Outline

Euler Formula

RSA

Euler Formula

Let integer a coprime to pq.

$$a^{\frac{(p-1)(q-1)}{g}} \equiv 1 \mod pq$$

$g = \gcd(p-1, q-1)$

proof:

$$\mathbb{Z}_{pq}^* \cong \mathbb{Z}_p^* \times \mathbb{Z}_q^*$$

$$a \longmapsto (\underset{\substack{\parallel \\ a_1}}{a \bmod p}, \underset{\substack{\parallel \\ a_2}}{a \bmod q})$$

$$a_1^{\frac{(p-1)(q-1)}{g}} = \left(a_1^{p-1}\right)^{\frac{(q-1)}{g}}$$

$$= 1 \mod p$$

$$a_2^{\frac{(p-1)(q-1)}{g}} = \left(a_2^{q-1}\right)^{\frac{(p-1)}{g}}$$

$$= 1 \mod q$$

$$(a_1, a_2)^{\frac{(p-1)(q-1)}{g}} = (1, 1)$$

$$a^{\frac{(p-1)(q-1)}{g}} = 1$$

Let p be prime.

Let a be an integer coprime to p.

Fermat's : $a^{p-1} \equiv 1 \mod p$

Let p, q be distinct primes.

Let a be an integer coprime to pq.

Euler's 1 : $a^{(p-1)(q-1)} \equiv 1 \mod p$

Euler's 2 : $a^{\frac{(p-1)(q-1)}{g}} \equiv 1 \mod p$

$g = \gcd(p-1, q-1)$

Proof

Euler's 1: $a^{(p-1)(q-1)} \equiv 1 \mod pq$

proof: $\left(a^{p-1}\right)^{q-1} \equiv 1 \mod p \rightarrow p \mid a^{(p-1)(q-1)} - 1$

$\left(a^{q-1}\right)^{p-1} \equiv 1 \mod q \rightarrow q \mid a^{(p-1)(q-1)} - 1$

$pq \mid a^{(p-1)(q-1)} - 1 \rightarrow a^{(p-1)(q-1)} \equiv 1 \mod pq$

Euler's 2: $a^{\frac{(p-1)(q-1)}{g}} \equiv 1 \mod pq$

$g = \gcd(p-1, q-1)$

proof: $\mathbb{Z}_{pq}^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*$

$b = a^{\frac{(p-1)(q-1)}{g}} \longmapsto (a_1, a_2)$

$a_1 = b \mod p = a^{(p-1)\frac{(q-1)}{g}} \mod p = 1$

$a_2 = b \mod q = a^{(q-1)\frac{(p-1)}{g}} \mod q = 1$

$b \longmapsto (1, 1)$

$b = 1$

Diffie-Hellman Exchange: $g^x = h \mod p$, solve x.

RSA problem: $m^e = c \mod N$, solve m.

"Find the e-th root of c modulo N".

Case 1: $N = p$ prime, $\gcd(e, p-1) = 1$ (Easy)

Given $m^e = c \mod p$

Since $\gcd(e, p-1) = 1$, there exist d such that $ed \equiv 1 \mod p-1$.

$$m = m^{ed} = c^d \mod p$$

So, use Extended Euclidean algorithm to find d and compute $m = c^d \mod p$.

Case 2: $N = pq$, p and q are distinct primes

$\gcd(e, (p-1)(q-1)) = 1$ (Easy if $(p-1)(q-1)$ is known)

compute d such that

$ed \equiv 1 \mod (p-1)(q-1)$ using

Extended Euclidean Algorithm and compute

$$m = m^{ed} \equiv c^d \mod pq$$

RSA assumption:

If m is uniformly distributed at random in $\mathbb{Z}_N^*$,

given N, e, c, it is hard to recover m.

(computationally intractable)

Weak Assumption

(1) m may not be uniformly distributed.

(2) partial information can be obtained from m.

If factoring is easy then RSA problem is easy.

The reverse is not known.

Primality test:

Given an integer p, it takes polynomial time to

check if p is prime.

Factoring large integers

Given an integer n, find the prime

factors of n. No polynomial time

algorithm is known.

Shor's Quantum Algorithm

Factoring integer is easy in Quantum

Computer.

RSA cryptosystem (textbook | plain RSA)

Bob	Alice
Key creation	
Choose secret primes p and q. Choose encryption exponent e with $\gcd(e,(p-1)(q-1))=1$. Publish $N=pq$ and e.	
Encryption	
	Choose plaintext m. Use Bob's public key (N,e) to compute $c \equiv m^e \pmod N$. Send ciphertext c to Bob.
Decryption	
Compute d satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. Compute $m' \equiv c^d \pmod N$. Then m' equals the plaintext m.	

Table 3.1: RSA key creation, encryption, and decryption

Correctness: Prove that $m' = c^d \bmod N$ is equal to m

proof: $m' = c^d \bmod N = m^{ed} \bmod N = m \bmod N$

Easy: encrypt: $m^e \bmod N$

decrypt: Bob solves for d using Extended Euclidean algorithm and compute $c^d \bmod N$

Hard: break: Eve knows N, e, c. Based on RSA assumption, it is hard to recover m.

N : modulus
e : encryption exponent } public
d : decryption exponent
p, q : primes } private

RSA in practice

① Efficient Decryption : $C^d \bmod N$ using Euler

$ed \equiv 1 \bmod \frac{(p-1)(q-1)}{g}$, $g = \gcd(p-1, q-1)$

Example :

$p = 229$
$q = 281$
$N = 64349$
$C = 43927$
$e = 17389$

$(p-1)(q-1) = 63840$
$g = \gcd(p-1, q-1) = 4$
$\frac{(p-1)(q-1)}{g} = 15960$

First compute d : $17389 \cdot d \equiv 1 \bmod \cancel{63840}$ 15960

$d = \cancel{53509}$ 5629

Compute $m = 43927^{\cancel{53509}\ 5629} \bmod 64349$

Roughly ~~log 53509~~ time

log 5629 time

② Efficient decryption using CRT

$$m = c^d \bmod pq$$

Using Chinese Remainder theorem.

$$\mathbb{Z}_{pq}^{*} \cong \mathbb{Z}_p^{*} \times \mathbb{Z}_q^{*}$$

$$m = c^d \longrightarrow (m_1, m_2)$$

Compute

$$m_1 = m \bmod p = c^{d \bmod p-1} \bmod p$$

$$m_2 = m \bmod q = c^{d \bmod q-1} \bmod q$$

Compute

$$m = m_1 u p + m_2 v q$$

Where $pu + qv = 1$.

③ Decryption exponent should not be small.

- To avoid brute force attack
- However, large decryption exponent leads to long decryption time

$ed \equiv 1 \mod (p-1)(q-1)$

④ If encryption exponent e is small.

- short encryption time
- security issue :

$\gcd(e, (p-1)(q-1)) = 1.$

The smallest possible $e > 1$ is 3.

(a) When message m is small.

$m^3 < N$

Ciphertext C is m^3 without modulo reduction.

Eve receives C, she just have to compute $C^{\frac{1}{3}}$ over integers.

To find cube root over integers $\mathbb{Z}$, binary search, polynomial time.

(b) Same message m is sent to receivers with public parameter e

$(N_1, 3)$

$(N_2, 3)$

$(N_3, 3)$

$\gcd(N_i, N_j) = 1 \quad i \neq j.$

Eve sees $c_1 = m^3 \bmod N_1$

$c_2 = m^3 \bmod N_2$

$c_3 = m^3 \bmod N_3$

and $N_1, N_2\ N_3$.

Find c using CRT such that

$c \equiv c_1 \bmod N_1$

$c \equiv c_2 \bmod N_2$

$c \equiv c_3 \bmod N_3$

How to recover m from c?

Find c using CRT such that

$$c \equiv c_1 \bmod N_1$$
$$c \equiv c_2 \bmod N_2$$
$$c \equiv c_3 \bmod N_3$$

How to recover m from c?

$$c \equiv m^3 \bmod N_1 N_2 N_3$$

Since $c_i < N_i$, $m^3 < N_1 N_2 N_3$

Solve m by taking cube root of c over integers.

5 common modulus attack.

(N, e_i, d_i) ← common modulus

Leave as exercise!

Plaintext RSA

Plaintext RSA is deteministic. which raises security issue.

For example, suppose Eve just need to know if the message sent to Alice is m.

Eve just need to compute m^e where e is the public encryption exponent of Alice and compare m^e with the ciphertext sent to Alice that E has intercepted.

Padded RSA

Embed message in a random string.