



O Sistema Merkle-Hellman Knapsack

Bernardo Rodrigues César Silva
a79008@alunos.uminho.pt a77518@alunos.uminho.pt

Maria Francisca Fernandes
a72450@alunos.uminho.pt

Universidade do Minho — 3 de Maio de 2019

Resumo

Este documento apresenta os vários passos e considerações feitas para implementação do sistema em questão. Assim como, alguns factos relativos a este.

Conteúdo

1	Introdução	2
2	Implementação	3
2.1	Geração de Permutações	3
2.2	Geração de um sequência super crescente aleatória	3
2.3	Geração de Coprimos	4
2.4	Geração da Chave	4
2.4.1	Versão Multi-Iterada	4
2.5	Encriptação	4
2.6	Decriptação	4
2.6.1	Versão Multi-Iterada	4
3	Conclusões	5
A	Código??	6

Capítulo 1

Introdução

Ao contrario do RSA nao da para fazer assinaturas criptograficas - wiki 1976 Diffie Hellman introduzem a ideia de criptografia de chave publica Este trabalho foi desenvolvido no ambito da Unidade Curricular de *Teoria de Números Computacional*. De entre as escolhas possiveis, foi escolhido estudar o sistema *Merkle-Hellman Knapsack*.

Este foi um dos pioneiros da criptografia de chave pública, inventado por **Ralph Merkle** e por **Martin Hellman** em 1978. A ideia por detrás deste sistema é mais simples do que a de sistemas como o *RSA*, assentando no problema (tendo já sido quebrado – meter isto noutra sitio??).

Capítulo 2

Implementação

Ao longo das secções deste capítulo apresentamos os vários algoritmos seguidos para a codificação do sistema.

2.1 Geração de Permutações

Um dos passos da Geração da Chave Pública(criar ref) consiste em gerar uma permutação de uma sequência. Para tal utilizamos o algoritmo proposto por **Sandra Sattolo**. Este itera uma lista - *seq* - a partir do ultimo índice desta - n . Em cada passo calculamos um índice aleatório - j - tal que $1 \leq j < n$ e de seguida trocamos os valores de seq_j e seq_n e (explicar resto). Continuamos assim até que $n = 1$.

A implementação deste pode ser visualizada em criar ref.

O algoritmo pode ser visto aqui. criar ref.

2.2 Geração de um sequência super crescente aleatória

Um dos componentes da Chave Privada é uma sequência, esta é considerada super crescente se:

Definição 1. Consideremos uma sequência de números b_1, \dots, b_n . Esta diz-se super crescente se:

$$b_i > \sum_{j=1}^{i-1} b_j \text{ para cada } i \text{ tal que } 2 \leq i \leq n.$$

Como tal, conseguimos deduzir:

$$b_1 + b_2 + \dots + b_k < 2 \times b_k$$

Ou seja, precisamos apenas de considerar o último valor gerado para calcular um possível próximo. Usaremos este facto para gerar a nossa sequência super crescente.

Com isto apresentamos o nosso algoritmo.

Algorithm 1 Geração da sequência super crescente aleatória

Recebe: n - o tamanho da sequência

Devolve: $\{x_1, \dots, x_n\}$ - uma sequência super crescente aleatória

- 1: k um limite superior aleatoriamente grande
 - 2: f uma função que satisfaz criar ref para a dedução
 - 3: $x_1 \leftarrow j$ tal que $1 \leq j \leq k$ aleatório
 - 4: **for** x_i com $i := 2$ **até** n **do**
 - 5: $x_i = f(x_{i-1})$
 - 6: **end for**
-

2.3 Geração de Coprimos

2.4 Geração da Chave

2.4.1 Versão Multi-Iterada

2.5 Encriptação

2.6 Decriptação

2.6.1 Versão Multi-Iterada

Algorithm 2 Solução da soma do subconjunto super crescente

```
1:  $i \leftarrow n$ 
2: while  $i \geq 1$  do
3:   if  $s \geq b_i$  then
4:      $x_i \leftarrow 1$ 
5:   else
6:      $x_i \leftarrow 0$ 
7:   end if
8:    $i \leftarrow i - 1$ 
9: end while
```

Capítulo 3

Conclusões

O RSA é melhor, passados 4 anos da sua criação o Knapsack foi quebrado.
Falar em quebrar uma mensagem em pedaços?? Criticar : fácil de quebrar.
Trabalho futuro: implementação de um quebra.

Apêndice A

Código??

Será que vai num ficheiro separadamente?