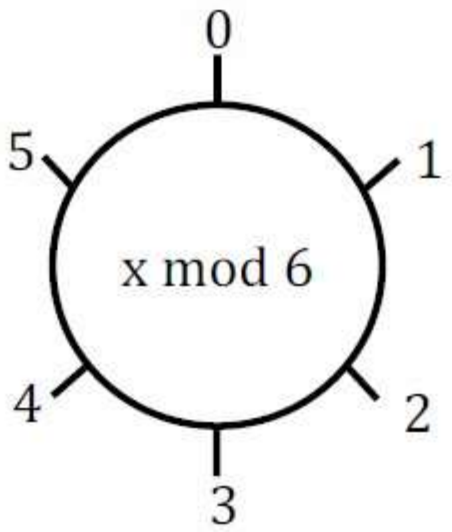
Encryption Algorithms & Protocols

Public Key Cryptography

Dr. Omar Abusada
E-mail: abossada1@gmail.com

Clock Arithmetic

- For integers x and n, the value of x modulo n, which is abbreviated x mod n, is defined to be the remainder when x is divided by n. Note that the remainder when a number is divided by n must be one of the values 0 , 1 , 2 . . . , n-1
- For example, the mod 6 clock appears below
  - 7 mod 6 = 1
  - 33 mod 5 = 3
  - 33 mod 6 = 3
  - 17 mod 6 = 5
  - 17 mod 5 = 2



Modular Addition

- Notation and Facts
  - 7 mod 6 = 13 mod 6 = 1 mod 6 = 1
    - ((a mod n)+(b mod n)) mod n =(a + b) mod n
    - ((3 mod 6) +(5 mod 6)) mod 6
    - (3 + 5) mod 6
    - 8 mod 6 = 2

(3 +5) mod 6

8 mod 6=2

- Examples
  - 3 + 5 = 2 mod 6
  - 3 + 3 = 0 mod 6
  - (7 + 12) mod 6 = 19 mod 6 = 1
  - (7 + 12) mod 6 = (1 + 0) mod 6 = 1

Modular Multiplication

- ((a mod n) (b mod n)) mod n = a×b mod n
- Examples:
  - 3 . 4 = 0 mod 6
  - 2 . 4 = 2 mod 6
  - 5 . 5 = 1 mod 6
  - (7 . 4) mod 6 = 28 mod 6 = 4 mod 6
  - (7 . 4) mod 6 = (1 . 4) mod 6 = 4 mod 6

Modular Inverse

- Additive inverse of x mod n, is the number that must be added to x to get 0 mod n.
  - -2 mod 6 = 4
  - -14 mod 6 = 4
  - -26 mod 6 = 4
  - -17 mod 6 = 1
- Multiplicative inverse of x mod n, is the number that must be multiplied by x to get 1 mod n.
  - 3^{-1} mod 7 = 5 =(5x3)=1
  - 2^{-1} mod 7 = 4 =(4x2)=1
  - 5^{-1} mod 9 = 2 =(5x2)=1
  - 3^{-1} mod 8 = 3 =(3x3)=1
  - 4^{-1} mod 9 = 7 =(4x7)=1
  - 2^{-1} mod 6 = 1 =(?x2)=1 ?? (Modular Inverse do not exist).
- X^{-1} mod n exist when x and n are relatively prime.
- X and n are relatively prime if they have no common factor other than 1.

Public Key Cryptography

- Public key crypto is sometimes know as "asymmetric cryptography"
- In symmetric key cryptography, the same key is used to both encrypt and decrypt the data.
- In public key cryptography, one key is used to encrypt and a different key is used to decrypt
- Two keys
  - Sender uses recipient's public key to encrypt
  - Recipient uses private key to decrypt

Public Key Cryptography

- Based on “trap door one-way function”
- The "trap door" feature ensures that an attacker cannot use the public information to recover the private information.
  - “One-way” means easy to compute in one direction, but hard to compute in other direction
  - Example: Given p and q, product N = pq easy to compute, but given N, it’s hard to find p and q
  - “Trap door” used to create key pairs.

Public Key Cryptography

- To do public key crypto, Bob must have a ***key pair*** consisting of a ***public key*** and a corresponding ***private key***.
- Anyone can use Bob's public key to encrypt a message intended for Bob's eyes only, but only Bob can decrypt the message, since, by assumption only Bob has his private key.
- Bob can also apply his ***digital signature*** to a message ***M*** by "encrypting“ it with his private key.

Knapsack Problem

- The knapsack problem can be stated as follows. Given a set of n weights labelled as:
  - $W_0, W_1, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; W_{n-1}$
- The desired sum S, find $a_0,\ a_1, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ a_{n-1}, where\ each\ a_i \in \{0,1\}\ so\ that$
  - $S = a_0W_0 + a_1W_1 + \ldots\ldots\ldots a_{n-1}W_{n-1}$

  EX:
- provided this is possible. For example, suppose the weights are:

  85,13,9,7,47,27,99,86
- and the desired sum is S = 172. Then a solution to the problem exists is given by
  - since 85 + 13 + 47 + 27 = 172.
  - $a = a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7 = (11001100)$
  - **1**∗85 + **1**∗13 +**0**∗9+**0**∗7+**1**∗ 47 +**1**∗ 27+**0**∗99+**0**∗86 = 172.

Knapsack Problem

- The (general) knapsack is NP-complete.
- General knapsack (GK) is hard to solve.
- But super-increasing knapsack (SIK) is easy.
- A ***super-increasing knapsack*** is similar to the general knapsack except that, when the weights are arranged from least to greatest, each weight is greater than sum of all previous weights. For example:

3,6,11,25,46,95,200,411

is a super-increasing knapsack. Solving a super-increasing knapsack problem is easy.

Knapsack Problem

3,6,11,25,46,95,200,411

- Suppose we are given the set of above given weights and the desired sum S = 309.
- To solve this, we simply start with the largest weight and work toward the smallest to recover the a_i in linear time.
- Since S < 411, we have $a_7 = 0$. Then since S > 200, we must have $a_6 = 1$, since the sum of all remaining weights is less than 200.
- Then we compute S = S — 200 = 109 and this is our new target sum.
- Since S > 95, we have $a_5 = 1$ and we compute S = 109—95 = 14. Continuing in this manner, we find a = 10100110, which we can easily verify solves the problem since 3 + 11 + 95 + 200 = 309.

Knapsack Cryptosystem

1. Generate super-increasing knapsack (SIK).
2. Convert SIK into “general” knapsack (GK).
3. Public Key: GK.
4. Private Key: SIK plus conversion factor.

- Ideally…
  - Easy to encrypt with GK (Public key).
  - With private key, easy to decrypt (convert ciphertext to SIK problem).
  - Without private key, must solve GK.

Knapsack Cryptosystem

- Start with (2,3,7,14,30,57,120,251) as the SIK
- Choose m = 41 and n = 491 (*multiplier*, *m* and *modulus* *n* relatively prime, *n* exceeds sum of elements in SIK)
- Compute "general" knapsack as: GK= W x m mod n
- 2 x m = 2 × 41 mod 491 = 82
- 3 x m = 3 × 41 mod 491 = 123
- 7 x m = 7 × 41 mod 491 = 287
- 14 x m = 14 × 41 mod 491 = 83
- 30 x m = 30 × 41 mod 491 = 248
- 57 x m = 57 × 41 mod 491 = 373
- 120 x m = 120 × 41 mod 491 = 10
- 251 x m = 251 × 41 mod 491 = 471

Knapsack Cryptosystem

- "General" knapsack (GK) : (82,123,287,83,248,373,10,471)
- The public key is the general knapsack, Public key: (82,123,287,83,248,373,10,471).
- The private key is the super-increasing knapsack together with the multiplicative inverse of the conversion factor, i.e., $m^{-1}\ mod\ n$. For this example, we have:

  Private key: (2,3,7,14,30,57,120,251) and $\mathbf{41^{-1}}\ \boldsymbol{mod}\ \mathbf{491 = 12}$.

  Public key (GK) : (82,123,287,83,248,373,10,471).

Knapsack Cryptosystem

- Suppose Bob's public and private key pair are those given in previous slide respectively. Suppose that Alice wants to encrypt the message (in binary) M = 10010110 for Bob. Then she uses the 1 bits in her message to select the elements of the general knapsack that are summed to give the ciphertext. In this case, Alice computes
- Ciphertext = 82 + 83 + 373 + 10 = 548.
- To decrypt this ciphertext, Bob uses his private key to find

$$c * m^{-1} \, mod \, n = 548 * 12 \, mod \, 491 = 6576 \, mod \, 491 = 193.$$

Obtain plaintext 10010110

Private key: (2,3,7,14,30,57,120,251)

… Thank you …