**COLLABORATIVE PROJECT WITH INTEL**

**PROJECT TITLE :** Cryptography Simulation with mbedTLS/OpenSSL Library Usage and User Interaction.

**TEAM NAME :** ByteForge

**TEAM MENTOR :** Dr T V RAJINI KANTH, Professor & Head,

Department of CSE-AI&ML

[**rajinikanthtv@sreenidhi.edu.in**](mailto:rajinikanthtv@sreenidhi.edu.in)

[**rajinitv@gmail.com**](mailto:rajinitv@gmail.com)**,** Ph. No: 9849414375

**TEAM MEMBERS :**

S.Neelothpal 21311A6618 CSM Team lead [21311A6618@sreenidhi.edu.in](mailto:21311A6618@sreenidhi.edu.in)

D.Sathvika 21311A6619 CSM Team member [21311A6619@sreenidhi.edu.in](mailto:21311A6619@sreenidhi.edu.in)

M.Sri Priya Bhargavi 21311A6626 CSM Team member [21311A6626@sreenidhi.edu.in](mailto:21311A6626@sreenidhi.edu.in)

MD.Abdul Azeez21311A6633 CSM Team member [21311A6633@sreenidhi.edu.in](mailto:21311A6633@sreenidhi.edu.in)

M.Anudeep 21311A6665 CSM Team member [21311A6665@sreenidhi.edu.in](mailto:21311A6665@sreenidhi.edu.in)

**INSTITUTE NAME** **:** Department of CSE-AI&ML

Sreenidhi Institute of Science and Technology

Yamnampet, Ghatkesar

Hyderabad - 501301

**DATE OF SUBMISSION:** 14-07-2024

**ABSTRACT**

This project presents an interactive simulation tool for cryptography, utilizing mbedTLS. The tool is designed to educate users on the principles of encryption, decryption, key generation, and various cryptographic algorithms such as RSA, AES, DES, and SHA. By implementing a robust cryptographic engine and an intuitive user interface, the tool provides an engaging platform for both theoretical understanding and practical application of cryptographic concepts. This simulation aims to serve as a valuable resource for students, educators, and professionals in the field of cybersecurity, enhancing their knowledge and skills through hands-on interaction and experimentation.

**KEYWORDS**

Cryptography, mbedTLS, Encryption, Decryption, Key Generation, Interactive Simulation.

**INTRODUCTION**

Cryptography is an essential field in information security, underpinning the protection of data in various domains such as communication, finance, healthcare, and government. The rapid advancement of digital technologies has increased the need for robust cryptographic solutions to safeguard sensitive information. However, the complexity of cryptographic algorithms and their underlying mathematics often poses a significant barrier to understanding and applying these techniques effectively.

This project aims to address these challenges by developing an interactive educational tool that simulates key cryptographic processes. By leveraging widely-used cryptographic libraries such as mbedTLS, the tool enables users to experiment with encryption, decryption, and key generation, providing a hands-on approach to learning.

**MOTIVATION**

The motivation behind this project is to provide an accessible and engaging way to learn about cryptography. Traditional learning methods often lack interactivity, making it difficult to grasp complex concepts. This tool addresses that by offering a hands-on approach, enabling users to see the effects of different cryptographic algorithms and parameters in real-time. The project aims to enhance learning experiences for students, educators, and professionals in the field of cybersecurity.

**TECHNOLOGIES USED**

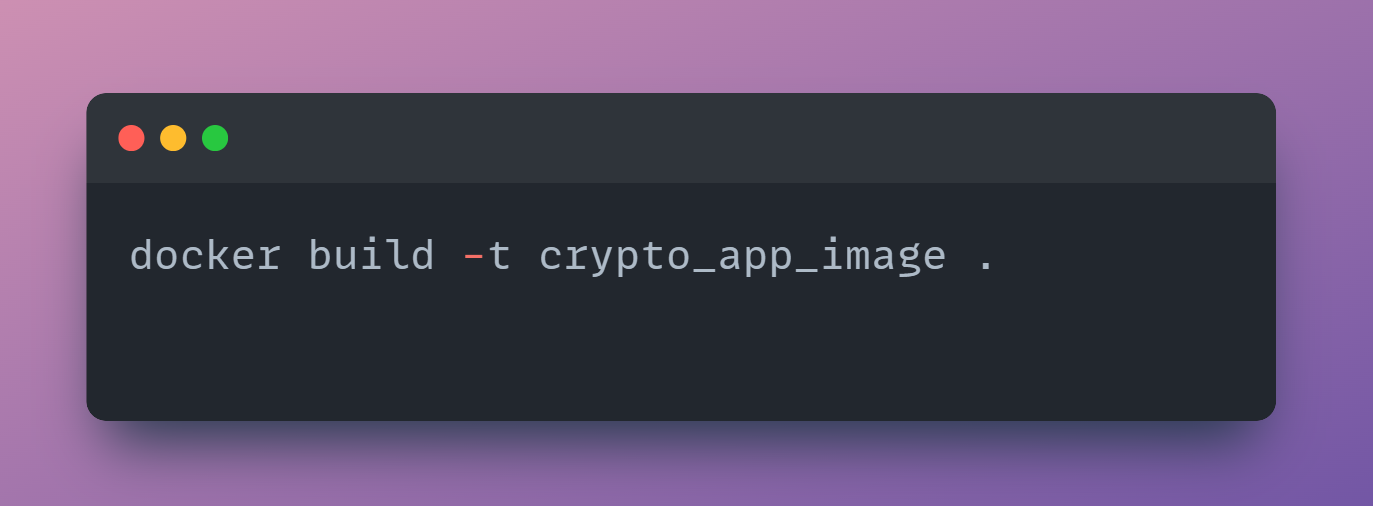
* **Development Tools:** GCC (GNU Compiler Collection), Git
* **Libraries:** mbedTLS
* **User Interface:** Command Line
* **Documentation**: Markdown
* **Containerization:** Docker

**PREREQUISITES**

* **Docker:** Ensure that you have docker installed on your system**.**

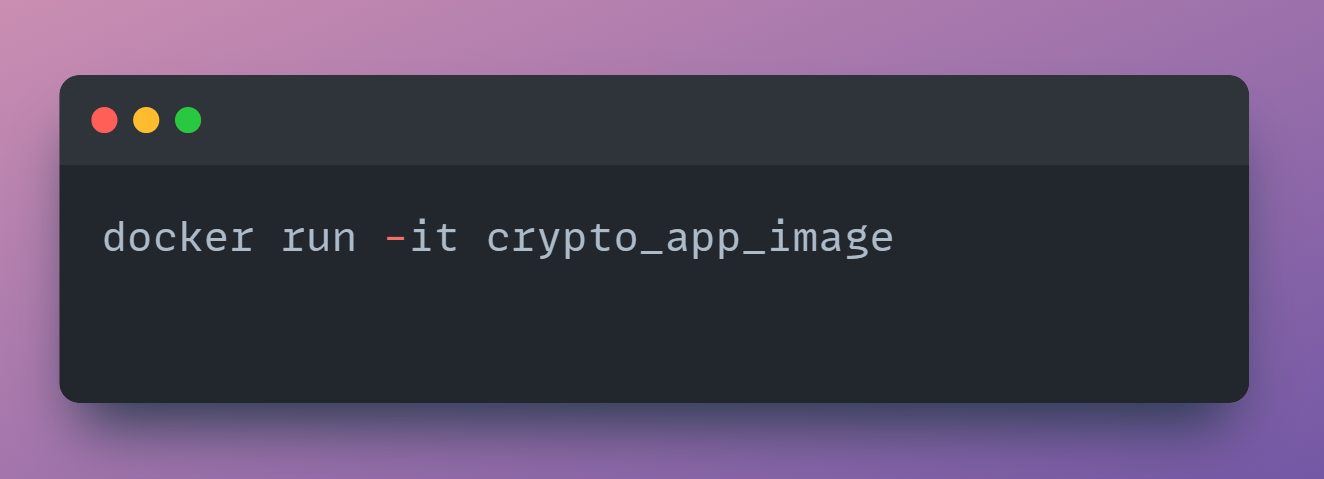
**COMPILATION**

To compile the code and create a docker image use the following command:



**USAGE**

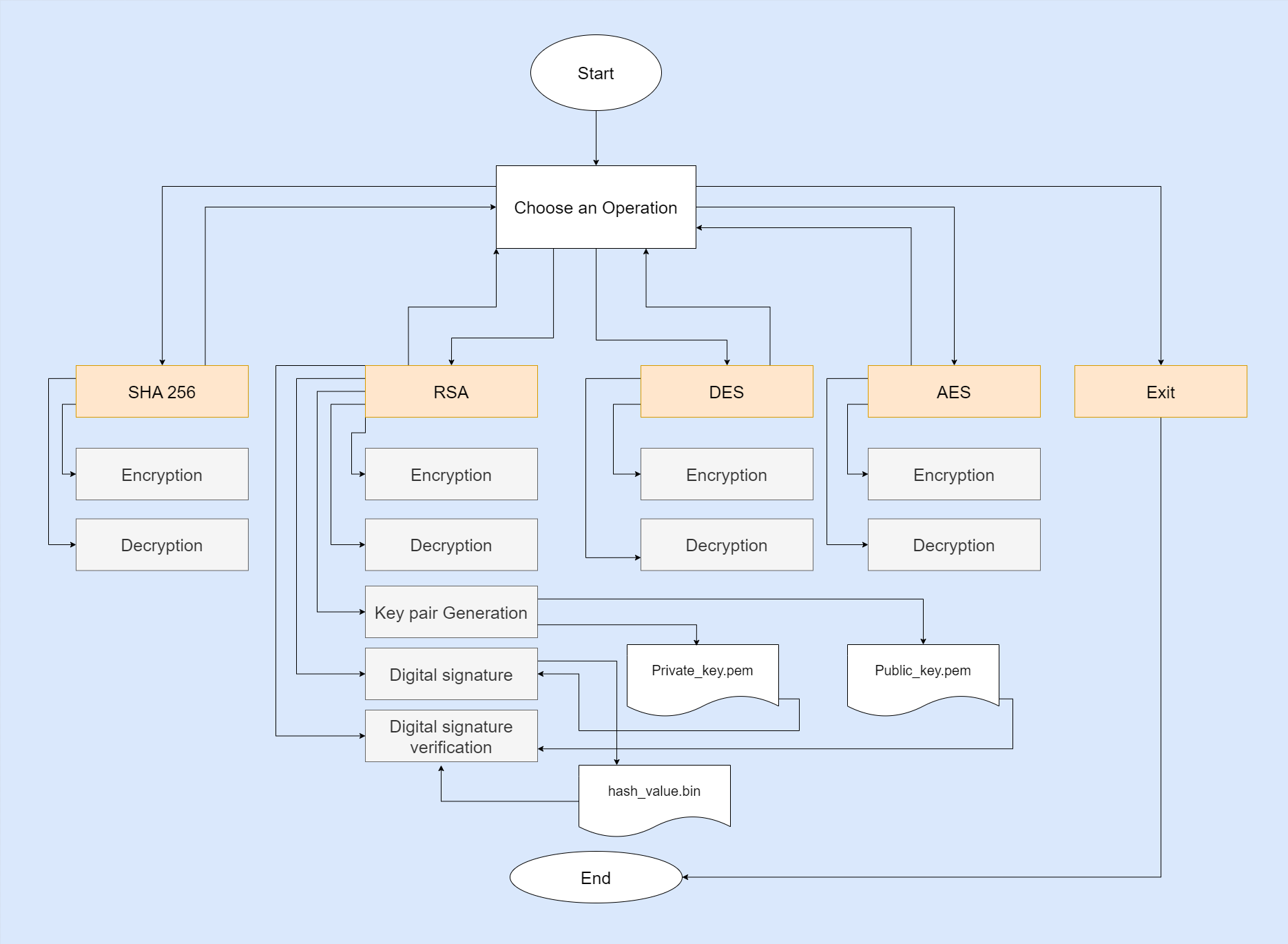
Run the compiled program and follow the prompts to select the desired cryptographic operation:

****

**FILES**

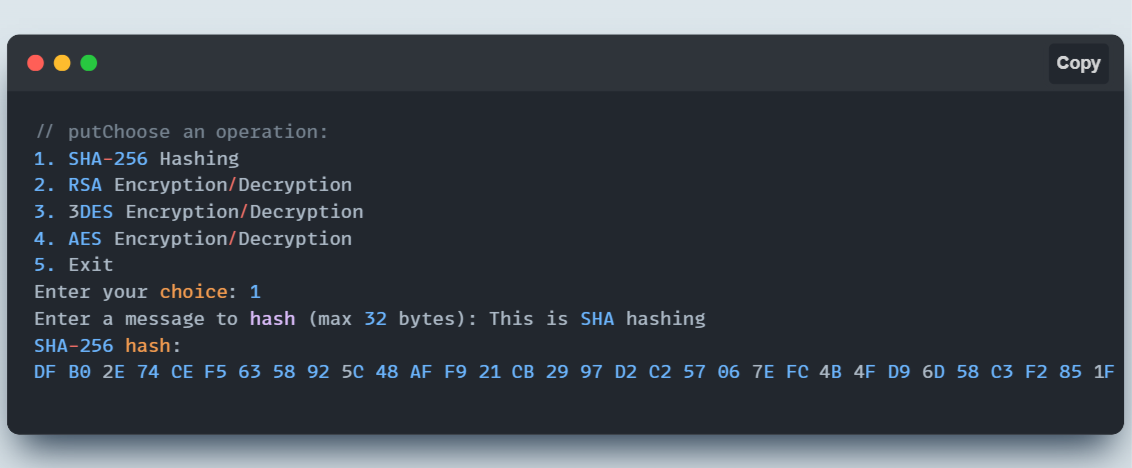
* **`private\_key.pem`:** Contains the RSA private key.
* **`public\_key.pem`:** Contains the RSA public key.
* **`hash\_value.bin`:** Contains the hash value of a message for digital signature.
* **`crypto\_operations`:** The compiled binary executable.
* **`dockerfile`:** The docker file that is used to create the docker image.

**ARCHITECTURE**

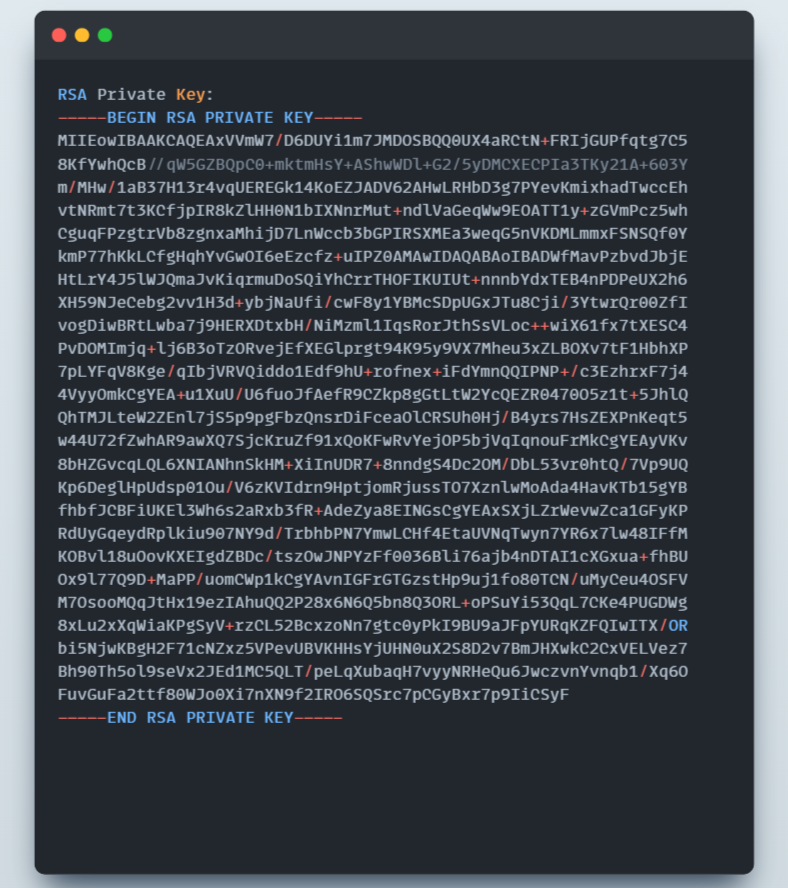
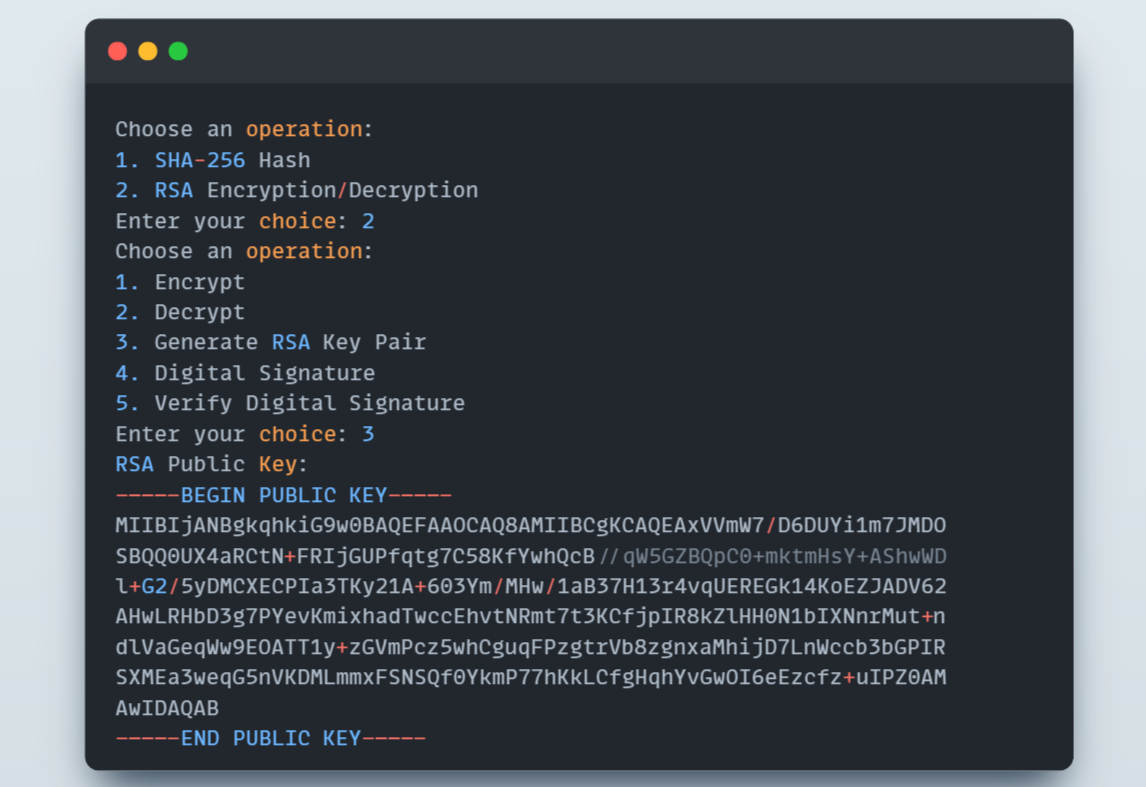
****

**RESULTS**

#### SHA-256 Hashing:



#### RSA Algorithm:



#### 3DES Algorithm:

#### 

#### AES Algorithm:

**CONCLUSION**

In conclusion, the cryptography simulation project represents a significant advancement in the field of cybersecurity education. By providing an interactive and user-friendly platform, the project has succeeded in demystifying complex cryptographic concepts and making them accessible to a broad audience. The usage of mbedTLS library has enabled users to experiment with various cryptographic algorithms, enhancing both their theoretical understanding and practical skills.

Looking ahead, the project's open-source nature and commitment to continuous improvement ensure its ongoing relevance and impact. As the field of cybersecurity evolves, the cryptography simulation tool will remain a vital educational resource, adapting to new challenges and innovations. Ultimately, this project underscores the importance of practical, interactive learning in the advancement of cryptographic knowledge and skills.

**FUTURE SCOPE**

Future enhancements of this project:

* Expanding the range of cryptographic algorithms supported
* Developing a more sophisticated GUI for enhanced user interaction
* Adding support for mobile platforms (iOS, Android)
* Integrating real-world case studies and interactive tutorials
* Providing multilingual support to reach a wider audience
* Implementing advanced cryptographic concepts such as elliptic-curve cryptography (ECC) and quantum-safe algorithms

**REFERENCES**

* **mbedTLS Documentation:** <https://tls.mbed.org/tech-updates/releases>
* **OpenSSL Documentation:** <https://www.openssl.org/docs/>
* **GCC Documentation:** <https://gcc.gnu.org/onlinedocs/>
* **Git Documentation:** <https://git-scm.com/doc>
* **GTK Documentation**: <https://www.gtk.org/docs/>
* **Qt Documentation:** <https://doc.qt.io/>

**SOURCE CODE AND CONFIGURATION FILES**

Source code for the project is attached to the GitHub Link:

[**Cryptography Simulation**](https://github.com/Neelothpal/Intel-Project)