Practical Project Part 2

Introduction:

This is the second part of the practical project, this part is for implementing ECIES (elliptic curve integrated encryption scheme) and elliptic Schnorr digital signatures.

For this part I since the requirement asked for byte array input, I used `bin-text.txt` instead of `text.txt` .

`bin-text.txt` will store hexadecimal value instead of plain text

You can either use command line like below, do not forget to add `javac Main` in front of the command.

Second option is to use the run configurations included in the project.

Result:

- Generate an elliptic key pair from a given passphrase and write the public key to a file. **Completed**
- Encrypt a data file with ECIES under a given elliptic public key file, and write the ciphertext to a file. **Completed**
- Decrypt with ECIES a given elliptic-encrypted file from a password-derived private key and write the decrypted data to a file. **Completed**
- Sign with Schnorr a given file from a password-derived private key and write the signature to a file. **Completed**
- Verify with Schnorr a given data file and its signature file under a given public key file. **Completed**
    - **BONUS**: Combine encryption and signing as above in a single operation that signs with Schnorr a given data file under the sender’s password and encrypts the file and the signature with ECIES under the recipient’s public key. **Not Attempted**
    - **BONUS**: Combine decryption and verification as above in a single operation that decrypts the file and the signature with ECIES under the recipient’s password-derived private key and verifies the resulting data file with Schnorr under the sender’s public key file. **Not Attempted**

How to:

- ## Generate key pair:

```
generate <publickeyfile> <passphrase>
```

Example:

```
javac ./src/Main.java generate public_key.bin mypassphrase
```

- ## Encrypt a message from file:

```
encryptecies <datafile> <publickeyfile> <passphrase>
```

Example:

```
javac ./src/Main.java encryptecies bin-text.txt public_key.bin mypassphrase
```

- ## Decrypt a cryptogram under a passphrase:

```
decryptecies <encryptedfile> <passphrase>
```

Example:

```
javac ./src/Main.java decryptecies ECIES-encrypted.bin mypassphrase
```

- ## Sign with Schorr:

```
signature <datafile> <outputfile> <passphrase>
```

Example:

```
javac ./src/Main.java signature bin-text.txt signature.bin mypassphrase
```

- **Verify a signature:**

```
verify <datafile> <signaturefile> <publickeyfile>
```

Example:

```
javac ./src/Main.java verify bin-text.txt signature.bin public_key.bin
```