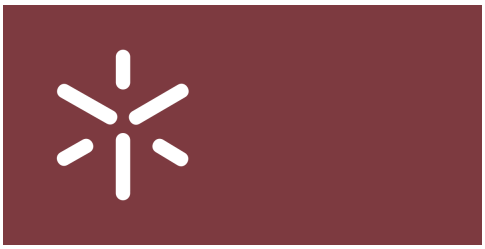

O Sistema Merkle-Hellman Knapsack

Bernardo Rodrigues
a79008@alunos.uminho.pt

César Silva
a77518@alunos.uminho.pt

Maria Francisca Fernandes
a72450@alunos.uminho.pt

Universidade do Minho — 2 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

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 noutro 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 implementacão 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, um sequência $b_1, ..., b_n$ é considerada super crescente se:

$$b_i > \sum_{j=1}^{i-1} b_j$$

para cada i tal que $2 \leq i \leq n$.

Como tal, conseguimos deduzir:

$$b_1 + b_2 + ... + b_j < 2 \times b_j$$

2.3 Geração de Coprimos

2.4 Geração da Chave

2.4.1 Versão Multi-Iterada

2.5 Encriptacão

2.6 Decriptação

2.6.1 Versão Multi-Iterada

Algorithm 1 Solução da soma do subconjunto super crescente

```
1: i ← n
2: while i ≥ 1 do
3:     if s ≥ b_i then
4:         x_i ← 1
5:     else
6:         x_i ← 0
7:     end if
8:     i ← i − 1
9: end while
```

Capítulo 3

Conclusões

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

Apêndice A

Código??

Será que vai num ficheiro separadamente?