"RSA"

Knapsack Cryptosystem

Input: $\{a_1, a_2, \ldots, a_n\}$, S

Output: $X = (x_1, x_2, \ldots, x_n)$ where $x_i \in \{0, 1\}$

⇒ Sum of Subset Problem

↓

We know that it is Hard for large n

→ But if we take $\{a_1, a_2, \ldots, a_n\}$ is a super-increasing sequence then it is simpler to solve

Super Increasing Sequence

$$a_2 > a_1$$
$$a_3 > a_1 + a_2$$
$$a_4 > a_1 + a_2 + a_3$$
$$\vdots$$
$$a_n > a_1 + a_2 + a_3 + \ldots + a_{n-1}$$

→ Algo: I/P: $\{a_1, \ldots a_n\}$ in Super Increasing order

Sum value: S

O/P: $\{x_1, \ldots x_n\}$ where $x_i \in \{0, 1\}$

```
for i = n down to 1
{
    if (S >= a_i)
    {
        x_i = 1
        S = S - a_i
    }
    else
        x_i = 0
}
```

O(n)

If S == 0 then

$X = (x_1, x_2, \ldots, x_n)$ is the soln

Else

No solution

Note: a_i is in Super-Increasing order.

→ This algo is useful for Decryption, as it is easy i.e. O(n) is Time Complexity.

→ Super Increasing sequence.

Ex: I/P: $\{7, 15, 25, 50\} = a$

$S = 65$

$n = 4$

$i = 4$ $\quad 65 \geq a_4 = 50$

$\therefore \boxed{x_4 = 1}$

$S = S - a_i$

$= S - a_4$

$= 65 - 50$

$\therefore \boxed{S = 15}$

$i = 3$ $\quad 15 \neq 25$

$\therefore \boxed{x_3 = 0}$

$i = 2$ $\quad 15 \geq 15$

$S = S - a_2$

$= 15 - 15$

$= 0$

$\boxed{x_2 = 1}$

$i = 1$ $\quad 0 \neq a_1 = 7 \Rightarrow \boxed{x_1 = 0}$

$S = 0$

$\Rightarrow X = (x_1, x_2, x_3, x_4)$

$= (0, 1, 0, 1)$ is the solution

Note: If array a is public then Alice can encrypt plaintext message $X = (x_1, x_2, \ldots, x_n)$

bits $x_i \leftarrow \{0, 1\}$

by $S = \sum_{i=1}^{n} a_i \cdot x_i$

→ Bob can use S and array a to get the X i.e. plaintext bits

↓ will not work practically!

why?

Eve can also use S, a(Public key) to get X. So, we need more than this.

"RSA"

Date Page

Key generation (By Bob)

(1) Say $a' = (a'_1, a'_2, \dots, a'_n)$ is Super-Increasing Sequence

i.e. $a'_2 \geq a'_1$

$a'_3 \geq a'_1 + a'_2$

$\vdots$

$a'_n \geq a'_1 + a'_2 + \dots + a'_{n-1}$

(2) Select $m \geq \sum_{i=1}^{n} a'_i$

(3) Select w such that $\gcd(m, w) = 1$

(4) Now Compute $a = (a_1, a_2, \dots, a_n)$ where

$$a_i = (w * a'_i) \text{ MOD } m$$

This step breaks the Super Increasing nature of a'_i and will create a_i which is not Super-Increasing

(5) → Public Key $a = (a_1, a_2, \dots a_n)$

→ Private Key $a' = (a'_1, a'_2, \dots, a'_n)$

m

w

★ Encryption by Alice

① Use public key a and Encrypt Message X

$$X = (x_1, x_2, \ldots, x_n)$$

Encode message X in to bits

$x_i \in \{0, 1\}$

② $$S = \sum_{i=1}^{n} a_i x_i$$

↑

Encrypted Output

★ Decryption by Bob (using Private key

① Multiply S by $\vec{w}^{-1}$ with respect to Modulo m

$$\vec{w}^{-1} S \bmod m$$

$$= \vec{w}^{-1} \sum_{i=1}^{n} a_i x_i \bmod m$$

$$= (\vec{w}^{-1} a_1 x_1 + \vec{w}^{-1} a_2 x_2 + \ldots + \vec{w}^{-1} a_n x_n) \bmod m$$

We know that

$a_i = (w * a_i') \bmod m$

(From key Generation)

$\therefore \bar{w}^1 a_i \equiv a_i' \bmod m$

$(\because \gcd(w, m) = 1)$

Thus $\left.\begin{aligned} \bar{w}^1 a_1 &\equiv a_1' \\ \bar{w}^1 a_2 &\equiv a_2' \\ &\vdots \\ \bar{w}^1 a_n &\equiv a_n' \end{aligned}\right\}$ with respect to MOD m

$\therefore \bar{w}^1 s \bmod m$ can be written as

$(a_1' x_1 + a_2' x_2 + \ldots + a_n' x_n) \text{ MOD } m$

$\therefore \bar{w}^1 s \bmod m = \sum_{i=1}^{n} a_i' x_i \text{ MOD } m$

Bob knows $\bar{w}^1 s \bmod m$ i.e. He can find it from s and w, m

Bob knows a' (Private to Bob)

w's known
x_i Not known
m known
a_i' known

a_i' known and Important is It is Super-Increasing

⇓

We need to find x_i using the algorithm discussed i.e. in $O(n)$ time.

Note: This can be done in $O(n)$ time only by Bob but not by Eve.

Why?

⇓

→ Eve doesn't know Private key i.e. a', w, m

→ Eve knows 'a' only but It is not Super-Increasing.

∴ Eve requires exponential time to break the Cipher.

This is what we wanted.

Same Ex: I/P: $a' = \{7, 15, 25, 50\}$

(1) Key generation

$$m \geq \sum_{i=1}^{n} a_i'$$

$$= 7 + 15 + 25 + 50$$

$$= 97$$

$\therefore \boxed{m \geq 97}$

(2) Select $m = 97$

(3) Select w such that $\gcd(m, w) = 1$

Let's take $w = 3$

(4) Compute (a_1, a_2, a_3, a_4)

$$a_1 = (w * a_1') \bmod m$$

$$= (3 * 7) \bmod 97$$

$$= 21$$

$$a_2 = (3 * 15) \bmod 97$$

$$= 45$$

$$a_3 = (3 * 25) \bmod 97$$

$$= 75$$

$$a_4 = (3 * 50) \bmod 97$$

$$= 150 \bmod 97$$

$$= 53$$

(5) Public key $a = (21, 45, 75, 53)$

$\vec{a}$ is not super Increasing

$(\because 53 \not\geq 21 + 45 + 75)$

Private key: $a' = \{7, 15, 25, 50\}$

$m = 97$

$w = 3$

✱ Encryption by Alice :-

Say $\vec{x} = (1, 0, 1, 0)$; Message

Encrypt $\vec{x}$ using a (Public key)

$\vec{x} = (1, 0, 1, 0)$

$\vec{a} = (21, 45, 75, 53)$

$$S = \sum_{i=1}^{4} a_i x_i$$

$= 21 \times 1 + 45 \times 0 + 75 \times 1 + 53 \times 0$

$= 21 + 75$

$\therefore \boxed{S = 96}$

☆ Decryption by Bob :-

① $\bar{w}^1 S \bmod m$

$= (\bar{3}^1 * 96) \bmod 97$

$\bar{3}^1 \text{ Mod } 97$

q	r_1	r_2	r	t_1	t_2	t
32	97	3	1	0	1	−32
3	3	1	0	1	−32	97
	1	0		−32	97	

$\therefore \bar{3}^1 \text{ Mod } 97 = -32 + 97$

$= \boxed{65}$

$\therefore \bar{w}^1 S \text{ Mod } m$

$= (65 * 96) \text{ Mod } 97$

$= 6240 \text{ Mod } 97$

$= 32$

② Now apply O(n) algo. for 32 using a' as private key.

I/P: $a^1 = \{7, 15, 25, 50\}$

w^1's MOD M = 32 (Here S=32 as required by algo.)

Algo

$i=4$ $\quad 32 < 50$

$\therefore \boxed{x_4 = 0}$

$i=3$ $\quad 32 \geq 25$

$\therefore \boxed{x_3 = 1}$

$S = S - a^1_3$

$= 32 - 25$

$= 7$

$i=2$ $\quad 7 < 15$

$\therefore \boxed{x_2 = 0}$

$i=1$ $\quad 7 \geq 7$

$\therefore \boxed{x_1 = 1}$

$S = S - a^1_1$

$= 7 - 7$

$= 0$

$\therefore$ There is soln $X = (x_1, x_2, x_3, x_4) = (1, 0, 1, 0)$ which is the desired answer.