



# **Criptografia e Segurança em Redes**

ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA

**2023/2024**

## **Trabalho Prático 3**

**Experiências de criptografia**

David Alexandre Baptista Oliveira – [a86732@alunos.uminho.pt](mailto:a86732@alunos.uminho.pt)

José Pedro Fernandes Peleja – [a84436@alunos.uminho.pt](mailto:a84436@alunos.uminho.pt)

Miguel Fernandes Pereira – [a94152@alunos.uminho.pt](mailto:a94152@alunos.uminho.pt)

## Índice

1. Introdução.....	4
2. Experiências .....	5
2.1. Função de sentido único .....	5
2.2. Cifra por blocos .....	6
3. Testes e resultados .....	8
3.1. Função de sentido único .....	8
3.2. Cifra por blocos .....	8
Conclusão .....	11

## Índice de figuras

Figura 1 - Fluxograma experiência 1. ....	5
Figura 2 - Fluxograma encriptação simplificada. ....	7
Figura 3 - Fluxograma descriptação simplificado.....	7
Figura 4 - Resultados da experiência 1. ....	8
Figura 5 - Ficheiro de texto "teste".....	8
Figura 6 - Teste 1 da experiência 3, cifragem do ficheiro “teste”.....	8
Figura 7 - Teste 1 da experiência 3, decifragem do ficheiro “resultado1”.....	8
Figura 8 - Ficheiro de texto "output1".....	9
Figura 9 - Teste 2 da experiência 3, decifragem com a chave errada.....	9
Figura 10 - Ficheiro de texto "output2".....	9
Figura 11 - Teste 3 da experiência 3, nº de argumentos errado. ....	9
Figura 12 - Teste 4 da experiência 3, operação inválida.....	9
Figura 13 - Teste 5 da experiência 3, ficheiro inexistente. ....	9
Figura 14 - Ficheiro "resultado1".....	10
Figura 15 - Ficheiro "resultado1" modificado. ....	10
Figura 16 - Teste 6 da experiência 3, ficheiro cifrado modificado.....	10
Figura 17 - Ficheiro "output3".....	10

## 1. Introdução

O trabalho prático proposto consiste na escolha de duas experiências sugeridas pelo professor. Serve o presente relatório para explicar as decisões e escolhas do grupo, resultados alcançados e testes efetuados assim como as várias dificuldades encontradas durante o processo.

O trabalho começa por escolher duas experiências, ao que o grupo escolheu a experiência 1 por parecer uma experiência interessante e que despertou curiosidade dos mesmos. Esta consiste na descoberta de uma password, que se encontra entre as 200 mais utilizadas, sabendo previamente o seu *hash*.

Escolhemos também a experiência 3, pois a implementação de uma aplicação de linha de comando para cifrar e decifrar ficheiros usando AES-128-GCM é uma tarefa prática que demonstra a aplicação direta de conceitos criptográficos em situações do mundo real, sendo ainda um modo cada vez mais utilizado devido ao seu desempenho.

De modo a sermos capazes de cumprir com os objetivos deste trabalho prático é necessário aplicar os conhecimentos obtidos ao longo do semestre na UC e com isso consolidar os mesmos através de experiências práticas que recorrem a diferentes sistemas criptográficos.

## 2. Experiências

Na presente secção do relatório estão presentes as decisões e os passos tomados para a implementação e desenvolvimento das duas experiências escolhidas pelo grupo.

### 2.1. Função de sentido único

Nesta experiência, é-nos fornecida uma *hash* que faz parte das 200 passwords mais utilizadas em todo o mundo. Esta é armazenada em hexadecimal, e o serviço utiliza o SHA256 para gerar a *hash*. A mesma é representada da seguinte forma “96cae35ce8a9b0244178bf28e4966c2ce1b8385723a96a6b838858cdd6ca0a1e”. Para a sua descoberta começamos por analisar as passwords mais comuns.

De seguida, com o objetivo de analisar e gerar *hashes* para todas as palavras-passe, criámos uma lista abrangendo todas as opções disponíveis. Após, escolhemos aleatoriamente uma delas para criar o *hash* e comparar com o fornecido associado à password que queremos descobrir. Com o intuito de otimizar e automatizar esse processo, implementamos um ciclo que percorre a lista de maneira aleatória, calculando o *hash*, usando o SHA256, de cada palavra-passe até encontrar aquela cujo *hash* seja igual ao fornecido, como é possível observar no seguinte fluxograma.

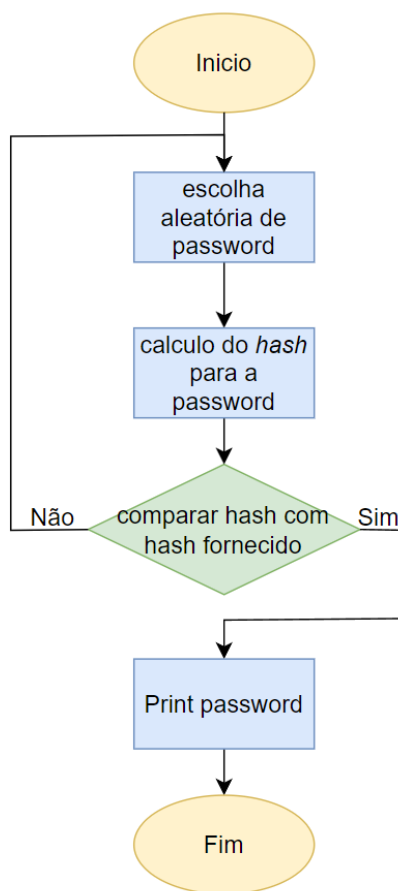


Figura 1 - Fluxograma experiência 1.

A decisão de ser de forma aleatória e não sequencial é se a lista fosse imensamente maior e a password fosse a última, demoraria muito mais tempo a encontrar, com a aleatoriedade este tempo é reduzido.

A password descoberta foi a “123123”, sendo esta a 13.<sup>a</sup> mais utilizada no mundo com um tempo de descoberta abaixo de 1 segundo.

Por fim, introduzimos um *timer* para que pudéssemos avaliar o tempo necessário para encontrar a palavra-passe, proporcionando uma análise mais detalhada do desempenho do processo.

## 2.2. Cifra por blocos

O algoritmo AES (Advanced Encryption Standard) é uma das duas cifras simétricas selecionadas para os novos protocolos de transporte, nomeadamente o TLSv1.3. Deve ser implementado com uma chave de 128 bits e no modo de operação GCM (AES-128-GCM). O modo de operação GCM combina a criptografia simétrica (AES) com autenticação de mensagem. Ele usa uma técnica chamada Galois/Counter Mode para fornecer confidencialidade e integridade dos dados.

O GCM produz uma sequência de bytes que engloba o Vetor de Inicialização (IV), o texto cifrado e uma tag de autenticação. Este relatório propõe o desenvolvimento de uma aplicação de linha de comando que utiliza o AES-128-GCM para cifrar e decifrar um ficheiro. Os parâmetros essenciais, como a operação (cifrar ou decifrar), chave, ficheiro de entrada e ficheiro de saída, são fornecidos como argumentos da linha de comando, seguindo o formato “aes -operação chave input\_file output\_file”.

Para a implementação das operações de cifragem e decifragem utilizando o AES-128-GCM, adotamos a biblioteca Cryptography, conforme recomendação do professor. Por meio desta biblioteca, desenvolvemos funções dedicadas à encriptação e desencriptação, possibilitando a manipulação segura de dados e informações contidas em ficheiros. Para obter esses dados, é necessário abrir os ficheiros correspondentes e efetuar a leitura dos seus conteúdos e de seguida escrever num ficheiro novo.

Durante o processo de encriptação, um valor de IV é gerado de forma aleatória para garantir mais segurança. Após a encriptação dos dados, estes são registados num novo ficheiro, juntamente com o IV utilizado e a tag de autenticação. Na fase de desencriptação, é necessário ler o IV armazenado no ficheiro, a tag de autenticação e o texto encriptado para realizar a desencriptação. A tag de autenticação é utilizada para verificar se o ficheiro não sofreu alterações ou se a chave utilizada é a correta. De seguida, estão presentes os fluxogramas do processo de encriptação e de desencriptação.

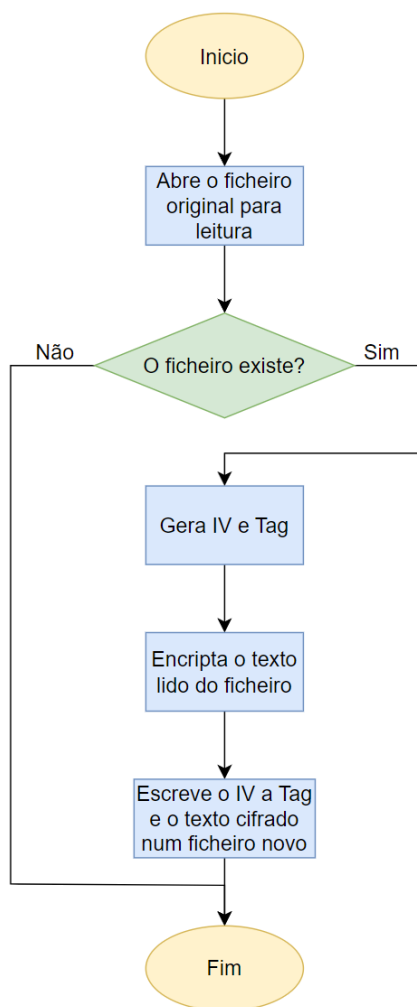


Figura 2 - Fluxograma encriptação simplificada.

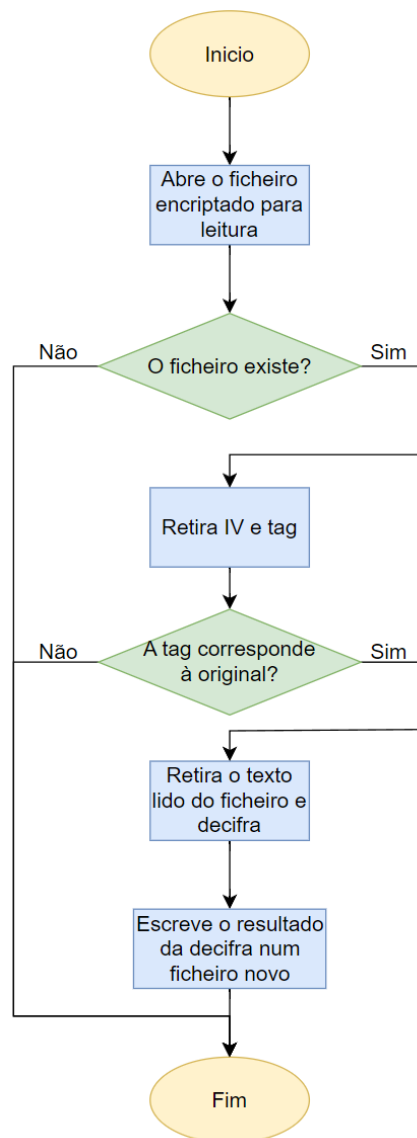


Figura 3 - Fluxograma desencriptação simplificado.

Conforme indicado pelo modo AES-128-GCM, é essencial utilizar uma chave de 128 bits. Para ser possível utilizar uma chave com tamanho variável, optamos por aplicar um processo de *hash* à password do utilizador, garantindo que a chave resultante seja sempre de dimensão fixa, independentemente da password fornecida. A chave utilizada é o *hash* resultante desse cálculo.

Embora tenhamos escolhido o MD5 devido ao seu retorno de *hash* de 128 bits, para obter sempre o mesmo do tamanho da chave e o tamanho utilizado pelo AESGCM, é importante destacar as limitações associadas a este algoritmo. O MD5 possui vulnerabilidades conhecidas, especialmente em relação à sua resistência a colisões. Este fator pode resultar em questões de segurança, uma vez que permite a existência de diferentes *inputs* que geram o mesmo *hash*, comprometendo a integridade do sistema. Portanto, embora o MD5 atenda ao requisito de tamanho de chave, é importante reconhecer e avaliar as suas limitações de segurança ao considerar a sua implementação.

### 3. Testes e resultados

Nesta secção são apresentados os resultados dos testes realizados. Estes foram realizados no computador pessoal com o auxílio do editor de código VSCode.

#### 3.1. Função de sentido único

Executando o primeiro programa na consola é possível visualizar que foi encontrada a palavra-passe “123123” e a duração da pesquisa. Notamos que a obtenção da senha varia devido à aleatoriedade, podendo ser tão rápida que o "timer" quase se torna instantâneo. Além disso, destaca-se a eficácia e rapidez dessa operação, revelando a vulnerabilidade das *passwords*.

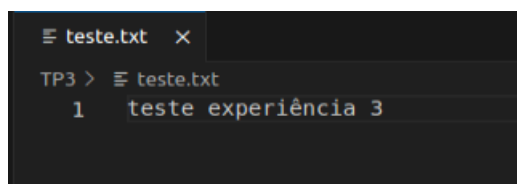
```
PS C:\Users\Pedro Peleja\Desktop\cripto\tp3> python3 exp1.py
Password found: 123123
It took about 0.0 seconds
PS C:\Users\Pedro Peleja\Desktop\cripto\tp3> python3 exp1.py
Password found: 123123
It took about 0.0005085468292236328 seconds
```

Figura 4 - Resultados da experiência 1.

#### 3.2. Cifra por blocos

Para a terceira experiência foram realizados alguns testes, de modo a verificar as funcionalidades básicas, ocorrência de erros e casos especiais.

Para o primeiro teste ciframos um ficheiro de texto com uma chave, e com a mesma deciframos o ficheiro obtido garantindo a funcionalidade básica do programa.



```
teste.txt x
TP3 > teste.txt
1 teste experiência 3
```

Figura 5 - Ficheiro de texto "teste".

Para a cifragem do ficheiro de texto “teste” foi utilizado o seguinte comando, tendo como parâmetros o tipo de operação, a chave, o nome do ficheiro a ser cifrado e o ficheiro de output.

```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py cifra chave teste.txt resultado1
Encryption complete. Output written to resultado1
```

Figura 6 - Teste 1 da experiência 3, cifragem do ficheiro “teste”.

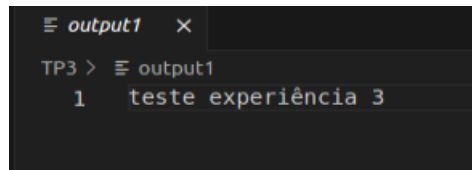
Para a decifragem do ficheiro “resultado1” foi utilizado o comando seguinte:

```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py decifra chave resultado1 output1
Decryption complete. Output written to output1
```

Figura 7 - Teste 1 da experiência 3, decifragem do ficheiro “resultado1”.



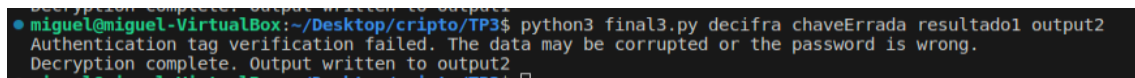
É possível verificar que o ficheiro “output1” contém o conteúdo original, figura 8, tendo sido completado o teste 1 sem quaisquer erros de cifragem e decifragem.



```
TP3 > cat output1
1  teste experiência 3
```

Figura 8 - Ficheiro de texto "output1".

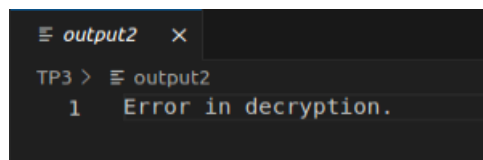
Num segundo teste foi tentado decifrar o mesmo ficheiro “resultado1” com uma chave distinta, resultando numa mensagem de aviso de que os dados podem ter sido corrompidos ou que a chave utilizada está errada, originando o ficheiro “output2”.



```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py decifra chaveErrada resultado1 output2
Authentication tag verification failed. The data may be corrupted or the password is wrong.
Decryption complete. Output written to output2
```

Figura 9 - Teste 2 da experiência 3, decifragem com a chave errada.

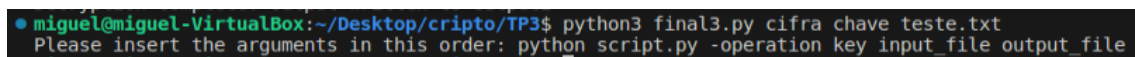
No ficheiro “output2” é possível verificar a mensagem escrita após o erro de decifra, como visível na figura 10.



```
TP3 > cat output2
1  Error in decryption.
```

Figura 10 - Ficheiro de texto "output2".

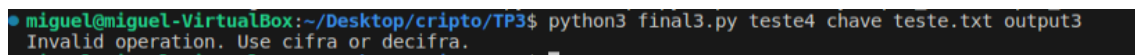
Para o terceiro teste iremos correr o comando de arranque com o número de argumentos errado, sendo mostrado uma mensagem de aviso, e o utilizador é então alertado de que deverá correr o comando com a designada formatação.



```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py cifra chave teste.txt
Please insert the arguments in this order: python script.py -operation key input_file output_file
```

Figura 11 - Teste 3 da experiência 3, nº de argumentos errado.

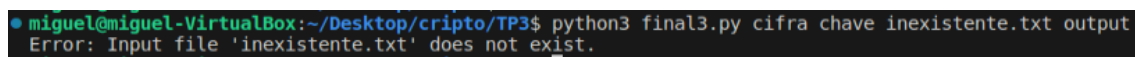
Para o quarto teste escolhemos o modo de operação errado, sendo mostrado uma mensagem de erro a fim de avisar o utilizador a escolher um dos modos de operação válidos.



```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py teste4 chave teste.txt output3
Invalid operation. Use cifra or decifra.
```

Figura 12 - Teste 4 da experiência 3, operação inválida.

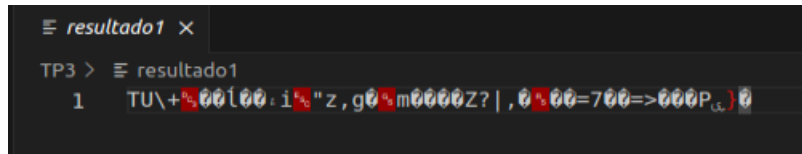
Para o quinto teste foi tentado cifrar um ficheiro inexistente, ocorrendo um erro onde é mostrado uma mensagem de aviso de que o ficheiro de input não existe.



```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py cifra chave inexistente.txt output
Error: Input file 'inexistente.txt' does not exist.
```

Figura 13 - Teste 5 da experiência 3, ficheiro inexistente.

No último teste deciframos o ficheiro “resultado 1”, figura 14, previamente cifrado no teste 1, onde os dados foram alterados, figura 15.



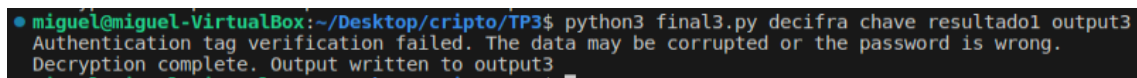
```
≡ resultado1 ×
TP3 > ≡ resultado1
1 TU\+%00{00.i%"z,g0%m0000Z?|,0%00=700=>000P(,)
```

Figura 14 - Ficheiro "resultado1".



```
≡ resultado1 ×
TP3 > ≡ resultado1
1 TU\+%00{00.i%"z,g0%mZ?|,0%00=700=>00
```

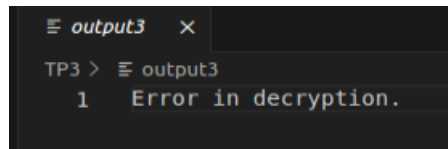
Figura 15 - Ficheiro "resultado1" modificado.



```
miguel@miguel-VirtualBox:~/Desktop/cripto/TP3$ python3 final3.py decifra chave resultado1 output3
Authentication tag verification failed. The data may be corrupted or the password is wrong.
Decryption complete. Output written to output3
```

Figura 16 - Teste 6 da experiência 3, ficheiro cifrado modificado.

É possível verificar que ao decifrar um ficheiro modificado obtemos um erro, visível na figura 16 acima, sendo avisado de que os dados do ficheiro podem ter sido corrompidos, e foi gerado um ficheiro “output3” com uma mensagem de erro.



```
≡ output3 ×
TP3 > ≡ output3
1 Error in decryption.
```

Figura 17 - Ficheiro "output3".

## Conclusão

Com o concluir deste trabalho prático consideramos que obtivemos e consolidamos conhecimentos básicos de criptografia e segurança em redes, apesar das dificuldades encontradas. O grupo conseguiu com sucesso concluir as duas experiências que se propôs a realizar, sendo estas a experiência de descoberta de password e a de encriptação de ficheiros usando AES-128-GCM.

Na experiência de descoberta de palavra-passe, o grupo desenvolveu duas soluções distintas, optando por manter aquela que se mostrou mais lógica e simples. Inicialmente, enfrentamos a dificuldade de determinar como poderíamos pesquisar pelas *passwords* mais frequentemente utilizadas. No entanto, conseguimos encontrar uma solução simples que nos permitiu progredir rapidamente no trabalho.

Por outro lado, na experiência 3 só surgiu a dificuldade do tamanho da chave que foi solucionado com a utilização de um hash no lugar da chave, para que esta tivesse tamanho fixo. Esta solução envolveu a aplicação de conhecimentos adquiridos na unidade curricular num contexto diferente, uma vez que tínhamos utilizado o hash anteriormente apenas para garantir a integridade dos dados e não como chave de encriptação.

Em suma, consideramos que este trabalho prático permitiu aprofundar os conhecimentos previamente adquiridos na unidade curricular, refletindo um desempenho satisfatório por parte do grupo.