Segurança Informática



**(imagem de capa!!)**

# Índice

[Índice 2](#_Toc526532404)

[Questão 1 3](#_Toc526532405)

[Questão 2 3](#_Toc526532406)

[Questão 3 4](#_Toc526532407)

[3.1 4](#_Toc526532408)

[3.2 4](#_Toc526532409)

[Questão 4 5](#_Toc526532410)

[Questão 5 - MD5 Collision Attack Lab 6](#_Toc526532411)

[2.1 6](#_Toc526532412)

[2.2 8](#_Toc526532413)

[2.3 8](#_Toc526532414)

[Questão 6 8](#_Toc526532415)

[Questão 7 8](#_Toc526532416)

## Questão 1

Se o atacante conseguir que o resultado da passagem de duas mensagens distintas (x e x’) na função de hash seja igual, então este estaria na condição de conseguir forjar uma assinatura digital fazendo-se passar por outra pessoa.

## Questão 2

Se T é um esquema MAC logo:

* ***m*** passará por um canal inseguro e será possível interceptar este valor.

Ao aplicar E já sabemos qual é:

* a sua chave: ***T(k1)(m)***
* e a sua mensagem: ***m*** (devido a esta passar em T no seu canal inseguro)

Com estes dois valores conhecendo algoritmo **E** conseguimos decifrar o seu resultado.

Desta forma estando na posse de ***T(k1)(m)*** e ***Es(T(k1)(m)1::L)(m)*** e fazendo a sua concatenação bit a bit obtém-se o criptograma CI(m). Por este meio prova-se que CI não é fiável para ser um criptograma de autenticidade visto que é possível ser quebrado.

## Questão 3

### 3.1



Figura 1

### 3.2

1. **o que é o texto em claro?**
2. Neste modo de operação é possível realizar paralelização de trabalho na operação de cifragem visto que para cifrar o bloco seguinte não é necessário utilizar um resultado previamente obtido. No extremo oposto a este caso está o modo de operação CBC que para cifra um bloco *“i”* necessita conhecer o resultado da cifra do bloco *“i-1”* para realizar o *XOR* com o resultado do bloco Ep(k). Caso i=0 então o valor com o qual será realizado o *XOR* será o de um Vetor Inicial (IV).

## Questão 4

## Questão 5 - MD5 Collision Attack Lab

### 2.1

#### Questão 1 - If the length of your prefix file is not multiple of 64, what is going to happen?

Dimensão prefixo: 44 bytes (ficheiro em anexo “Q1prefix.txt”)

Dimensão de ficheiros de saída: 192 bytes (ficheiros em anexo “Q1out1.txt” e “Q1out2.txt”).

Hash utilizado obtido através do comando “*md5sum Q1out1.txt”*: 61459646d70cdcb913c9eb4b5a380215

Utilizando um ficheiro não múltiplo de 64 podemos observar, na Figura 2 e Figura 3, que o algoritmo através do mecanismo de *padding* adiciona zeros ao bloco a cifrar.

Na Figura 2 e Figura 3 podemos ver o resultado da comparação dos dois ficheiros resultantes da experiência. Como referido acima é possível visualizar os zeros bem como quais os bytes que são diferentes.

De salientar que os ficheiros de saída dão sempre maiores que o ficheiro de saída.

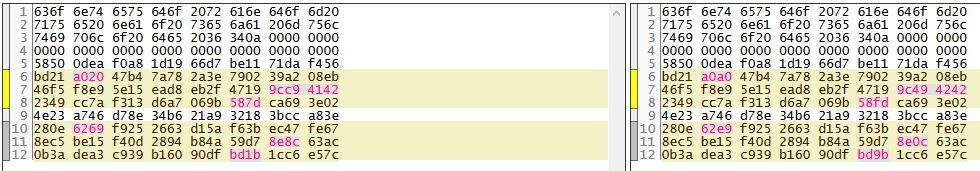


Figura 2 – ex 1 output 1

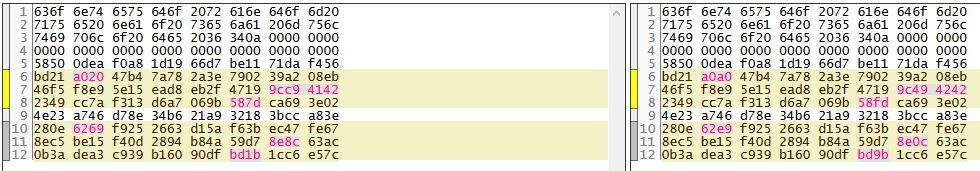


Figura 3 – ex 1 output 2

#### Questão 2 - Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what

#### happens.

Dimensão prefixo: 64 bytes (ficheiro em anexo “Q2prefix64.txt”)

Dimensão de ficheiros de saída: 192 bytes (ficheiros em anexo “Q2out1.txt” e “Q2out2.txt”).

Hash utilizado obtido através do comando “*md5sum out1.txt”*: 8e587cc6f8b6a188ed9535f11bbe829d

Utilizando desta vez um ficheiro diferente, mas com exactamente 64 bytes ao analisar a

Figura 4 e Figura 5 verificamos logo à partida que não existem manchas com bytes a zero, o que significa que não houve necessidade de adicionar *padding* ao output dos ficheiros.

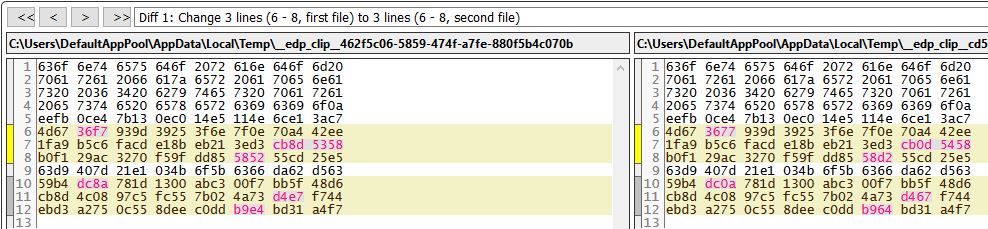


Figura 4 – ex 2 output 1

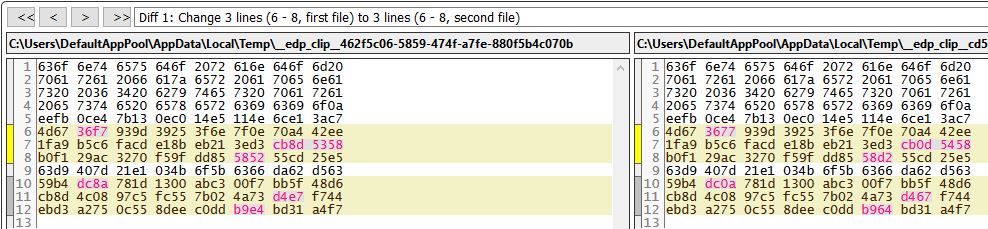


Figura 5 – ex 2 output 1

#### Questão 3 - Are the data (128 bytes) generated by md5collgen completely different for the two

#### output files? Please identify all the bytes that are different.

Analisando as figuras das questões 1 e 2 é percetível identificar as diferenças entre os ficheiros de saída do algoritmo. Podemos observar que só existem 7 diferenças nos ficheiros e estas são sempre no mesmo byte.

### 2.2

### 2.3

## Questão 6

## Questão 7