Text Encryption and Decryption using Cryptography Algorithms

April 22, 2025

1 Background

Cryptography is a critical field in both theoretical and applied mathematics, particularly in the realm of secure communications. The RSA [RSA78] and ElGamal [Gam84] encryption algorithms are two prominent public-key cryptosystems that have been widely adopted for their robust security properties. RSA, named after its inventors Rivest, Shamir, and Adleman, leverages the difficulty of factoring large numbers to provide secure encryption and decryption. ElGamal, on the other hand, is based on the discrete logarithm problem and offers a different approach to achieving secure communication. Both algorithms are essential for understanding modern cryptographic practices and are widely used in various applications, from secure web transactions to encrypted messaging.

2 Objective

The objective of this project is to provide a comprehensive understanding of the RSA and ElGamal encryption algorithms through practical implementation and analysis. You will implement both algorithms in Python, focusing on key generation, encryption, and decryption processes. You will also analyze the security and performance characteristics of each algorithm, comparing their strengths and weaknesses. By the end of the project, students should be able to apply these algorithms to encrypt and decrypt messages and understand their theoretical foundations as well as practical applications in securing digital communications.

3 Task Description

Each group will implement a modular encryption/decryption system for textual messages:

- Implement the **RSA** algorithm with the key generation, encryption, and decryption functions.
- Implement the **ElGamal** algorithm with the key generation, encryption, and decryption functions.
- Test the implementation with the encryption/decryption of short and long **"textual"** messages.
- Analyze the performance and security level of **RSA** and **ElGamal**.

Students are encouraged to experiment with different parameters (e.g., sizes of prime numbers, public exponent, etc) and analyze their impact on performance. Beyond the required cryptography methods, students are also welcome to explore more recent or advanced cryptography techniques or propose and prototype novel approaches. Innovation and thoughtful system design will be taken into account in the grading rubric, particularly under the criterion of "Application and Innovation."

4 Tools

You may use any standard Python packages (but not existing RSA or ElGamal libraries) for implementation.

5 Team Organization

- Each team should consist of **4 students**.
- Team members are encouraged to divide responsibilities (e.g., basic functions, evaluation, and drafting report) and all members should participate in the algorithm implementation.
- Teams are free to organize themselves, but must submit one final project as a group.

6 Deliverables

Each group is required to submit the following components:

- **Code**: A well-organized Python project with modular structure, inline documentation, and clear instructions for installation and execution (e.g., via `README.md` or Jupyter notebooks).
- **Report**: A concise report (3–5 pages) summarizing the project. The report should include the following sections:
  - **Motivation**: Why shortest path finding matters and what this project aims to solve.
  - **Related Work**: Brief summary of the used technology.
  - **System Design**: Description of the architecture, modules, and key design choices.
  - **Method Comparison**: Analysis of the two approaches in terms of performance and their security levels.
  - **Experimental Setup and Results**: Analyzed results, and visualizations.
  - **Conclusion and Future Work**: Summary and potential directions for improvement.
- **Presentation**: Each group will attend a short (8-minute) presentation session with TAs and lecturers to demonstrate their system, findings, and answer questions. Presentation quality will also be considered in grading.

References

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.