**Caesar Cipher and RSA Encryption**

**1. Project Title:**

**Cryptography: Implementation and Analysis of Caesar Cipher and RSA Encryption**

**2. Abstract:**

The primary goal of this project is to demonstrate the concepts of cryptography by implementing two widely studied encryption algorithms: **Caesar Cipher** and **RSA Encryption**. The project involves encoding and decoding messages using both algorithms to explore their functionality and compare their strengths and weaknesses. Caesar Cipher is a simple substitution cipher, while RSA is an advanced asymmetric encryption algorithm based on number theory, offering enhanced security. By implementing these algorithms in C++, this project aims to provide a practical understanding of classical and modern encryption techniques.

**Sure! Here are the problems in bullet points:**

**3. Problems:**

* **Security of Caesar Cipher:**
  + **The Caesar Cipher is vulnerable to brute-force attacks since it has a limited number of possible shifts (only 25 possibilities).**
  + **The simplicity of the algorithm makes it insecure for real-world applications, as it can be easily decrypted without knowing the key.**
* **Complexity of RSA Algorithm:**
  + **The RSA algorithm requires efficient prime number generation and large number calculations, which can be computationally expensive and time-consuming, especially with large key sizes.**
  + **Ensuring the security of the private key and preventing attacks such as factoring large numbers is challenging.**
* **Performance and Efficiency:**
  + **RSA encryption and decryption can be slow for large datasets due to the complexity of the algorithm, making it less efficient for real-time applications unless optimizations are applied.**
  + **The time complexity of RSA is significantly higher compared to simpler algorithms like Caesar Cipher, which may not be suitable for large-scale use cases without additional measures.**
* **Implementation Challenges:**
  + **Understanding and correctly implementing the key generation process for RSA can be difficult, especially when working with large prime numbers.**
  + **Handling mathematical operations efficiently and accurately, such as modular exponentiation for RSA, requires careful consideration of computational resources.**

**4. Objectives:**

* **Implementation of Caesar Cipher**: To understand and demonstrate the simplicity of symmetric encryption through a classical algorithm.
* **Implementation of RSA Encryption**: To explore and implement an asymmetric encryption algorithm that ensures higher security through the use of public and private keys.
* **Security Comparison**: To evaluate the security strengths and weaknesses of both encryption methods.
* **Performance Analysis**: To compare the computational efficiency of both algorithms, with focus on time complexity.
* **Practical Application**: To provide a working demo that encrypts and decrypts messages using both algorithms.

**5. Scope of Work:**

The scope of this project includes:

* Implementing the **Caesar Cipher** algorithm where the plaintext is shifted by a constant key value.
* Implementing the **RSA Encryption** algorithm with key generation, encryption, and decryption processes.
* Performing a **comparison of the algorithms** in terms of security, efficiency, and practical use cases.
* Creating a **C++ program** that allows users to input plaintext, select the encryption method, and view the results.
* **Documentation** detailing the algorithms, their implementation, performance metrics, and comparison.

**6. Methodology:**

* **Caesar Cipher:**
  + A substitution cipher where each letter in the plaintext is replaced by a letter with a fixed number of positions down or up the alphabet.
  + The encryption and decryption process involves shifting the alphabet using a key (e.g., shifting 3 positions).
* **RSA Encryption:**
  + **Key Generation**: Choose two large prime numbers and calculate the public and private keys.
  + **Encryption**: Convert the plaintext message into numerical form, and then encrypt using the public key.
  + **Decryption**: The encrypted message is decrypted using the private key, recovering the original message.

Both algorithms will be implemented using C++, ensuring the program is well-structured and the encryption/decryption processes are efficiently handled.

**7. Tools and Technologies:**

* **Programming Language**: C++
* **Development Environment**: Visual Studio, Code::Blocks, or any other C++ IDE.
* **Libraries**: Standard C++ libraries for basic I/O and mathematical operations.

**8. Expected Results:**

* A functional C++ program that successfully implements both Caesar Cipher and RSA encryption algorithms.
* A comparison report analyzing the encryption strength and efficiency of the two algorithms.
* A clear demonstration of how each algorithm works and where each is best applied.

**9. Deliverables:**

* **C++ Code**: Well-documented code for Caesar Cipher and RSA Encryption algorithms.
* **Project Report**: A comprehensive report detailing the implementation process, challenges, and results.
* **Presentation**: A concise presentation that summarizes the project, including key points and findings.
* **Working Demo**: A simple user interface where users can input text and select encryption methods.

**10. Challenges and Limitations:**

* **Complexity of RSA**: The RSA algorithm requires careful handling of large numbers and prime number generation. Implementing efficient key generation may require additional libraries for advanced number theory.
* **Security of Caesar Cipher**: Caesar Cipher is easy to break with brute-force methods, making it unsuitable for real-world use, but it’s valuable for understanding basic encryption concepts.
* **Performance**: RSA is computationally more expensive compared to Caesar Cipher, especially with large keys, which may slow down encryption and decryption.

**11. Conclusion:**

This project will provide a comprehensive overview of cryptography through the practical implementation of two key encryption algorithms—Caesar Cipher and RSA Encryption. It will help develop a deeper understanding of encryption techniques, their applications, and limitations. Additionally, this project will improve skills in C++ programming, algorithm design, and cryptography, all of which are essential for any student pursuing computer science or cybersecurity.