谢泽钰 2020012544 致理-数02

HW-1 RSA Encryption/Decryption Cryptosystem Implementation

Description

In this project, I independently implemented an RSA encryption/decryption cryptosystem using C++, which has been verified for reliability by 334 groups of RSA-128 random tests.

However, there are still two unresolved disadvantages with this project:

1. I employed the Miller-Rabin Test to generate random large prime numbers. However, I have only identified 8 pseudoprimes up to the first eight prime numbers. I have not found research on larger pseudoprimes, nor can I compute them using my MacBook. This implies that although the likelihood is small, it is still possible to generate composite numbers.
2. I packaged the entire project within a Docker container and ran it on both my MacBook and Lenovo PC. Surprisingly, it runs approximately 5 times faster on the Mac than on the PC. Nonetheless, it still takes 10 seconds to process one group of data. Despite attempting various optimization methods, the speed remains unsatisfactory. 🐌

Despite these two disadvantages, I completed the testing and recorded the data in `log.txt`, while other files comprise the codes and executable files. `Bint.hpp` serves as the header file for a class designed to process very large integers, whereas `func.hpp` contains simple functions necessary for the process.

For validation purposes, you can execute `docker build -t gcc_11 .` to construct the Docker image based on the current file. To run the container, please utilize `docker run -i -v .:/app -t gcc_11`. If you are utilizing a Unix-based system, it is preferable to directly execute `./main.sh`, which is a simple script file for smoother execution of the Dockerfile.

P.S. Alternatively, you may choose to run the makefile independently without using Docker. However, please be aware that the random seed I have employed might not function as anticipated, particularly on Windows platforms.

Algorithm

RSA Encryption/Decryption

Given a short string s (no more than 128 bits), we aim to encrypt and decrypt it with RSA-128.

Firstly, we convert s into a large integer M, ensuring that $M < 2^{128} < 10^{39}$.

Next, we find two random prime numbers p and q such that $p, q \geq 10^{40} > 10^{39} > M$.

Let $n = pq$, hence $\phi(n) = (p-1)(q-1)$.

Let $e = 2^{16} + 1 = 65537$, it is established that e is a prime number. We require $\gcd(e, p-1) = \gcd(e, q-1) = \gcd(e, \phi(n)) = 1$. If this condition is not met, we regenerate p and q until success.

Using the extended Euclidean algorithm, we can find d such that $d \equiv e^{-1} \bmod \phi(n)$.

With these parameters, we define the RSA public key as (n, e) and the private key as (n, d).

During encryption, we calculate $M' \equiv M^e \bmod n$, and M' becomes the encrypted data. During decryption, we compute $M'' \equiv (M')^d \equiv M^{ed} \bmod n$. Accoring to Euler's Theorem, $M^{\phi n} \equiv 1 \bmod n$, thus $M^{ed} \equiv M^{k\phi(n)+1} \equiv M \bmod n$.

Hence, $M'' = M$ represents the plaintext and decrypted data.

Miller-Rabin Test

We seek to determine whether an positive integer n is prime. For small n, we can attempt division with all prime number $p \leq n$. However, this approach costs $O(\sqrt{n})$ time, which becomes impractical for large n.

Miller-Rabin devised a clever method to address this issue.

Given a prime number p and another positive integer a such that $\gcd(a,p) = 1$, by Fermat Little Theorem, we have $a^{p-1} \equiv 1 \bmod p$. Suppose $p - 1 = 2^s d$, then we consider the following s numbers: $a^d, a^{2d}, a^{2^2 d}, \ldots, a^{2^{s-1}d} \bmod p$.

If none of these are congruent to $-1 \bmod p$, then we obtain $a^d \equiv 1 \bmod p$. Otherwise, $-1 \in \{a^d, a^{2d}, a^{2^2 d}, \ldots, a^{2^{s-1}d}\} \bmod p$.

Conversely, if there exists a number a comprime to n $(n \geq 2)$, such that

$$\begin{cases} a^d \not\equiv 1 \pmod n \\ -1 \notin \{a^d, a^{2d}, a^{2^2 d}, \ldots, a^{2^{s-1}d}\} \pmod n \end{cases} \tag{1}$$

then n is not a prime number.

For a composite number n and a positive integer a such that $\gcd(a, n) = 1$, if a satisfies (1) it is termed a "witness" of n. For a prime number n such that none of the first m prime numbers (i.e. $P_m = \{p_1 = 2, p2 = 3, \ldots, p_m\}$) is a witness of n, then n is called a "pseudoprime" of m.

When $m = 8$, $P_8 = \{2, 3, 5, 7, 11, 13, 17, 19\}$. It has been proved (*Jiang and Deng, 2014*) that in the range of $[1, Q_{11}]$, there are only 8 pseudoprimes. Thus, it suffices to test P_8 and exclude those 8 pseudoprimes.

However, for a larger range than $[1, Q_{11}]$, information on pseudoprimes remains unknown. Therefore, there are still risks in directly using P_8.

References

1. *Jiang, Yupeng; Deng, Yingpu (2014). "Strong pseudoprimes to the first eight prime bases". Mathematics of Computation. 83 (290): 2915–2924.*
2. *National Institute of Standards and Technology. (2020). Federal Information Processing Standards Publication 186-5: Digital Signature Standard (DSS). Retrieved from https://doi.org/10.6028/NIST.FIPS.186-5*
3. *Wikipedia contributors. (2024, March 12). Miller–Rabin primality test. In Wikipedia, The Free Encyclopedia. Retrieved 05:16, March 20, 2024, from https://en.wikipedia.org/w/index.php?title=Miller%E2%80%93Rabin_primality_test&oldid=1213361293*