TP0-Problema3

March 7, 2022

1 TRABALHO PRÁTICO 0 - GRUPO 14

1.1 Problema 3

O problema 3 consiste em comparar os algoritmos apresentados no problema 1 e 2.

1.1.1 Resolução do Problema

Imports

[ ]:
```
import os,timeit
```

Medição dos tempos de execução Para medir o tempo de execução e a eficiência dos esquemas de cifra dos problemas 1 e 2, utilizou-se o package **timeit** que mede o tempo de execução de uma porção de código em segundos. Para isso, mediu-se o tempo antes e depois da execução do código e calculou-se a diferença dos valores de forma a obter o tempo de execução de cada um dos esquemas de cifra.

[ ]:
```
#tempo de execução do problema 1
starttime = timeit.default_timer()
print("PROBLEM 1")
os.system('python Problema1.py "Mensagem a enviar"')
print("Time of problem 1:", timeit.default_timer() - starttime)

#tempo de execução do problema 2
starttime2 = timeit.default_timer()
print("PROBLEM 2")
os.system('python Problema2.py "Mensagem a enviar" 4')
print("Time of problem 2:", timeit.default_timer() - starttime2)
```

1.1.2 Cenários de Teste

[ ]:
```
!python Problema3.py
```

1.1.3 Conclusão

Na tabela abaixo apresentamos várias medições do tempo de execução em segundos dos algoritmos apresentados nos problemas 1 e 2.

Problema	Medição 1	Medição 2	Medição 3	Média
1	16,82	33,27	22,65	24,25
2	3,66	4,25	4,26	4,06

Como podemos observar na tabela, o algoritmo 1 tem um tempo de execução maior que o algoritmo 2, pois o primeiro utiliza chaves assimétricas para gerar a chave partilhada a ser utilizada no processo de cifragem e decifragem. A necessidade de geração do par de chaves assimétricas implica uma diminuição da eficiência do algoritmo e consequente aumento do tempo de execução. Desta forma, concluímos que a utilização de algoritmos *DSA* e *DH* para que as entidades obtenham chave partilhada faz com que todo o processo de cifra seja mais lento.