STMC HKOI Training

RSA Encrpytion

Chan Yan Mong

December 18, 2021

Goal today

- Learn theory behind RSA encrpytion
- Implement proof-of-concept RSA encryption in Python

Useful mathematical results

Theorem (Chinese Remainder Theorem)

Let $m_1, m_2, \cdots, m_n$ be a set of n relatively prime integers (i.e. $\gcd(m_i, m_j) = 1$ if $i \neq j$) the the following sysmtem of linear congruence:

$$\begin{aligned} x &\equiv a_1 \mod m_1 \\ x &\equiv a_2 \mod m_2 \\ &\cdots \\ x &\equiv a_n \mod m_n \end{aligned}$$

has a unique solution $\mod m_1 m_2 m_3 \cdots m_n$

Useful mathematical results

Corollary

Let $n = pq$ and p, q are distinct primes, then solving

$$x \equiv a \mod n$$

is equivalent as solving

$$x \equiv a \mod p$$
$$x \equiv a \mod q$$

Useful mathematical results

Theorem (Fermat's Little Theorem)

Let p be a prime number and $p \nmid a$, then

$$a^{p-1} \equiv 1 \mod p$$

Corollary (Another form of Fermat's little theorem)

Let a be any integers, then

$$a^p \equiv a \mod p$$

Proof: *If $p \nmid a$, then by the theorem above $a^{p-1} \equiv a \mod p$ and thus $a^p \equiv a \mod p$; Otherwise, $p|a$ and $a^p \equiv a \equiv 0 \mod p$*

Useful mathematical results

We can generalize Fermat's little theorem using the Euler Phi function to obtain the Euler's theorem

Theorem (Euler's Theorem)

Let a be an integer with $\gcd(a, n) = 1$*, then:*

$$a^{\phi(n)} \equiv 1 \mod n$$

where $\phi(n)$ *denotes the Euler phi function*

RSA Encrpytion: Motivation

- We want to work out a way such that

RSA Encryption: Preparation

1. Find two large prime numbers p, q
2. Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$
3. Choose an integer c relatively prime to $\phi(n)$ (i.e. $\gcd(c, \phi(n)) = 1$)
4. Compute the multiplicative inverse of c mod $\phi(n)$. Call it d

$$cd \equiv 1 \mod \phi(n)$$

5. Keep $p, q, d, \phi(n)$ secret while make (n, c) public

RSA Encryption: Encrypt and Decrypt

Now, d is your **private key** and (n, c) is your **public key**. Let M be your **message** and $M < n$ (Otherwise break M into smaller parts that are less than n). To **encrypt** a message, compute:

$$E = M^c \mod n$$

To **decrypt** a message, compute:

$$M = E^d \mod n$$

RSA Encryption: Example with context

Ben has followed the procedure above and computed a public key (n, c) and private key d which he kept secret. Now let's say Amy wants to send him a secret message M. Now instead of sending M unencrypted through the internet, she will first take Ben's private key c and compute:

$$E = M^c \mod n$$

She then send this encrypted message E over the net. Ben, upon recieving E will be able to decrypt her secret message by computing:

$$M = E^d \mod n$$

RSA Encryption: Why it works

We will now prove the *correctness* of the RSA algorithm (i.e. Prove that we can actually decrypt the original message M from E through the procedure)

Proof.

Recall $M < n$ and $n = pq$. So if we consider $\gcd(M, n)$, there are 4 cases:

1. $\gcd(M, n) = 1$
2. $\gcd(M, n) = p$
3. $\gcd(M, n) = q$
4. $\gcd(M, n) = n$

RSA Encryption: Why it works

Proof.

Case 1 $\gcd(M, n) = 1$: Since $cd \equiv 1 \mod \phi(n)$, $cd = 1 + t\phi(n)$ for some integer t. Hence:

$$E^d = M^{cd} = M\left(M^{\phi(n)}\right)^t \equiv M(1) \equiv M \mod n$$

Here we have used Euler Theorem $M^{\phi(n)} \equiv 1 \mod n$ if $\gcd(M, n) = 1$

RSA Encryption: Why it works

Proof.

Case 2 $\gcd(M, n) = p$: We cannot apply Euler's theorem directly. However, we can use Chinese remainder theorem to convert our original congruence from:

$$E^d \equiv M^{cd} \mod n$$

To

$$E^d \equiv M^{cd} \mod p$$
$$E^d \equiv M^{cd} \mod q$$

RSA Encryption: Why it works

Proof.

Now $\gcd(M, n) = p$, so $p|M$ and thus:

$$M^{cd} \equiv M \equiv 0 \mod p$$

On the other hand, $\gcd(M, q) = 1$ and $cd = 1 + t\phi(n) = 1 + t(p-1)(q-1)$, so:

$$M^{cd} \equiv M \left(M^{q-1}\right)^{t(p-1)} \equiv M(1) \equiv M \mod q$$

Here we used the Fermat's little theorem $M^{q-1} \equiv 1 \mod q$ if $\gcd(M, q) = 1$

RSA Encryption: Why it works

Proof.

Case 3 $\gcd(M, n) = q$: Same as Case 2 except p, q swapped
Case 4 $\gcd(M, n) = pq$: In this case $n | M$ so:

$$E^d \equiv M^{cd} \equiv M \equiv 0 \mod n$$

So in all 4 cases we have $E^d \equiv M \mod n$. This means $E^d = M + nt$ for some integer t. But $M < n$, so $t = 0$ and $E^d \mod n = M$ $\square$

RSA Encryption: Finding the primes

- To compute RSA keys, we need to find two large primes p, q
- How can we find these primes?
- The idea is simple: We find large numbers at random, and check if they are primes
- But how to check primes *efficiently*?

Primality test: Trial division

- This is the simplest but also slowest algorithm
- To test the primality of n, we first find all the primes from 2 to $\sqrt{n}$, call it $\{p_i\}$

$$\{p_i\} = \{p \text{ is prime} | 2 \leq p \leq \sqrt{n}\}$$

- We then loop over p_i and check if n is divisible by any of it; If so, it's not a prime, otherwise it is a prime
- We will prove the *correctness* of the algorithm in the next slide

Primality test: Trial division

Theorem (Correctness of Trial division)

Let n be an integer and $\{p_i\}$ be the collection of all primes $\leq \sqrt{n}$, then n is a prime if and only if $p_i \nmid n$ for all $p_i \in \{p_i\}$

Proof.

(Only if): If n is a prime, obviously it is not divisible by p_i. Done.
(If): Suppose not, $p_i \nmid n$ for all $p_i \in \{p_i\}$ but n is composite. Then there exist some prime $q \notin \{p_i\}$ and $q|n$. Now $q > \sqrt{n}$, otherwise it would have been included in $\{p_i\}$. Furthermore, consider $h = n/q$. Obviously $h|n$ and $h < \sqrt{n}$ because $q > \sqrt{n}$. Now no matter if h is composite or prime, there exist some prime number $s|h$ and so $s|n$ but $s < \sqrt{n}$, which is a contradiction. $\square$

Primality test: Fermat's test

- Trial division is slow, so we wish to find something faster
- Here we introduce Fermat test, which can test if a number is **not** prime

Theorem (Fermat Test)

Let a, n be integers and $\gcd(a, n) = 1$, n is composite if

$$a^{n-1} \not\equiv 1 \mod n$$

Proof.

If n is prime, then by Fermat's little theorem $a^{n-1} \equiv 1 \mod n$, which is a contradiction. □

Primality test: Fermat's test

- Note that n passing the Fermat test **does not imply** n is a prime
- For example, $21 = 7 \times 3$ is certainly not a prime and $\gcd(55, 21) = 1$. However,

$$55^{21-1} \equiv 1 \mod 21$$

  so 21 passes the Fermat test to the base 55
- In general, suppose $\gcd(a, n) = 1$ and $a^{n-1} \equiv 1 \mod n$. Then we said n is a **pseudoprime base** a

Primality test: Carmichael numbers

- Worse still, there exist some n such that for any $\gcd(a, n) = 1$, $a^{n-1} \equiv 1 \mod n$
- That is, even if you check all possible (reasonable) a and find out n passes all these Fermat tests, you will still not be able to conclude if n is prime
- These are called **Carmichael number**
- For example, $561 = 3 \times 11 \times 17$ is a smallest carmichael number.

Primality test: Carmichael numbers

Theorem

561 is a Carmichael number

Proof.

Note that $561 = 3 \times 11 \times 17$ *satisfy some interesting properties. Namely:*

$$
\begin{aligned}
(3-1) &| (561-1) \\
(11-1) &| (561-1) \\
(17-1) &| (561-1)
\end{aligned}
$$

Furthermore, since $3, 11, 17$ *are all primes. For any* $\gcd(a, 561) = 1$*, we have* $3 \nmid a$*,* $11 \nmid a$*,* $17 \nmid a$

Primality test: Carmichael numbers

Proof.

Hence by Fermat's little theorem:

$$a^{3-1} \equiv 1 \mod 3$$
$$a^{11-1} \equiv 1 \mod 11$$
$$a^{17-1} \equiv 1 \mod 17$$

Primality test: Carmichael numbers

Proof.

Furthermore, by the since 561 satisfy those interesting properties above, we can raise each of the congruence equation to an integer power so that the exponent all becomes $561 - 1$

$$a^{561-1} \equiv 1 \mod 3$$
$$a^{561-1} \equiv 1 \mod 11$$
$$a^{561-1} \equiv 1 \mod 17$$

Thus, by Chinese remainder theorem: $a^{561-1} \equiv 1561$ for any $\gcd(a, 561) = 1$ $\square$

Primality test: Carmichael numbers

- We can generalize the proof here easily to show the following theorem:

Theorem (Korselt's criterion)

A positive composite integer n is a Carmichael number if and only if n is square-free, and for all prime divisors p of n, it's true that $p - 1 | n - 1$

Proof.

Refer to Theorem 5.3 of Elementary Number Theory by Burton (2011) □

Primality test: Miller-Rabin test

- Since Fermat test have bad properties, we need a better test
- Miller-Rabin test is another stronger test for testing if a number is **not** prime
- Turns out, it avoids the problem of Fermat's test and allow use to test for primes *probabilistically*