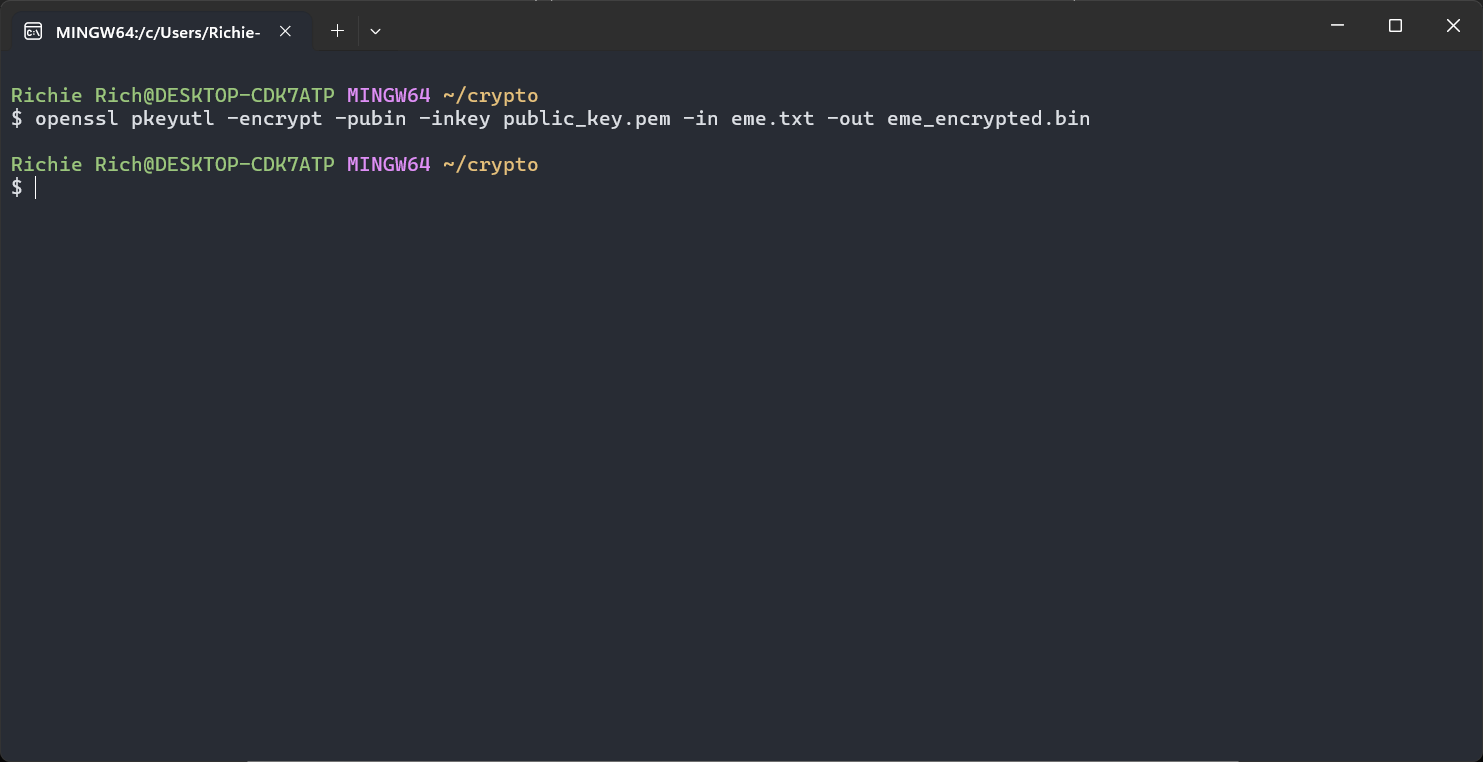
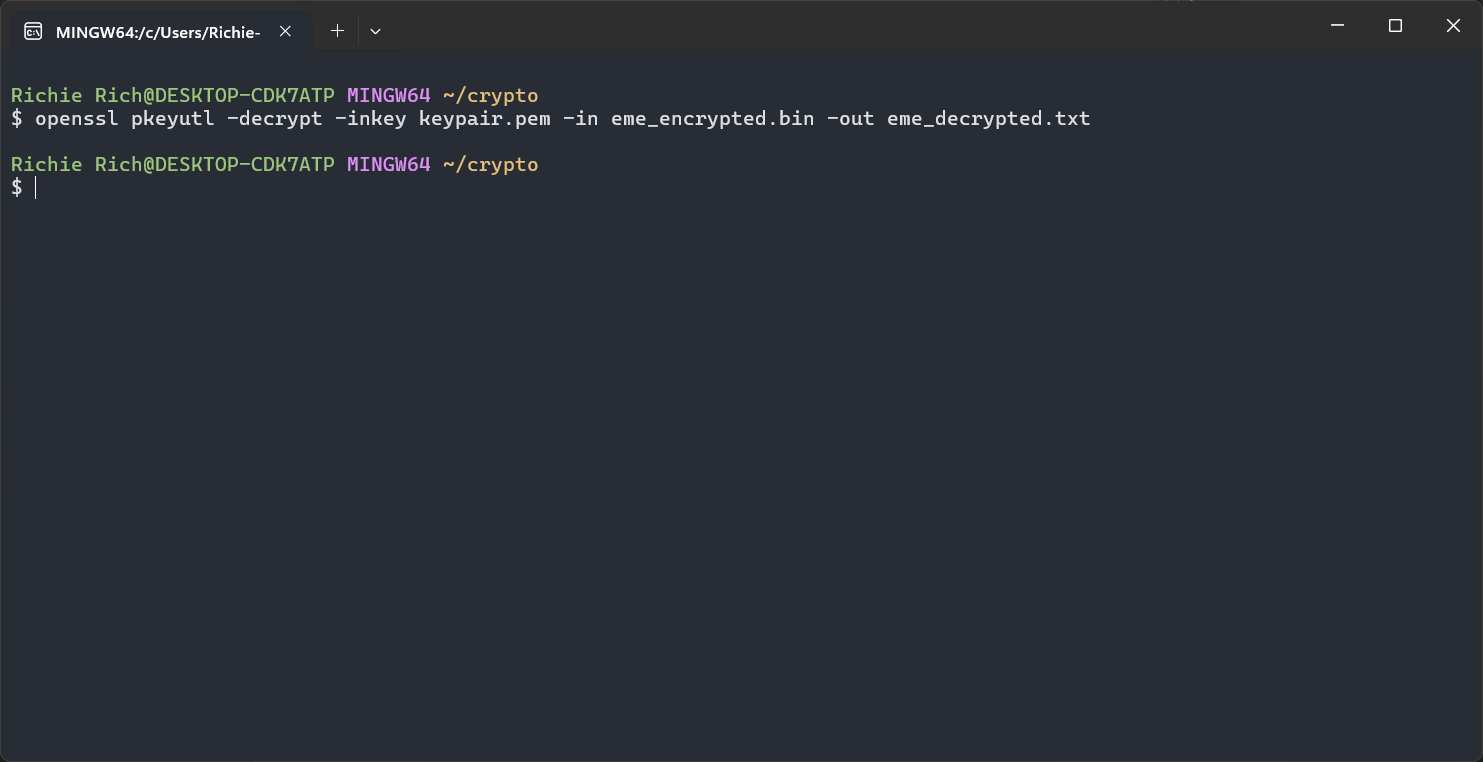
1. Encrypt “eme.txt” using DES to produce “dme.txt”; the key is password, and use a salt value of your choice. 
2. Encrypt “eme.txt” using DES in CBC mode to produce “dme-cbc.txt”; same key. 
3. Encrypt “eme.txt” using DES in EBC mode to produce “dme-ebc.txt”; same key. 
4. Encrypt “eme2.txt” using DES in EBC mode to produce “dme2-ebc.txt”; same key. 
5. Compare “dme-ebc.txt” and “dme2-ebc2.txt”. 
6. Encrypt “eme.txt” using RC4 with a 40bit key, decrypt and check the output. 
7. Create a file called “pass.txt” that contains “password” and use it to carry out sub-question c. 

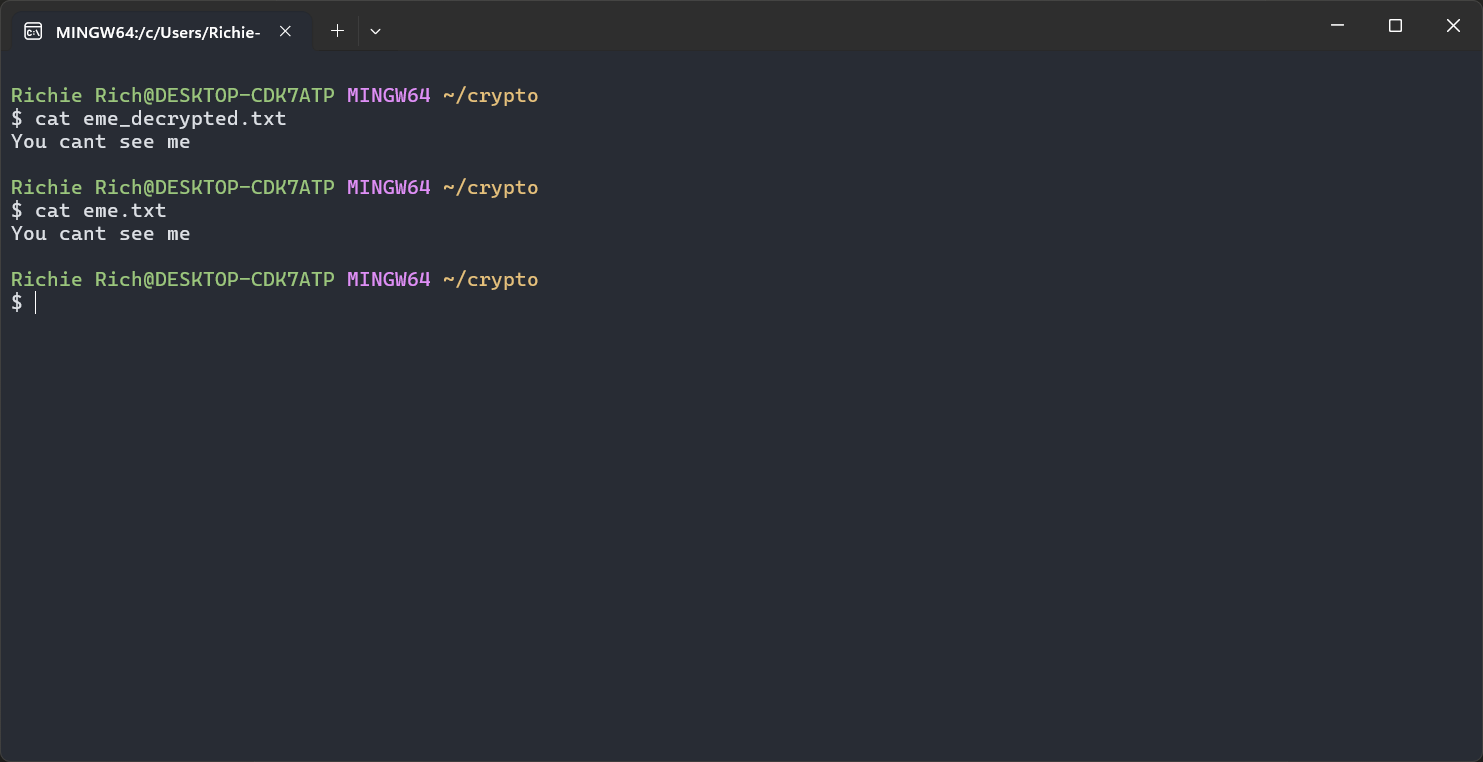
# PART 3: Use OpenSSL for Asymmetric Encryption

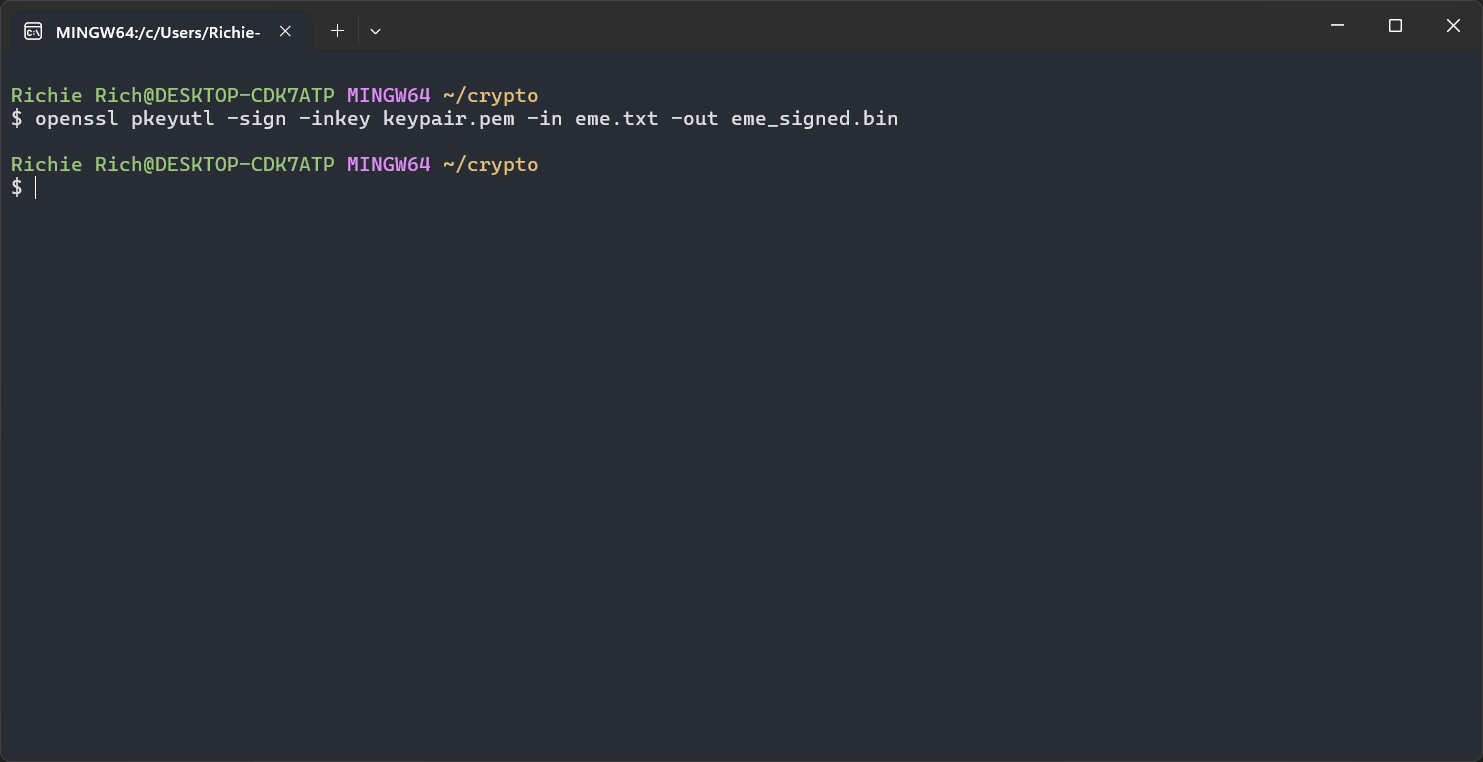
a. What is the PEM format?

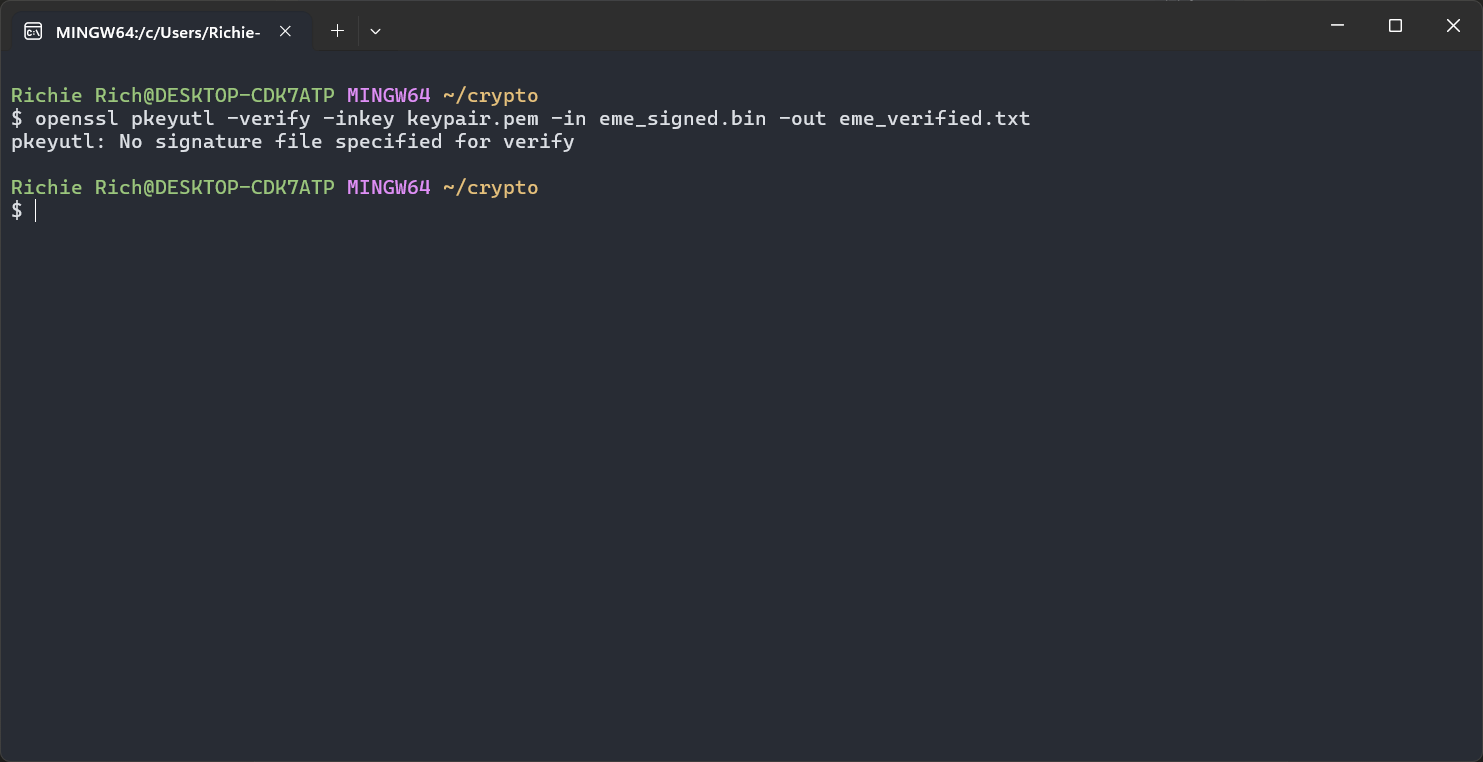
PEM (Privacy Enhanced Mail) is a base64 encoded DER certificate. It's a widely used format for storing and transmitting cryptographic keys, certificates, and other data. PEM files are text-based and typically have headers and footers that specify the type of data they contain (e.g., "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----").

b. Generate a RSA pair of keys with e=3 and a modulus of 1024 bits. 

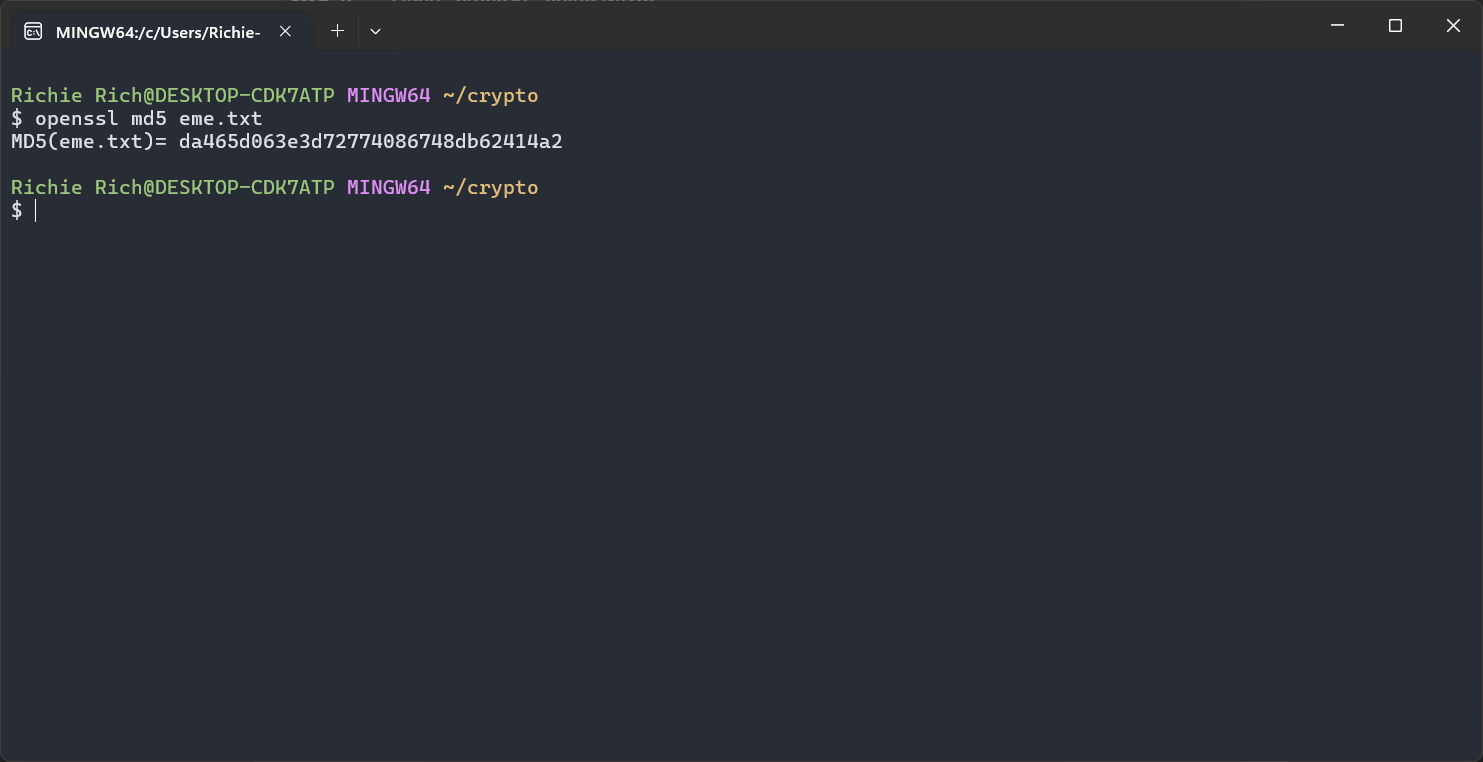
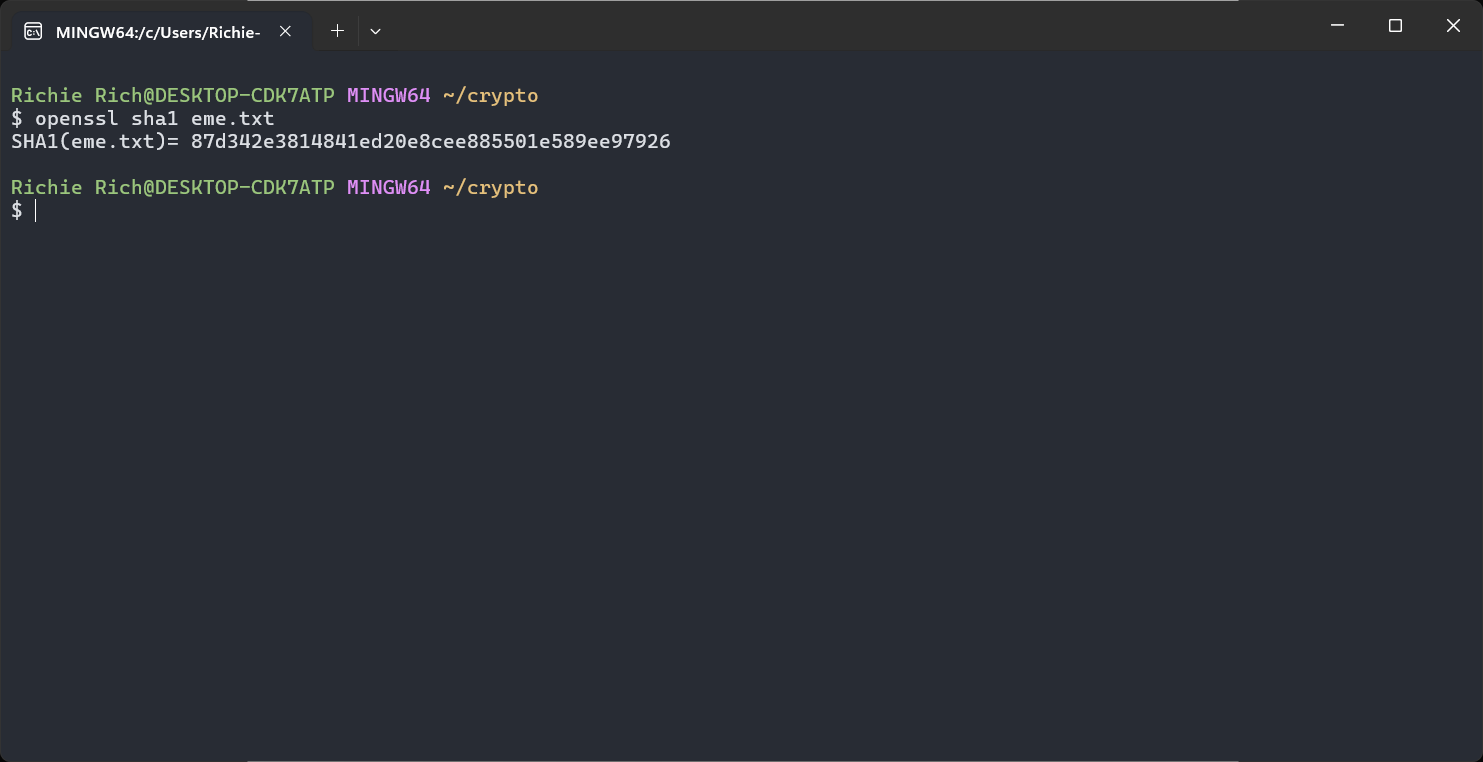
c. Use the public key to encrypt “eme.txt”, then decrypt it and compare the output to “eme.txt”.   



d. Use the private key to encrypt “eme.txt”, then decrypt it and compare the output to “eme.txt”. 



# PART 4: Use OpenSSL for Hashing

1. Hash “eme.txt” using MD5. 
2. Hash “eme.txt” using SHA1. 
3. Reproduce the digital signature process as described in the course slides. Sign “eme.txt” using SHA1 and RSA (use the keys you generated in part 3).

# PART 5: Use OpenSSL for Certificates

1. What is the CSR format?

CSR stands for Certificate Signing Request. It's a message sent from an applicant to a Certificate Authority (CA) to request a digital identity certificate. The CSR contains information that will be included in the certificate, such as the organization name, common name (domain name), locality, and country. It also contains the public key for which the certificate should be issued. The CSR is usually encoded in PKCS#10 format.

1. What is the ASN.1 format? How is it used in certificates context?

ASN.1 (Abstract Syntax Notation One) is a standard interface description language for defining data structures that can be serialized and deserialized across different platforms. It's used to describe the structure of data, not how the data is encoded.

In the context of certificates:

* ASN.1 is used to define the structure of various certificate-related data, including X.509 certificates, CSRs, and CRLs (Certificate Revocation Lists).
* It provides a way to specify complex data structures in a language-independent manner.
* ASN.1 definitions are used as the basis for encoding rules like DER and BER.

1. What is the DER format? How is it used in certificates context?

DER (Distinguished Encoding Rules) is a binary encoding format for ASN.1 data structures. It's a subset of BER (Basic Encoding Rules) with additional constraints to ensure a unique encoding for each ASN.1 value.

In the context of certificates:

* DER is commonly used to encode X.509 certificates, CSRs, and CRLs.
* It provides a compact, unambiguous way to represent certificate data.
* DER-encoded certificates can be further encoded into PEM format for easier handling in text-based systems.

1. Create a self-signed X.509 certificate for the keys generated in part 3. 