# Encryption and Speed-up of RSA

CAN304 Group-28

Letian Xie Longyang Wang Xinzi Li
Xiaoyue Ma Tianlei Shi

# CONTENT

# Introduction

01

PART ONE

# Introduction



## RSA Algorithm Process:

- Select Two large prime numbers $p$ and $q$
- Calculate the product $n = pq$ with $\varphi(n) = (p - 1)(q - 1)$
- Select a random integer e ($\varphi(n) > e > 1$) with $\gcd(e, \varphi(n)) = 1$
- Calculate $d$ such with $de = 1 \; mod \; \varphi(n)$
- For each key $k = (n, p, q, d, e)$, the encryption transformation is defined as $C = M^e \bmod n$, the decryption transformation is defined as $M = C^d \bmod n$, when $C$ means the cipher text and $M$ means the original message
- $\{e, n\}$ is the public key and $\{d, n\}$ is the private key.

# Problems Define

02

PART ONE

# Problems Define



- Inconvenient to use:
  - Require two large prime numbers
  - Limitations of prime number generation
- Slow:
  - Key length
  - Standardization of the data format

# Large Primes Generation

03

PART ONE

# Large Primes Generation



- According to the prime number theorem, let $\pi(n)$ denote the number of primes less than n, then we have

$$\pi(n) \approx \frac{n}{\ln n} (n \to \infty)$$

- Assume integer r be chosen randomly in the interval $1 \leq r \leq 10^{100}$, and then we can get the following corollaries:

Corollary I: $\Pr(r \text{ is prime}) = \frac{\pi(n)}{n} = \frac{1}{\ln n}$

Corollary II: $\Pr(r \text{ is prime} \mid r \text{ is odd}) = \frac{\pi(n)}{n/2} = \frac{2}{\ln n}$

Corollary III: $\Pr(r \text{ is prime} \mid r \text{ is odd and } \mathrm{GCD}(r, 3) = 1) = \frac{\pi(n)}{n/2 \bullet (1 - 1/3)} = \frac{3}{\ln n}$

# Large Primes Generation



- The probability of generating prime numbers can became no lower than $\frac{3}{\ln n}$ by only generating odd integers and using small primes to verify (trial division), and this be improved at least 200% compared to $\frac{1}{\ln n}$ in previous.

The procedure of large prime generation:

1. randomly generate an integer r in the interval [n/2, n] based on the specified large integer n, and perform r = r × 2 + 1
2. conduct trial division to check r by using all prime numbers less than 100, and set r = r + 2 and repeat step 2 if r failed in checking
3. conduct M-R test. If r failed in the test, set r = r + 2 and return to step 2, and we produce a prime otherwise

# Faster Decryption

04

PART TWO

# Faster Decryption



- Through THE CHINESE REMAINDER THEOREM, the problem of speeding up the decryption time can be well solved.
- THE CHINESE REMAINDER THEOREM :

premise: recipient knows the two prime numbers p and q

If p<q then define A, which 0<A<q-1 and A*p ≡1 mod q

Calculate: dp≡d mod(p-1), dq≡d mod(q-1)

Cp≡C modp, Cq≡C modq

Mp≡C^dp modp, Mq ≡C^dqmodq

To decryption: $\widetilde{M}$ = [((Mq+q- Mp)*A)modq]*p+ Mp

# Pre-processed Plaintext

05

PART TWO

# Pre-processed Plaintext



- Encryption

Step1:
Convert every character of message M to binary representation according to Ascii table (every character must consist of 8 of 0s and 1s) from left to right.

Step2:
Convert whole representation(S) of M to decimal representation $M_1$

Step3:
Apply RSA encrypting function on the number $M_1$ and get number$M_2$

# Pre-processed Plaintext



- Decryption

Step1:
Apply RSA decrypting function on the number $M_2$ and will get number $M_1$

Step2:
Convert $M_1$ to binary representation and ensure the number of digits must divided by 8 (we can add zeros to the left)

Step3:
Divide that representation into bytes (8 digits) from left to right

Step4:
Convert every bytes into character according to Ascii table

# Code Demo

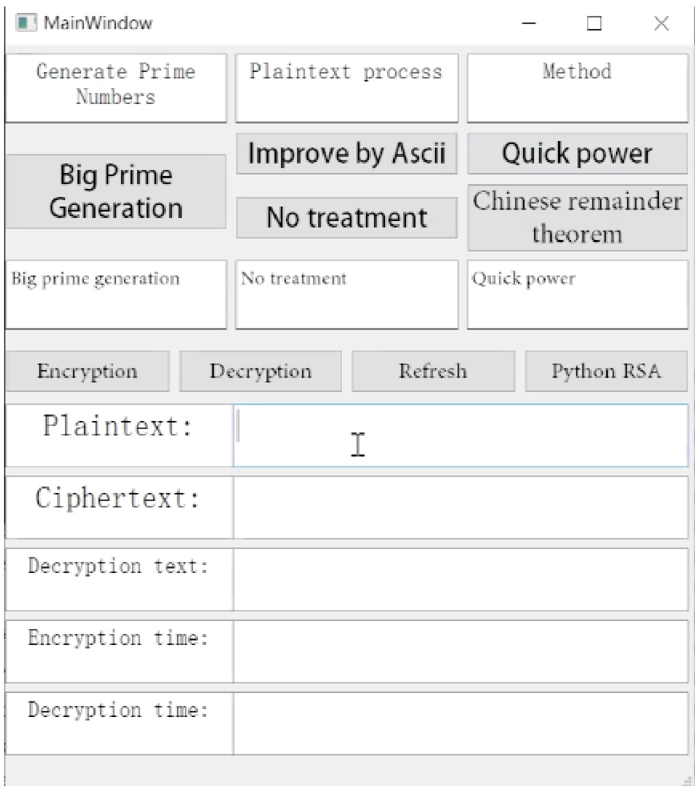
06

PART TWO

# Code Demo



- The code demo is shown as below.




# Experimental Result

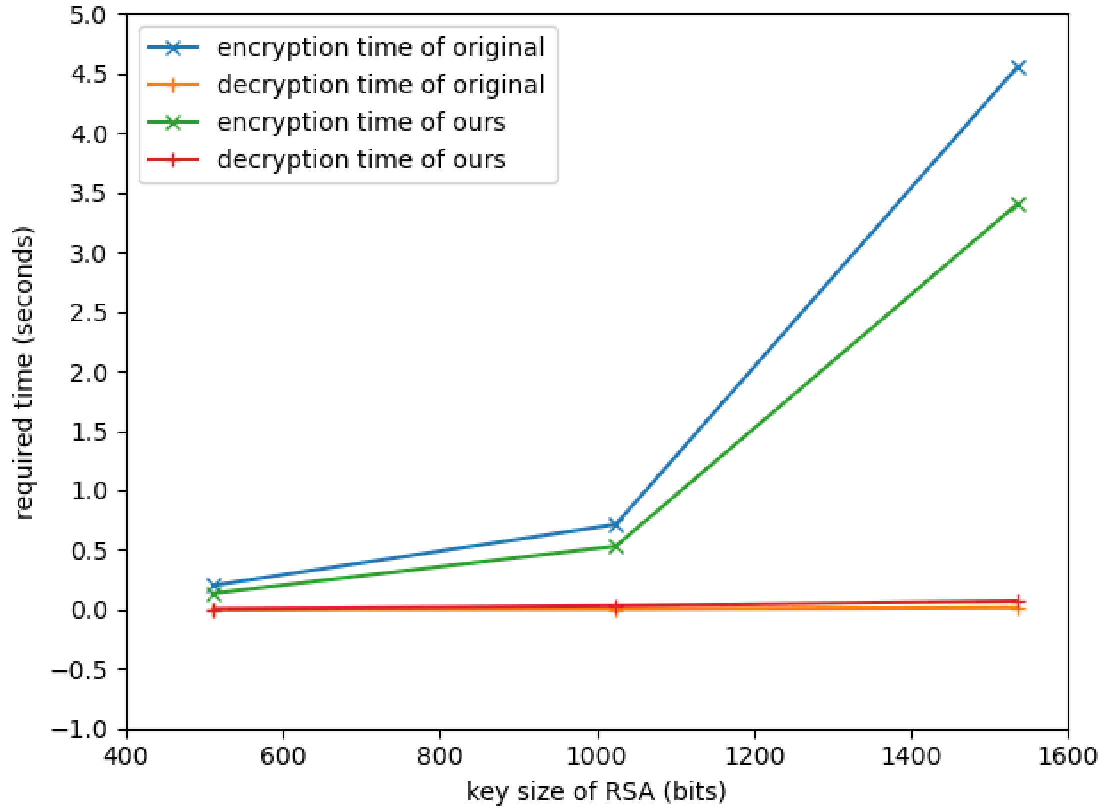
07

PART TWO

# Experimental Result



- The results show that the improved version has significantly improved the encryption speed for key sizes larger than 1000 bits.



# Thank You