Introdução

O sistema de enciptação Merkel-Hellman knapsack foi um dos primeiros sistemas de encriptação de chave publica a ser inventados sendo criado por Ralph Merkle and Martin Hellman em 1978.
No entanto apesar do metodo ser mais simples que as normas establecidas anteriormente, foram rapidamente encontradas falhas no algoritmo.

O metodo de encriptação deste algoritmo é assimétrico pelo que gera duas chaves, uma **publica** e outra **privada**.

Ex:

> O Tiburcio e Gertruncia querem partilhar um documento confidencial e para isso vão usar o sistema de encriptação Merkle and Hellman.
>
> O sistema de encriptação vai gerar a cada um, um par de chaves, uma privada que não devem partilhar e outra publico para proceder á encriptação.
> O documento vai ser enviado da Gertruncia para o Tiburcio, então para isso , primeiro a Gertruncia terá que pedir a chave publica do Tiburcio.
>
> Após receber a chave a Gertruncia terá que fazer uma copia do ficheiro e encriptar essa copia através do algoritmo de Merkel-Hellman. Esta cópia é necessária pois após ser feita a encriptação com a chave publica do Tiburcio só este a pode desencriptar o documento com a sua chave privada.
>
> No final ela terá de enviar o documento ao Tiburcio que á chegada, ele usará a sua chave privada para proceder á desencriptação.

Encriptação de uma palavra de n bits

Neste trabalho vamos explorar uma verão simplificada deste algoritmo onde o problema de knapsack será substituido por um problema de somas de um subconjunto.

Geração de chaves

A chave privada consiste em 3 elementos:

> **Super sequencia gerada aleatóriamente.**
>
> $$W = \{w_1, w_2, w_3, \ \dots, w_n\}$$
>
> $$\text{onde}$$
>
> $$w_{i+1} > \sum_{i=0}^{n} w_i$$
>
> **Constante *m* aleatória**
>
> $$m > \sum_{i=0}^{n} wi$$
>
> **Constante *a* aleatória coprima de *m***
>
> $$mdc(m, a) = 1$$

A chave publica **P** que consiste num array de inteiros

> $$P = \{p_1, p_2, p_3, \ \dots, p_n\}$$
>
> $$\text{que obtemos ordenando}$$
>
> $$W = \{w_1, w_2, w_3, \ \dots, w_n\}$$
>
> $$\text{onde}$$
>
> $$w'_i = a * w_i \bmod m$$

Encriptação

Dado um array **B** de **n** bits o resultado da encriptação será um numero inteiro **C'** tal que:

$$C' = \sum_{i=0}^{n} b[i] * p[i]$$

Desencriptação

Primeiro temos de descobrir **C**

$$C = C' a^{-1} \bmod m$$

Depois temos que encontrar o indice **idx** apratir do qual os elementos de **W** são maiores que **C**

```
for(int i=n-1;i>0;i--){
    if(w[i]<C){
        idx = i;
        break;
    }
}
```

A seguir temos por cada combinação de arrays de bits possiveis com lenght **idx** fazer o seuguinte:

```
...
for(int i=0;i<=idx;i++){
    S += b_[i]*w[i];
}
...
```

Se S for igual a C então b = b_

(Penso que seja assim noentanto esta parte ñ nos importa pois o objetivo do trabalho é chegar ao array de bytes original sem a chave).