Number Theory, Public Key Cryptography, RSA

Ahmet Burak Can
Hacettepe University
abc@hacettepe.edu.tr

The Euler Phi Function

- Definition:
  For a positive integer n, if 0<a<n and gcd(a,n)=1, a is relatively prime to n.
- Definition:
  Given an integer n, φ(n) is the number of positive integers less than or equal to n and relatively prime to n.

The Euler Phi Function

- Theorem: Formula for φ(n)
  Let p be prime, e, m, n be positive integers
  1) φ(p) = p-1
  2) if gcd(m,n)=1, then φ(mn)=φ(m)φ(n)
  3) $\varphi(p^e) = p^e - p^{e-1}$
  4) If $n = p_1^{e_1} p_2^{e2} \ldots p_k^{ek}$ then

$$\varphi(n) = n(1-\frac{1}{p_1})(1-\frac{1}{p_2})\ldots(1-\frac{1}{p_k})$$

Fermat's Little Theorem

- Fermat's Little Theorem

  If p is a prime number and a is a natural number that is not a multiple of p, then
  $a^{p-1} \equiv 1 \pmod{p}$

Euler's Theorem

- Euler's Theorem
  Given integer n>1, such that gcd(a,n)=1 then
  $a^{\varphi(n)} \equiv 1 \pmod{n}$
- Corollary
  Given integer n>1 such that gcd(a,n)=1, then
  $a^{\varphi(n)-1}$ mod n is a multiplicative inverse of a mod n.

Consequence of Euler's Theorem

- Principle of Modular Exponentiation
  Given integer n>1, x, y, and a positive integers with gcd(a,n)=1. If x≡y(mod φ(n)), then
  $a^x \equiv a^y \pmod{n}$
- Proof idea:
  $a^x = a^{k\varphi(n)+y} = a^y(a^{\varphi(n)})^k$
  by applying Euler's theorem we obtain
  $a^x \equiv a^y \pmod{n}$

Diffie-Hellman Key Exchange

- Diffie-Hellman proposed a cryptographic protocol to exchange keys among two parties in1976.
  - Public parameters:
    p: A large prime
    g: Base (generator)
  - Secret parameters:
    $\alpha, \beta \in \{0, 1, 2, \ldots, p-2\}$

Alice — Bob

$g^{\alpha} \bmod p$ →

← $g^{\beta} \bmod p$

Alice computes $(g^{\beta})^{\alpha} \bmod p$ — Bob computes $(g^{\alpha})^{\beta} \bmod p$

$K = g^{\alpha\beta} \bmod p$

Security of Diffie-Hellman

- Discrete Logarithm Problem (DLP):
  - Given p, g, $g^{\alpha} \bmod p$, what is α?
  - easy in Z, hard in Z_p
- Diffie-Hellman Problem (DHP):
  - Given p, g, $g^{\alpha} \bmod p$, $g^{\beta} \bmod p$, what is $g^{\alpha\beta} \bmod p$?
- DHP is as hard as DLP.

Commutative Encryption

- Definition:
  An encryption scheme is commutative if
  $E_{K1}[E_{K2}[M]] = E_{K2}[E_{K1}[M]]$

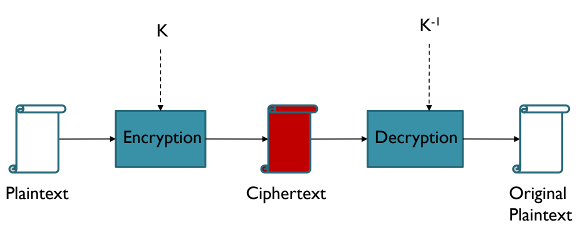
  Given a commutative encryption scheme, then
  $D_{K1}[D_{K2}[E_{K1}[E_{K2}[M]]] = M$
- Most symmetric encryption scheme are not commutative such as DES and AES.

Pohlig-Hellman Exponentiation Cipher

- A commutative exponentiation cipher
  - encryption key (e, p), where p is a prime
  - decryption key (d, p), where $ed \equiv 1 \pmod{p-1}$ or in other words $d \equiv e^{-1} \pmod{p-1}$
  - to encrypt M, compute $C = M^e \bmod p$
  - to decrypt C, compute $M = C^d \bmod p = M^{ed} \bmod p$

Asymmetric Encryption Functions

- An asymmetric encryption function:
  - Encryption (K) and decryption (K^{-1}) keys are different.
  - Knowledge of the encryption key is not sufficient for deriving the decryption key efficiently.
  - Hence, the encryption key can be made "public".



Public Key Encryption

- Each party has a PAIR (K, K^{-1}) of keys:
  - K is the public key
  - K^{-1} is the private key

$$D_{K^{-1}}[E_K[M]]=M$$

- The public-key K may be made publicly available.
- Many can encrypt with the public key, only one can decrypt.
- Knowing the public-key and the cipher, it is computationally infeasible to compute the private key.

Solutions with Public Key Cryptography

- Key distribution solution:
  - Alice makes her encryption key K public
  - Everyone can send her an encrypted message: $C = E_K(P)$
  - Only Alice can decrypt it with the private key K^{-1}: $P = D_K^{-1}(C)$
- Source Authentication Solution:
  - Only Alice can "sign" a message, using K^{-1}.
  - Anyone can verify the signature, using K.
  - Only if such a function could be found...

RSA Algorithm

- Invented in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman
  - Published as R L Rivest, A Shamir, L Adleman, "On Digital Signatures and Public Key Cryptosystems", Communications of the ACM, vol 21 no 2, pp120-126, Feb 1978
- Security relies on the difficulty of factoring large composite numbers
- Essentially the same algorithm was discovered in 1973 by Clifford Cocks, who works for the British intelligence

RSA Public Key Crypto System

- Choose large primes p, q
  - Compute $n = pq$ and $\varphi(n) = (q-1)(p-1)$
- Choose e, such that $\gcd(e, \varphi(n)) = 1$.
  - Take e to be a prime
- Compute $d \equiv e^{-1} \bmod \varphi(n)$, such that $ed \equiv 1 \bmod \varphi(n)$
  - Public key: n, e
  - Private key: d
- Encryption: $C = E(M) = M^e \bmod n$
  Decryption: $D(C) = C^d \bmod n$

RSA Encryption

- Encryption: $C = E(M) = M^e \bmod n$,
- Decryption: $D(C) = C^d \bmod n$.
- Why does it work?

$$\begin{aligned} D(C) &= (M^e)^d \bmod n = M^{ed} \bmod n \\ &= M^{k\varphi(n)+1} \bmod n, \text{ for some } k \\ &= (M^{\varphi(n)})^k M \bmod n \\ &= M \end{aligned}$$

- RSA problem: Given n, e, $M^e \bmod n$, what is M?
  - Computing d is equivalent to factoring n.
  - The security is based on difficulty of factoring large integers.

RSA Example

- Let p = 11, q = 7, then
  - n = 77, $\varphi(n) = 60$
- Let e = 37, then
  - d = 13 (ed = 481; ed mod 60 = 1)
- Let M = 15, then $C \equiv M^e \bmod n$
  $C \equiv 15^{37} \pmod{77} = 71$
- $M \equiv C^d \bmod n$
  $M \equiv 71^{13} \pmod{77} = 15$

RSA Implementation

- The security of RSA depends on how large n is, which is often measured in the number of bits for n.
  - Current recommendation is 1024 bits for n.
- p and q should have the same bit length, so for 1024 bits RSA, p and q should be about 512 bits.
- p-q should not be small.
  - In general, p, q randomly selected and then tested for primality
  - Many implementations use the Rabin-Miller test, (probabilistic test)