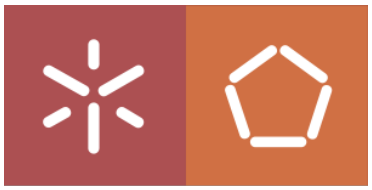


Universidade do Minho
Escola de Engenharia

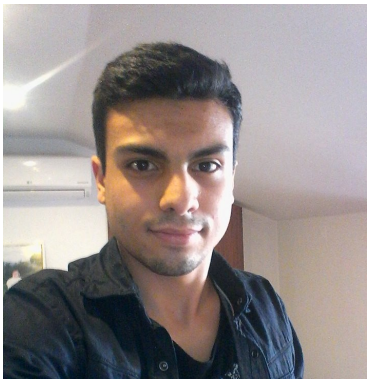
Universidade do Minho
Mestrado Integrado em Engenharia Informática

Tecnologia Criptográfica

Trabalho prático 5

27 Dezembro 2020

Ana Margarida Campos
(A85166)



Nuno Pereira
(PG42846)

Contents

Introdução

1.1 Contextualização

O presente relatório foi elaborado no âmbito do quinto Trabalho Prático da Unidade Curricular de Tecnologia Criptográfica, que se insere no 1º semestre do 4º ano do Mestrado Integrado em Engenharia Informática (1º ano MEI).

1.2 Objetivos e Trabalho Proposto

Este trabalho prático encontra-se dividido em duas partes. A primeira parte do trabalho consistiu na implementação do teorema chinês do resto de modo a resolver os dois sistemas de congruências modulares do enunciado. A segunda parte do trabalho consiste na decifragem de um criptograma gerado a partir da cifra RSA.

1.3 Estrutura do Relatório

O relatório encontra-se dividido em duas partes (conforme o trabalho prático). Na primeira parte é dada uma pequena introdução do teorema chinês do resto seguida da descrição detalhada da estratégia utilizada para a resolução dos dois sistemas. A segunda parte inicia-se com uma breve introdução sobre a cifra RSA e, posteriormente, é apresentada a estratégia usada para a decifragem do criptograma fornecido.

Primeira Parte

2.1 Teorema Chinês do Resto

O Teorema Chinês do Resto define que um sistema de congruências lineares, admite uma solução simultânea referente ao produto dos módulos calculados no sistema. Ou seja:

Sejam n_1, n_2, n_3, ..., n_k números inteiros positivos tais que gcd(n_i, n_j) = 1, para i $\neq$ j (números primos entre si). O sistema de congruência lineares

$x \equiv a_1 \pmod{n_1}$
$x \equiv a_2 \pmod{n_2}$
$x \equiv a_3 \pmod{n_3}$
. . .
$x \equiv a_k \pmod{n_k}$

admite uma solução simultânea, que é única módulo o inteiro $n = n_1 n_2 n_3 ... n_k$. gcd representa o máximo divisor comum.

2.2 Estratégia de Resolução e Programa Desenvolvido

De modo a resolver os sistemas de congruências lineares fornecidos foi implementado um programa em *python*. Este programa baseia-se essencialmente na seguinte estratégia de resolução: Considerando como base o sistema de congruências lineares:
$x \equiv a_1 \pmod{n_1}$
$x \equiv a_2 \pmod{n_2}$
$x \equiv a_3 \pmod{n_3}$
É necessário efetuar os seguintes cálculos para chegar ao valor final de x:

$N = n_1 \times n_2 \times n_3$

a_i	N_i	x_i	$a_i \times N_i \times x_i$
a_1	$n_2 n_3$	x_1	$a_1 N_1 x_1$
a_2	$n_1 n_3$	x_2	$a_2 N_2 x_2$
a_3	$n_1 n_2$	x_3	$a_3 N_3 x_3$

x_i é obtido calculando inicialmente a_i mod n_i seguido de encontrar um valor para o qual multiplicado com valor resultante do módulo, o resultado seja congruente com 1.

$$x = \sum_{3}^{i=1} b_i \times N_i \times x_i \pmod{N}$$

No caso das equações no formato $c_i\ x \equiv a_i$ *(mod* n_i*)* é preciso recorrer ao algoritmo de Euclides estendido de modo a calcular a inversa modular. Após recorrer a este algoritmo, é possível reescrever a equação no formato anterior pelo que depois é utilizada a mesma estratégia para resolver este tipo de sistemas.

O programa foi implementado de acordo com os seguintes passos:

1. São passados como parâmetros 3 tipos de arrays: um que indica os valores de a_i, outro com os valores de n_i e um último, sendo de carácter opcional, que coloca os valores de c_i caso o sistema tenha equações do modo $c_i\ x \equiv a_i\ (mod\ n_i)$.
2. É verificado se o último parâmetro é passado como *None*. Caso não seja, é necessário recorrer ao algoritmo de Euclides estendido de modo a tornar as equações no formato $x \equiv a_i \pmod{n_i}$.
3. É verificado se o $\gcd(n_i,\ n_j) = 1$, ou seja, é verificado se os valores são primos entre si. Se for verdade então prossegue para o cálculo do sistema. Caso contrário é retornado um erro.
4. De modo a começar a fase de cálculos e seguindo a estratégia enunciada em cima, é primeiramente cálculado o valor *N*. Posteriormente se existerem mais do que duas equações, é colocado num novo array os valores de *Ni*. Caso apenas existam duas equações (como é no caso do segundo sistema) os valores de n_i são invertidos e colocados no array diretamente, sem ser necessária nenhuma multiplicação. É feito um ciclo *while* de modo a calcular os valores de x_i.
5. Por último, ocorre a multiplicação de $a_i N_i x_i$ e a soma das mesmas para todo o i. É feito o módulo e retornado o valor final de x.

Os resultados obtidos para os dois sistemas propostos foram: $x = 7302$ para o primeiro sistema e $x = 55$ para o segundo sistema.

Segunda Parte

3.1 RSA

RSA é um sistema criptográfico de chave pública. Neste sistema a chave para cifrar uma mensagem é pública, o que significa que pode ser conhecida por todos, e existe uma diferente chave para decifrar que é privada. Toda a mensagem cifrada com uma determinada chave pública só pode ser decifrada usando a respetiva chave privada.

Uma variante insegura do RSA é o *Textbook RSA*. Este é um método criptográfico determinístico e por isso não é seguro contra *chosen plaintext attacks* ou *ciphertext attacks*. Neste método é necessário encontrar dois números primos (p e q) tais que $n = p \cdot q$. A chave pública, e, é escolhida garantindo que $gcd(e,(p\text{-}1) \cdot (q\text{-}1)) = 1$. No processo de escolha de uma chave privada, d, tem de se ter atenção a que $e \cdot d \equiv 1 \bmod ((p\text{-}1) \cdot (q\text{-}1))$. Ou seja, d é a inversa modular de e, $d \equiv e^{-1} \bmod ((p\text{-}1) \cdot (q\text{-}1))$. Os processos de cifragem e decifragem são os seguintes:

- **Cifragem:** para cifrar uma mensagem m, o criptograma c é gerado a partir de $c \equiv m^e \bmod n$.
- **Decifragem:** para decifrar um criptograma c, a mensagem m é gerada a partir de $m \equiv c^d \bmod n$.

3.2 Estratégia e Programa Desenvolvido

O programa implementado tem como principal objetivo decifrar o criptograma fornecido no enunciado deste trabalho prático. Para chegar à mensagem de texto limpo foram seguidos dois passos:

1. De modo a calcular a chave privada, foi inicialmente necessário fatorizar o número $n = 213271$ para, como resultado, ter os dois números primos necessários para o processo, p e q (419 e 509 respetivamente). Posteriormente, foi utilizado o algoritmo de Euclides estendido de modo a calcular a inversa modular de e (chave privada) a partir da chave pública e e de $(p\text{-}1) \cdot (q\text{-}1)$. Como resultado obtemos a chave privada $d = 74945$.
2. Como já sabemos a chave privada foi necessário calcular os valores da mensagem para cada um dos valores do criptograma. Para tal foi utilizada a fórmula de decifragem $m \equiv c^d \bmod n$. Após este cálculo foi necessário recorrer ao polinómio $27L_1^2 + 27L_2 + L_3$ de modo a encontrar as 3 letras ou espaços que correspondiam àquele valor de decifragem. A estratégia utilizada passou por ter uma lista com todas as letras do alfabeto e o espaço em que os índices correspondem ao valor da letra, ou seja, A = 0 até Z = 25 e o espaço com o valor de 26. Depois experimentar todos valores possíveis das letras (ou seja os valores dos índices) e calcular o polinómio até ao valor esperado. No final é imprimido o texto limpo num ficheiro.

O resultado obtido foi o seguinte texto limpo:

```
LET US THEREFORE PERMIT THESE NEW HYPOTHESES TO BECOME KNOWN TOGETHER WITH THE
ANCIENT HYPOTHESES WHICH ARE NO MORE PROBABLE LET US DO SO ESPECIALLY BECAUSE
THE NEW HYPOTHESES ARE ADMIRABLE AND ALSO SIMPLE AND BRING WITH THEM A HUGE
TREASURY OF VERY SKILLFUL OBSERVATIONS SO FAR AS HYPOTHESES ARE CONCERNED LET NO
ONE EXPECT ANYTHING CERTAIN FROM ASTRONOMY WHICH CANNOT FURNISH IT LEST HE ACCEPT
AS THE TRUTH IDEAS CONCEIVED FOR ANOTHER PURPOSE AND DEPART FROM THIS STUDY A
GREATER FOOL THAN WHEN HE ENTERED IT FAREW
```

Conclusão

Concluíndo, com a elaboração deste trabalho prático foi possível aplicar conhecimentos adquiridos nas aulas desta unidade curricular. Permitiu-nos pôr em prática o teorema chinês do resto resolvendo dois sistemas de congruências lineares e também nos permitiu verificar como a variante *textbook* do RSA é insegura uma vez que foi relativamente fácil de decifrar o criptograma fornecido.