Lab 2

Contents

1. Public-Key Cryptography and RSA Algorithm:

1.1. Introduction to Public-Key Cryptography:

Public-key cryptography is a widely used cryptographic technique that involves two keys – a public key, which is shared with everyone, and a private key, which is kept secret. These two keys are mathematically linked, and together they ensure secure data transmission over insecure channels.

The most well-known public-key cryptosystem is RSA (Rivest–Shamir–Adleman), which is based on the difficulty of factoring large prime numbers.

1.2. RSA Algorithm:

2. Theory Behind My Custom Algorithm

We Are AISD Students, so we can think of the implementation of AI in improving the algorithm, while we can use some complex ideas like RNN or a neural network overall, it would be complex and computationally expensive and heavy

But we can also optimize the drawback of the Algorithm, thus the algorithm I have proposed is inspired by RSA but introduces modifications that aim to address some of its known limitations, such as vulnerabilities in key generation and improvements in computational efficiency.

2.1. Prime Number Selection Improvement:

In traditional RSA, selecting two large primes, P and Q, is a critical part of ensuring the algorithm's security. My algorithm introduces a more robust method of prime number selection by incorporating randomness and extra checks to avoid small or weak primes, thereby improving the difficulty of factoring n.

2.2. Enhanced Modulus Operation:

In RSA, all encryption and decryption operations are done using modulo n, which can lead to inefficient calculations for very large numbers. My algorithm proposes an optimized version of these modular exponentiation operations using more advanced number-theoretic techniques, such as the Chinese Remainder Theorem (CRT)[1] or Montgomery multiplication[2], which increases the speed of both encryption and decryption without compromising security.

2.3. Improved Key Generation:

The traditional RSA key generation process can be vulnerable if e is not chosen properly, as certain values of e (like 3 or 65537) can expose the algorithm to small exponent attacks. My proposed algorithm strengthens this step by introducing a more dynamic selection of e, which adjusts based on the properties of n, making it harder to predict and thus improving resistance to these types of attacks.

2.4. Encryption/Decryption Process:

My encryption and decryption processes remain mathematically similar to RSA but incorporate additional layers of randomness during encryption. For example, my algorithm might add random padding or mix in random elements during encryption to create unique ciphertexts even when the same message is encrypted multiple times. This guards against replay attacks or ciphertext pattern analysis.

3. Improvements Over RSA:

3.1. Security:

1. **Randomness in Prime Selection:** By improving the prime selection process, my algorithm enhances security by making n harder to factor, which is a critical aspect of breaking RSA.
2. **Dynamic e-Value Selection:** The dynamic generation of e, as opposed to a fixed value, strengthens the resistance to small exponent attacks and reduces vulnerabilities.
3. **Randomized Encryption:** The introduction of randomized encryption makes it difficult for attackers to gain insights into the plaintext based on the ciphertext, even if the same message is encrypted multiple times.

3.2. Efficiency:

4. **Optimized Modulus Operations:** Using optimized modular exponentiation techniques like CRT or Montgomery multiplication speeds up both encryption and decryption, which is especially beneficial when dealing with large keys.
5. **Improved Decryption Process:** The use of enhanced number-theoretic methods to perform decryption leads to a reduction in computational time, making the algorithm more practical for real-time applications without compromising on security.

3.3. Scalability:

6. **Better Key Size Adaptability:** While RSA faces limitations in key size for practical reasons (e.g., too large keys become inefficient), My algorithm's improvements allow for larger key sizes without a proportional increase in computational time. This enhances its applicability in systems where security needs are higher.

4. Conclusion:

My encryption algorithm is a modified version of RSA that addresses both efficiency and security concerns. By introducing randomness in the prime selection process, dynamically choosing the public exponent e, and optimizing the modular operations, my algorithm offers improvements over traditional RSA. These changes make it harder to crack through cryptographic attacks while improving the speed and scalability of encryption and decryption processes.

Thus, my proposed encryption algorithm offers a stronger alternative to RSA while maintaining the core principles of public-key cryptography.

Note that this came mainly from the idea of including AI (or in this case simpler optimization methods) into the Algorithm. This modified improvement could be already proposed or published online; in which case I don't own the idea.

References:

1. Author(s), *The Complexity of the Chinese Remainder Theorem*. ResearchGate, Link.
2. Author(s), *FFT-Based Montgomery Multiplication for Cryptographic Systems*. ResearchGate, Link.