Practical Project Part 1

Introduction:

This project report outlines the implementation of a cryptographic library and application in Java, focusing on asymmetric encryption and digital signatures at the 256-bit security level. The project is part of the TCSS 487 Cryptography course and is developed in two parts.

I included multiple run configurations to use with Intellij IDE.

For example for running SHA3 hash function, go to run config and type in only `hash <filename>`

I used 1 single `text.txt` for all input. All input are in UTF-8 or plain text, then inputs are converted to byte array for processing. If you want a different input, just modify the `text.txt` file

Result:

I based most of my implementation on tiny_sha3, converting C code into Java code. The rest of the implementation is relatively simple. I used some online-tools to help with testing the accuracy of the hash function.

The project successfully implements the following key functionalities:

- Compute the SHA-3-256 and SHA-3-512 hashes for a user-specified file. **Completed**
  - **BONUS**: Compute also the SHA-3-224 and SHA-3-384 hashes of a user-specified file. **Completed**
- Compute SHAKE-128 and SHAKE-256 authentication tags (MACs) of user-specified length for a user-specified file under a user-specified passphrase. **Completed**
  - **BONUS**: Compute also SHAKE-128 and SHAKE-256 authentication tags (MACs) of user-specified length for text input by the user directly to the app (instead of having to be read from a file) under a user-specified passphrase. **Completed**
- Encrypt a user-specified data file symmetrically under a user-supplied passphrase. **Completed**
  - **BONUS**: Include a message authentication tag (MAC) in the cryptogram, using SHA-3-256 and the same symmetric key as for encryption. **Completed**
- Decrypt the symmetric cryptogram created by the encryption process above under the user-supplied passphrase. **Completed**
  - **BONUS**: Also verify the MAC from the cryptogram, computed as per the encryption bonus. **Completed**

Instruction:

Users can run the application from the command line: javac ./src/Main.java

- ## Compute SHA3 hash for file:

```
hashh <filename>
```

Exmaple:

```
javac ./src/Main.java hash text.txt
```

- ## Compute SHAKE MACs for file:

```
mac <filename> <security_level>
```

Example:

```
javac ./src/Main.java mac text.txt 256
```

- ## Encrypt a file symmetrically:

```
encrypt <filename> <passphrase>
```

Example:

```
javac ./src/Main.java encrypt text.txt mypassphrase
```

- ## Decrypt the symmetric cryptogram:

```
decrypt <filename> <passphrase>
```

Example:

```
javac ./src/Main.java decrypt text.txt.encrypted mypassphrase
```