

Aula 2 TP – 11/Fev/2019

Grupo 3

1. Números aleatórios/pseudoaleatórios

• Experiência 1.1

Executando o seguinte comando, que gera 1024 bytes pseudoaleatórios: *openssl rand -base64 1024*, obtemos o seguinte resultado:

```
OpenSSL> rand -base64 1024
kPxQ+9KztOnCHM2mnPktPVMQo0T8MA3K1xluSS+85y0CDAJlsYs0A07eZ8F4vAL
atgewtX8uIPaF/1GCS1oZr9oaLTtydMxsAn1kT9nkrSbj2MF1raqcurzF0rwomac
0N3SAmqyU+a8z99EzI92ic9faQgk/kS7gLfxx5LlSLfr9VX29NVhFNascn5QmKS3
kKHnXcyFJ9eWH+6Hasl1svSD1yqC9mwxFRWvvt1uMcUpdR6ALTGh6B9I/nemtJFa
Of8XJ33uUqzA0y8iVHC4ke0GPrDK43VVvxkZqIS/rJruxcU37SuhYS/NY0bpoTN3
Ftnd+lLFxoKXz8XLI//dLJ4xeH9Mk7ggBXX7GRHwQhDQccYLkS30VgiGkPxG+63
roLYjrUakIFLFLurDLdJgFXL7ggrCe9ZpAMVId02ZjOc6UHjKXbnL9C/EddBy5Uj
egAFn+5Am1HXmMr+56JmYs+QJ1m0jKB5Qc4mexVH2EnWVPyRohdCLLle0Tbr0LTN
1ZMfhGE0b5nSRk8Ej/gVON/uP9mt3MmUJLd+GCqAlElnnhzktv4ryIIeh3iZiitU
+4qpkRFshTxM7A+XZ462YNliLOkAX89EWTY/9dGrxDZJvDCMxKXujCXcNM8YV9Gf
MRGHsEEAaMzm25DFEosBpt9lm2D9rCKsfQq8eDafg+gcEwCVR7JEEynDzR1pGuvw
oErkAw7MvWAcoXE+YamaI2Yw9BKQNSk1g7Rvu9b0TrRisD+5RuyHZvYrZr8TRPYg
bWyayANeYzhNLH3Sf0aBI9vsppIgSzjxu6Bmx/Or/1YJt9YL8nQ5F2K5JQvhaVN
NYfn024AIReZUR3rwsstFfctdmF5l50PaWVg4Bd8rjsSaTasWy/tDHjbXD0iZFAgA
F5UDWmgxLNOLDwBR+LPgaTrSydsE6KCA5R5Dsd2Hcw90XFkjz94pTbd1G7CwVLFr
wEMRhpPTGyvLmc2edJEFcr3E5QFL7994PaAfs84D+aLdn4XN5lpsdCcSohb++/Ly
ormyAFhK9cSFM5M11xvepsliV15Y22swWuHYvodP/sjSbrmJAoLBoR3WhYsvy0f
Epc1lHi0CJX7YjTP09fA9V33lu1RQYhQ/84tBMCA80PXRzzV2K0pFRFoN9XnLfsN
Yof/+6mHDJ2joRyt/qFflgSx9C42ngNmTVFgf1QyxSawxE0xTg8h9KKvytGpaPNk
Ts4Eqxq1vDaT59kgGiYjJQB1l0iBTfEzrTZW1IHptfwNshF+s69tmLW6SOL0WwDz
KC02cd6ukZth8RIwXZqI1NmYZDNLfYbYP0L/fL080C0wCg2/RtMEWU0EUaOvRcw
qtL6ZZM7LzpbTIpF8b/SLg==
```

• Pergunta 1.1

Testando os seguintes comandos, vamos obter os seguintes resultados:

a) *head -c 32 /dev/random | openssl enc -base64*

```
diana@DIANAUVB:~$ head -c 32 /dev/random | openssl enc -base64
n74wuEeJAuztqi+5DFRA8QhKXoGiGK00rQL+luw76Qg=
```

b) *head -c 64 /dev/random | openssl enc -base64*

```
diana@DIANAUVB:~$ head -c 64 /dev/random | openssl enc -base64
B/lT0dk2xou5fYlpAFPq8t/LlIB63Xqpr6mCU26/kVl+YtcIxcgqFzUN0aK5JW
a2AQdkyJfYAN5UAb/DfVhA==
```

c) `head -c 1024 /dev/random | openssl enc -base64`

```
dtiana@DIANAUVB:~$ head -c 1024 /dev/random | openssl enc -base64
+1Q2wjEShDqSDVytF8IqWdVJdo8F+nrKAAkBGmIc9CG6RbNqAbIoWgVwAiOPxCL9
z0OzBxLXkhh+TDDhW80+UB+5mPOAz1S4fBedw12WyBERy7w2Vs8pJfA6TtWGAj7e
5mx8gl0fR8PFL0dJz8Y2cCoCAUxso1TZ/ZZ/Mkrqh7lqb2Mihcfm6xsc1SVV+mGN
GP00UFLHGryHqje0uHgImoolStuh85I1TKLEaBxDr2b6U9jhs/cgy2yezMA0vXsY
NNFch4avbTGkOY8iNx8IX3jtpPsczVngpLwe8oa9KKWURPzrcWLKKAkXhAdc6pe
pkZSaGcrffqDNiis9dwuZlC6FWtWmp0yND7a83uAizgd1TUZEcSJoRVhvbC/WFeR
kyCFiLHjv+h3QLtgM6rg0FVUK4m/LAn9Q1131DUXBTsmArXjL6rnKgFkX876c+Y6
GBET5eLehLuanZqaNSnC1tjd6WFLXc8uNSYDYQRF6NT21EEcdlqsm8w5GyDInC3
Mi5LFtu9HSZcydJVJLPfzfjd7W+g930a5Eo36jA66uhU1Ns1Bq4TSqjNeUFLU88
r6ILpEnL1+Xb5PCG2z0qscsJpqtMrH5sCaDa6uQT3mN4FKAm53eTXK7tMYpUBVaa
PCWpD4Qd7L2VqUKvY5KrWGA6H4NhYA63h8EM6M6Bb4GD39eYbePxxvEh2+sJGoYs0
bZTFy5JSRAUJmpHG+c1rGMn9UjPLHsrfuSjUseJgam7x61NqxlC8V7cATmTbXJk
0XzhxparcnaRSKR0TpzBkSBW2Y9oVps649VjUeT81Ptjh7Akp/UQoqh9+fUoVhA
epDR8cvNxNRGcu9uGcepyb9/C9ctuY4B+3R1TuvWgI7igF+mgUAYrV8VNYwbG69r
1pYrW/3s5p9NVQGL0tZGLxDMnmK9YctTucAF0cFPJ1J2Tf0LT7e/czdyvrP6YT
5tVFX2Uo+qH2AqxSczagw9UK24xkDbH9ZNO9r36yeugDA50CUXc6NbnDf9gBxdLK
8REwuIDiedvW4jwMZSvAuZ0JEa/dtW6gJ0DyK3nJnNacDwP7o43GLt3FarePx7Q5
48PL5IGHK78mVvLWjr/dwgnvVNQ6Kbqh+BChxVdFHTPhQyx+Qj3rWBQtn0Znkq8
oCpPV32VV1Qg/K5v8d6W1VPRD3wdXUoLqzgcqFqVfRLeMokY8+4RynfzyMDnP0
ZZ43z65V7iG6PlatNGS8Y0DtkvCUV+tf/dhev9o9io/c0pFVuy64n9A3TaJW3jf
0whc33TveSuaUP5Rg694GZ4H7Xff40rguvDFA/tGvZ2bTyCf2WQTrt50nx2qAnq
32cpm9oIo2Hu4uCuVctaAA==
```

d) `head -c 1024 /dev/urandom | openssl enc -base64`

```
dtiana@DIANAUVB:~$ head -c 1024 /dev/urandom | openssl enc -base64
qPiMJEvX2xADuR9ea4p/iQAv0i+efs6LE21LGiuAPiMnt5fdUPjED2ZPd0hWpX1t
gZw0+7Ure8tAVrAR1oK3LEKUG9Wcl91wny8lh0J9ij0nT9LP+d9/M0PL6SPN+0Ek
MTtZaa2shY0FFMU/jKMoIyyCBcmfJ4Ifo0610u/IG9JfTstHuc2w/WnNh7nWbK6m
2PhkZ1H3CrWnqM/yZzu3FNBBi77ezs+14rnEqi2yTzBppz+Lryxjj1TVJxx1H42b
Zq5Wfub2X9WK6kss06/GreAFLqvYiMoaE0W0ePtGepyN7ZDQ2XTVp4wgU3LUJD2t
rvskyyMIckjDS9EwxyqDSIwpXbGBok8KdVGSnMu2eRMQTHU13fgVtXh0f7NuW30
T9HSBovgAml4qILZmE6ac1SvV8s7Itw18ZtTbRRRW091/8L0SEUmasVe3oL/cK5
VKXJbbeHeuxLHCLu05YHKygtWmnVs1qCviIyLQHUMsEBM024t/gW7DncetCpsBHV
ZA9dXrUmfSyakXCzxTP+bvW47ZynueXtAlm5CPI+P23Si7SV6AmPqG3zSfisBbp
V6k5Wk0mkq0YC7Uurb4Fal/T2hx0/LsuS+Bga6VZj/r3eYCCX/KvncP0nt/aFlp5
LJK4ohBTvNXu0S/MYpt010rAwkLPCjaYToBRqLSnUNwKKc1CdJq0AbyNQ2ivIw1R
kzCaGOA4st9dfQ7D99tLOJJuZAD9sBKWE1wCCArNfxzq0fAkQcZfttHDSVJeJXzy
rISwgvlnm8atpEXun50puxepvwuSjKEQF0c0Q9Be800+x5+rWYxVw/7k54Unmqtg
w2shcosHd0lufjQvSdcX0o7GE/3DgrJvxuxYRST22Qjsmk4kvrXJJj0gb5yLdbQW
JxZ8mkFFt2tRb9B15HuYOANvfjbWUaSKFyoYxOePq+DTNnMLYgKwL91Px+MsLEx
c+WIOQXW/RyKkvJr5r+F2BVQrftPEfG/00USfCjicHP9FwrZBt+9rkcdRw3aLuAa
```

Estes testes permitem-nos concluir que é possível gerar números pseudoaleatórios através do `/dev/random` e `/dev/urandom`. No caso do `/dev/random`, este faz o *polling* do sistema em execução para ter entropia suficiente. Se não tiver entropia suficiente este ficará bloqueado, ou seja, quando o conjunto de entropia disponível está esgotado, será necessário aguardar até que a entropia adicional seja reunida para obter dados aleatórios. No nosso caso, como o fluxo de entropia do sistema se encontrava alto, devido à quantidade de aplicações a correr, os resultados foram obtidos rapidamente. No caso do `/dev/urandom`, este não ficará bloqueado, uma vez que ele reutilizará a entropia que existe no sistema, permitindo que haja uma resposta rápida. Contudo, será diminuída a qualidade da aleatoriedade ao longo do tempo, mesmo assim ainda considerado um PRNG criptograficamente seguro.

● Pergunta 1.2

Para instalar a package *havedged* na máquina virtual executamos o seguinte comando: `sudo apt-get install havedged`.

Ao testar novamente os seguintes comandos, onde obtemos 1024 bytes pseudoaleatórios do sistema e os apresentaram em base64:

a) *head -c 1024 /dev/random | openssl enc -base64*

```
root@CSI:~# head -c 1024 /dev/random | openssl enc -base64
k7qBpD8emwMAiWx2GbCdTIoDs1sFBH89xEUTD5dpz+ob02fqhe82y3i0ubjLsKJ
kq0JC1LZfVvYhfBomvZ4dmLglU8FF/a80KLl0Vu06BjQugfVso0UuSoYgCiecSaR
M2lFM9B/0cs0WbEsp8IYYSt/DXGI5QFZStM08FYueJxcZs5JarJCdzJp0D2WkZwI
FF5iqVjqQNT8Aov9AerIn5jMqs1jRJNM+AEpa0rV6jLLj5YbmyPDN8FpICbYZDM
zQvj5gzLe8Bdx0zMclMpe78a055K7HLPXs0EtoHSEKXtFoyOklnm8LRDq2dFLo8
W6eic0e0PxxDLZki2P5Tddt4Ur8LbQYCrCedp1NVTCTCY4xXJb1FwYraFkbGsdV
YpzlIKS3U8qLc5m14LSMskTFVHjqo8VEEB7P/sDP2cFAVT+o9dkNFv8hySfwL3f
mvvKmhY9TCeyzRCHN8PVKUEZ64G8jRN2/PVAyFMHEXSPZbecx5rbQnichNhgQYr
GVgyF14KgvZ2uApmLg4FnLXY5XV0SVVzWEbhJj4ZNV0QWLFwMU3Lh+XSD/1CCZje
Szn+n5bt19hdeM0qbU6K0CW5HGN3zFmKzFWG2BPSX5sVQECt7LsFLMkYRXPpKnc
AL3Xd95WrKoZ9/m0gMFuDBVjdpysge5fZ/Cf+TBvSG0oc2hQk8ASQzj0z9LYv+D
ArJNi8utCxZZCdvKwco7Y06s52bMDuw/+JllbknV6cKrnG/2ToX+MEeruW1ivLEc
rI042anLUV0sGZJi5acbloclSylFW23Cnn/9SJIxXnMwhz7BfHEXeDv2Qe+Xft97
fUuMfgwrRBkalzQq44x+0SToZyNcSQZS006wE/Ke6+RrDHT8EXYkIgCv9s4AYgH
jzZcicDA0YI1DTt00NkOnpE0yaRigckCL5AMLYQzGvEz75S3KTeaZcU6BgUzCw4/
V/+fjqCvQzVjZyc/LrhpWd+z2A4I0tqPg+nvKt8j73We/1kpZAJLJvR0dwofc5
Sm5EvfhVp07Ne+ZiJf0YcpRtpUvYLn7g6113ihr9LrscZ8DJBsgG/WOTMSGfHVM
FICrvF+KBCVLSIXEsV/eL1CN01h/jZU521yQQWqSv/uK0/xbCB66GsgQTf7JX2L4
qnNgJoqscYidGnwUKJmo+WKBGM2Ee01QpHmK+ELUdwcKRYfju3muQHw2MxLx8kw
upEKLwez9NaLiKxWk4gLwCYU1Qz8dv/VTZsBK/F4Ih1AghY903crKux9cj+ZW4
m+TuMzGRWz2ZCDwRe5ZGAJY9WbWwvtLn+tP15AuLBjWY7102NmA1J47cq5Q0jN5a
HABWUovuh6deL89s60MgJg==
```

b) *head -c 1024 /dev/urandom | openssl enc -base64*

```
root@CSI:~# head -c 1024 /dev/urandom | openssl enc -base64
spnrD+/PrdYUODdlbmAk2s0Zd9d5GTZRYmK8KafIhtW5340jJzFQc/f8a70ISy
WGHZIIoj+AXsR5FVQIsBchJH9h06T9rufKzZacpWmf567//tSULPiK+Cjy6I0kpX
yOpFTTjUjScsqm0tLQzd5j80BNHUI4Pi0QUMpCEfQk0sy0fdeJQzmb0PVNXuZcG6
TNTjLvgvdSbfq0MJL8qDUauLo7UgTocCf6hYyhe233Iu0Qb1ozAkBg0BKkpvhDx
P6+F/frjHvZ/j+9/uKWRhq3Ugm6Pw7hkeK8y3zFJpP1fbaRpX5CU7lygs6FAOL
pSNI34uTtlcKvGi2/hM7TNbGINq/nCvaN08UcCSttKjWUIMcvt+qglfyX0tphxt
rMdkS3hEpw6j3sy9m18MdQ1Vo1m+H+LVZp7fwjLL1UNXs0ja8XUk0UfXKzXu6xoJG
vDLcvcEpuD0dzrZh7tgvSisqGa7xVZjCPFSxFM0R4P86B9Hwyyq5ovw8E9aA14C
ay9DUHE8SNIIIVKNKcVhH+x7x08MCyPoilB2kae01fci7T42tBRqYHm90p1t0hdnj
H2oYMyx8l fhaPpqtS16EEVYcv9+Tgot6+zpmxLgeU5cPndc5sR3kyiPqo3Mz0e
Kcudj13WlykgfjvNtrLj0KKKBxIscl5bk2CLDH4IjNCL33ayw5i4RxoCHsx0sYua
ukZaRLc+taAB90uipMvvpjse2Fwb06EhP8Xe8rUzd8LYfpRksN3KYTPH+Rmc+rht
vFFHhZfAKXywpwZrV08/1Cdq7Rz+cv1hsyGm5nAZnCsBpcjY9Fgj43o3Dc64+dN1
ETjzCOTDsoo2M4LWjEi6WZELI4Ztf1bDXFc374s0fAIbwr4q86P0/tBcAdRISW4o
cUFGkiAYdyNHJLIwpHL3yXa/6K2dnn9KtSmg4UspmT059VY6LzGAtmGWLMIaUrhx
yGX1fQCMOU5eXAsxtH5tXAUkdEmrftutJxc/IBrdh/NtsYmSffFTLMkwKS90dj
yEwTpNGVW5j0fd+h06amk8JRq8C+CI9NaZvv4UV3+103YqpNsSg2pHI66y4+Hr4
C7tRUQFE6pHLs4v//iFq4McnSL3hj4d5nDZya8p906/tuDj0F+Pxn1QEE7e05WK78
M06DJA+1sXmK7HAzV4TvQ139cpVmqMWhZUeWunmeTSwn038H0mm9dm50jay9SP
auPmgN536s7MDhoxd3uNzoa6GfVxdAwNLZ9FU0WY+MXH+8pLItr80BzXDsa02Sgs
fy0D6Z5bwNdgoGSKY8wGh1F3hJHcnW3Bu25pnV1MK50rFJ1qUxL9o0Hu6o23GcmI
meKg6cdjCnZrnIt17bknQ==
root@CSI:~#
```

Analisando os resultados obtidos nas alíneas a) e b), vemos que o tempo de execução dos resultados é idêntico. Contudo, isto é conseguido devido à instalação do daemon de entropia adaptado do algoritmo HAVEGE. Esta package serviu para ultrapassar as condições de baixa entropia no dispositivo aleatório do Linux.

2. Partilha/Divisão de segredo (Secret Sharing/Splitting)

- **Experiência 2.1**

Através do *genSharedSecret.php* codificamos o segredo “EngenhariaSegurança” em 5 partes.

```

root@CSI:~/Desktop# php genSharedSecret.php "EngenhariaSegurança" 5
Codigo 0: 10110000110110111111101110000100111010100101001011000110101001101000
10010010000101000000000010010000101100011111000100001101110000011010101001001001
1001100101
Codigo 1: 100001001101000111011001100110011001110001010010111111010111000010
10001110010110001010000110000100011110010011101101111101011001111000100111101100
0110101110
Codigo 2: 0000111001100000101100110101010111011111000100100100110100110011110000
11001100110101101011110011100000110000101001110110110110010111110100001101111011
0001001111
Codigo 3: 0111110100110110100111111000110111100000010111001111110010010000101001
11110011100111100110001100101110010111010111110100010111010111010010100000111010
0101011001
Codigo 5: 0000001000110010011010011010000000001000111110000011111010101111101001
0111101101010000101001111000001110001010011101100100100111111111111100000001101
0110111100

```

Na imagem seguinte, primeiro usamos o *reconstroiSecret.php* para reconstruir o segredo juntando as 5 partes obtidas anteriormente, e obtivemos sucesso.

De seguida, tentamos reconstruir apenas com 4 partes e posteriormente com 6 (sendo que repetimos uma das partes). Em ambos os casos não obtivemos sucesso.

```

root@CSI:~/Desktop# php reconstroiSecret.php 101100001101101111111011100001001110101001
11011100000110101010010010011001100101 100001001101000111011001100110011011001110001010
10011110001001111011000110101110 000011100110000010110011010101011101111100010010011
01000011011110110001001111 011111010011011010011111100011011110000001011100111111001001
00001110100101011001 000000100011001001101001101000000000100011111000001111101010111110
1101011011100
Segredo: EngenhariaSegurança
root@CSI:~/Desktop#
root@CSI:~/Desktop#
root@CSI:~/Desktop# php reconstroiSecret.php 101100001101101111111011100001001110101001
11011100000110101010010010011001100101 100001001101000111011001100110011011001110001010
10011110001001111011000110101110 000011100110000010110011010101011101111100010010011
01000011011110110001001111 011111010011011010011111100011011110000001011100111111001001
00001110100101011001
Segredo: G\0f0_0i\0F0#00
root@CSI:~/Desktop# php reconstroiSecret.php 101100001101101111111011100001001110101001
11011100000110101010010010011001100101 100001001101000111011001100110011011001110001010
10011110001001111011000110101110 000011100110000010110011010101011101111100010010011
01000011011110110001001111 011111010011011010011111100011011110000001011100111111001001
00001110100101011001 000000100011001001101001101000000000100011111000001111101010111110
11010110111100 000000100011001001101001101000000000100011111000001111101010111110100101
10111100
Segredo: G\0f0_0i\0F0#00

```

Podemos concluir assim, que para reconstruir o segredo precisamos das N partes em que foi dividida a mensagem quando a codificamos. Usando menos ou mais partes não iremos obter o segredo.

- **Experiência 2.2**

Como se vê na imagem seguinte, usamos o algoritmo *shares.pl* para codificar o segredo “EngenhariaSegurança” em 7 partes, sendo que para depois reconstruir o segredo através do *reconstruct.pl* apenas precisamos de 3 dessas 7 partes.

```

root@CSI:~/Desktop# echo "EngenhariaSegurança" | perl shares.pl 3 7
3:1:10dd0553107a6a93b6babe942dfbf547e8da0b91:
3:2:1935b54d78df4452f7c4c9ef37a6dacd5ba747cf:
3:3:60787553a596f0b02b7f7475857721f2c92a5a1a:
3:4:e5a5466597a06cac54ecc027166eccb630644474:
3:5:a7bc28834efdba467109ac05ec8bd919925405dc:
3:6:a7bd1badcbacd97f82d9380f04ce481ceefb9e51:
3:7:e5a81fe30caec9568759654561361abf43570dd5:
root@CSI:~/Desktop# perl reconstruct.pl <<EOF
> 3:1:10dd0553107a6a93b6babe942dfbf547e8da0b91:
> 3:3:60787553a596f0b02b7f7475857721f2c92a5a1a:
> 3:5:a7bc28834efdba467109ac05ec8bd919925405dc:
> EOF
EngenhariaSegurança _

```

De seguida, tentámos reconstruir o segredo apenas com 2 partes, o que não foi possível, isto porque 2 partes não são suficientes, porque anteriormente dissemos que seriam necessárias 3 partes.

Tentámos também reconstruir com 4 partes e neste caso já obtivemos sucesso, pois como eram necessárias apenas 3, o algoritmo ignorou a última parte dada.

```

root@CSI:~/Desktop# perl reconstruct.pl <<EOF
> 3:1:10dd0553107a6a93b6babe942dfbf547e8da0b91:
> 3:2:1935b54d78df4452f7c4c9ef37a6dacd5ba747cf:
> EOF
too few shares at reconstruct.pl line 77, <STDIN> line 2.
root@CSI:~/Desktop# perl reconstruct.pl <<EOF
> 3:5:a7bc28834efdba467109ac05ec8bd919925405dc:
> 3:6:a7bd1badcbacd97f82d9380f04ce481ceefb9e51:
> 3:7:e5a81fe30caec9568759654561361abf43570dd5:
> 3:2:1935b54d78df4452f7c4c9ef37a6dacd5ba747cf:
> EOF
Ignoring share 2...
EngenhariaSegurança

```

- **Pergunta 2.1**

Primeiramente geramos uma private-key.pem e um certificado.

```

root@CSI:~/Desktop# openssl genrsa -aes128 -out private-key.pem 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for private-key.pem:
Verifying - Enter pass phrase for private-key.pem:

```



```
root@CSI:~/Desktop# openssl req -key private-key.pem -new -x509 -days 365 -out certificado.crt
Enter pass phrase for private-key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:PT
State or Province Name (full name) [Some-State]:MINHO
Locality Name (eg, city) []:BRAGA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Engenharia de Seguranca
Organizational Unit Name (eg, section) []:Grupo3
Common Name (e.g. server FQDN or YOUR name) []:grupo3
Email Address []:
```

A:

Para dividir o segredo "Agora temos um segredo extremamente confidencial" em 8 partes, com quorum de 5, foi usado o comando dado:

➤ `python createSharedSecret-app.py number_of_shares quorum uid private-key.pem`

onde `number_of_shares` = 8, `quorum` = 5, `uid` = 1 e a `private-key.pem` gerada anteriormente.

```

root@CSI:~/Desktop# python createSharedSecret-app.py 8 5 1 private-key.pemPrivate key passphrase: segredo
Secret: Agora temos um segredo extremamente confidencial
Component: 1
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 2
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 3
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 4
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 5
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 6
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 7
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 8
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 9
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 10
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0AritJnQYK0ssfWZiLC_o5d0wt3CBl0h8HkNKHTRwHLD-8kCXJEHKBw9LmEboSomC-YdLdNLSLR8emsgsQyB1u
Component: 11
eyJhbGciOiAiAiuLMyNTYiOiBibjEjMtyYjYzJi0tZmMzBld0G1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YXOTJkZThiYj0kMjcxd0k0MDU0ZjYwMDM3NDkMc2IwY2FjcNjQyZSNTY3RlZj03YThiIiwgIjEiLA1CA1LCA4LCAiMwU5YmIwYTUwYjNkM2EyYmRkMWZlNWNhN2IjZGVmY2IyZmN2ZjY2I2NDZmZkM2I0Y2ZkZmZlY2IwMmQlN0G1NSjNkN1Jdf0. RlU4w-wHlU86rf9UMLPfPBhBGZiI-PROF6WIsdEsqHZZ1XmUg90d8dzgFRXwnYAW_yc9-cY0XP5ara05VU306F0Arit
```

B:

Ao usarmos o ***recoverSecretFromComponents-app.py***, como mostra na figura em baixo, apenas precisamos de 5(quorum) partes para reconstruir o segredo.

```
root@CSI:~/Desktop# python recoverSecretFromComponents-app.py 5 1 certificado.crt
Component 1: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU
Component 2: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Component 3: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Component 4: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Component 5: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Recovered secret: Agora temos um segredo extremamente confidencialroot@CSI:~/Desktop# █
```

Porém, se usarmos um número superior ao quórum, por exemplo 6, também conseguimos reconstruir o segredo.

```
root@CSI:~/Desktop# python recoverSecretFromComponents-app.py 6 1 certificado.crt
Component 1: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU
Component 2: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU
Component 3: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Component 4: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Component 5: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Component 6: eyJhbGciOiAiUmlrNTYifQ.eyJvYmplY3QiOiBBIjEtYzJiOTZmMzBlODg1ZmFkNDViZTA3ZWU5YjNkYTAwY2MwOTBkNDdhNzRiY2ZmN2YxOTJkZThiYjK0Mjc0ODk0MDU0ZjYwMDM3NDkwM2IwYzcyNjQzY2Y5NTRlYzcyYThjIiwgIjEiLCA1LCA4LCA1MwU5YmIwYTUwYjNkM2EyYmRkMwZiNWNhOTJlZGVmNjAzYzU2Y2I2NDdjMzI0YzdkNDFlNzIwMmQ1NGlS5jNkNiJdfQ.R1U4w-wHlU86rfU9MLPfPBhBZG1i-PROf6WIsdEsqHzZlXmUg08odzgfXRwnYAW_yc9-cY0XPSara05VU306F0AritJnQYK0ssfwZILC_o5dowt3cBl0h8HkNKhTRwHLD-8kcXJEHkbW9LmEBoSomC-YdL18dNLSREmmsQBylU_Zhf8Y1Byc4lGEC0yBPhW09IuNDHLw
Recovered secret: Agora temos um segredo extremamente confidencialroot@CSI:~/Desktop# █
```

No caso do ***recoverSecretFromAllComponents-app.py*** é necessário o uso de todas as componentes. Isto é, se pusermos o quórum=5 não iremos conseguir reconstruir o segredo. É necessário que o quorum seja 8, isto é, as componentes todas são necessárias.

3. Authenticated Encryption

• Pergunta 3.1

A encriptação autenticada garante simultaneamente a confidencialidade e a autenticidade. Para garantir integridade da mensagem associamos um Mac. Usamos assim o modo Encrypt-Then-Mac.

Cifração:

Cifra \leftarrow Enc(mensagem, chave)

Mac \leftarrow Hash(cifra)

Return (Cifra || Mac)

Decifração:

Decifra \leftarrow Dec(Cifra, chave, Mac)

Return (Decifra)

4. Algoritmos e tamanhos de chaves

• Pergunta 4.1

O site <https://webgate.ec.europa.eu/tl-browser/> disponibiliza a lista de Entidades com serviços qualificados de confiança, de acordo com o Regulamento EU 910/2014 (eIDAS). Para a realização desta questão foram selecionadas seguintes Entidades de Certificação Francesas (EC).

a) Ministère de la Justice

```
root@CSI:~/Desktop#
root@CSI:~/Desktop#
root@CSI:~/Desktop# openssl x509 -in certificado.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            11:20:91:96:4d:71:dc:c8:59:5b:f3:4f:fc:9f:08:21:34:f9
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = FR, O = Justice, OU = 0002 110010014, CN = Autorité\CA9 de certification Justice
        Validity
            Not Before: Jun  9 00:00:00 2016 GMT
            Not After : Jun  9 00:00:00 2022 GMT
        Subject: C = FR, O = Ministère de la Justice, OU = 0002 110010014, CN = Autorité\CA9 de certification personnes 3
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
            Modulus:
                00:c7:24:cf:48:e4:33:91:9f:53:63:05:6d:62:b4:
                b4:ad:52:fb:a2:db:da:06:36:4c:09:56:4e:2d:99:
                6d:06:6c:0a:08:0f:04:b0:08:5d:74:1c:2f:37:
                7c:77:0b:b1:eb:2e:ec:cc:5b:08:39:02:82:a9:78:
                47:ff:c1:1b:e0:da:1c:1f:5c:9a:91:3b:2d:aa:90:
                37:0f:b5:e5:e7:49:18:05:7d:1e:8d:00:38:94:54:
                1b:00:ff:ab:50:12:d2:90:e3:a5:01:b2:34:37:04:
                73:32:46:39:76:91:8c:03:19:ac:f8:cc:15:25:08:
                ed:47:26:d3:85:aa:10:77:5b:08:0f:77:c6:e9:08:
                00:36:2f:9f:d2:b6:09:25:2e:a0:40:30:da:23:46:
                0a:1b:2f:aa:0e:07:19:8a:68:55:1d:b8:62:1c:e2:
                4b:35:74:c7:41:77:a4:64:09:16:47:b5:06:30:37:
                2b:0b:ef:57:85:e5:24:64:a0:53:9c:4b:33:0f:4f:
                fe:2b:bd:06:0c:82:5e:b0:d7:ce:a4:61:42:ef:ff:
                6d:b6:3d:5a:c8:5e:ab:a9:d2:00:7c:aa:b4:eb:b0:
                7c:82:a5:8c:09:b2:4d:5c:ab:03:fb:95:1e:30:79:
                00:10:42:ff:da:0e:20:04:11:a6:b7:14:0b:b8:7e:
                c4:b5
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
            Certificate Sign, CRL Sign
            X509v3 Certificate Policies:
            Policy: X509v3 Any Policy
            X509v3 Basic Constraints: critical
            CA TRUE, pathlen:0
            X509v3 CRL Distribution Points:

            Full Name:
            URI:http://www.justice.gouv.fr/igc/onts/mj_ar1.crl
            X509v3 Subject Key Identifier:
            98:E8:57:0C:F0:06:21:8B:A7:93:38:9D:F6:A7:82:31:6A:E0:68:5A
            X509v3 Authority Key Identifier:
            keyid:7C:E9:9B:15:B4:CD:43:89:11:A5:EE:9B:8A:AS:6D:35:08:B2:E1:D0
```

```
Signature Algorithm: sha256WithRSAEncryption
1f:cb:4b:ea:c0:a9:d5:e9:4d:7c:83:5a:55:8b:44:e9:b2:c1:
0e:60:ca:e5:a6:e0:2c:33:c5:64:de:eb:e2:f2:1c:f1:8b:3a:
22:78:29:ed:60:99:29:b2:73:4e:cf:57:49:23:e2:8a:a4:a7:
b4:b8:29:98:c1:2f:24:8e:9f:4a:50:47:e6:0b:ec:bc:dc:b4:
71:16:e7:14:a2:e7:02:4a:3b:c5:6f:9b:da:38:ea:aa:c3:d7:
84:f5:6e:fa:c2:e8:c6:e8:9e:29:99:28:68:b3:e9:a9:18:50:
f4:e3:85:1c:29:12:ee:84:bb:e1:d5:0e:bc:29:b8:3c:00:3e:
26:41:23:10:16:96:77:2b:7c:95:25:a9:41:4c:61:eb:00:86:
3a:3a:a3:4c:36:14:70:7c:74:0e:06:f8:2b:91:bb:c2:fe:b1:
39:41:27:b3:9d:06:57:e4:1b:33:33:e3:c6:b2:52:f4:1b:c5:
f8:67:dd:da:70:c7:94:99:7b:00:01:dc:c0:fd:d5:71:fd:03:
8f:aa:36:8f:54:92:bc:4f:e8:fc:5e:bb:58:e5:aa:a3:c5:eb:
f5:6e:be:94:f4:d4:83:2e:e7:57:ac:0e:8f:e8:3b:c2:8d:1a:
1f:16:55:6a:70:52:53:07:85:de:7e:21:5f:93:91:0e:9e:01:
b8:fa:56:98:97:12:d7:52:43:ab:ea:f1:38:7b:08:a7:8c:2b:
27:bb:94:df:23:a1:96:78:ce:7d:41:fa:19:89:59:11:43:fd:
70:b6:c8:2c:1f:87:a3:a8:f1:c4:3c:e2:2e:49:9c:33:15:d4:
d7:22:e6:99:41:88:e8:8f:fd:e6:06:da:32:7a:cb:f3:96:c6:
4e:5a:b1:72:c3:d7:e3:3a:58:8f:e7:e5:66:e6:f1:35:0e:d2:
31:49:53:ca:69:7c:cf:a7:6a:14:fd:b9:c5:85:47:84:97:9f:
cf:03:fc:99:d5:75:64:32:b1:6f:b4:aa:6b:01:bb:eb:8d:a0:
87:65:c4:81:32:34:b7:2e:46:eb:f6:13:03:d2:e2:7d:22:16:
8f:e3:97:5d:18:e9:48:88:a5:55:ba:18:7b:04:f2:ff:38:f6:
de:b1:21:fd:1a:25:94:f5:b9:5c:e0:ff:c4:d3:6d:4:35:e1:
63:5e:f3:77:91:99:e4:b4:b5:10:b0:76:2f:ed:d3:85:ba:78:
58:b1:52:48:a3:18:9a:b0:69:88:43:18:57:7e:67:e8:26:f2:
bd:a0:05:ac:57:e3:dc:62:a0:b4:ac:ef:25:14:e6:5d:54:41:
8d:cc:51:1f:5d:c4:75:ff:22:d2:72:7a:c6:27:4e:e0:8d:78:
63:d2:d9:2b:11:bc:72:90
```

root@CSI:~/Desktop#

b) Imprimerie Nationale

```
root@kali:~# openssl x509 -in certificado2.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            11:28:d2:ab:ab:6f:88:e5:a5:b6:e4:6c:e5:4c:21:4b:48:11
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = FR, O = Groupe Imprimerie Nationale, OU = 0002 410494496, 2.5.4.97 = NTRFR-410494496, CN = AC Imprimerie Nationale Racine
        Validity
            Not Before: Apr 28 00:00:00 2016 GMT
            Not After : Apr 28 00:00:00 2026 GMT
        Subject: C = FR, O = Groupe Imprimerie Nationale, OU = 0002 410494496, 2.5.4.97 = NTRFR-410494496, CN = AC Imprimerie Nationale Elev(CP)AR Personnel
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public Key: (4096 bit)
            Modulus:
                00:ef:37:37:a7:c7:af:60:d9:a2:8e:ff:00:2b:03:
                7f:f3:fa:33:3a:8c:21:29:da:f0:f6:ae:f2:3f:11:
                9c:a9:09:cd:04:07:03:c7:d3:20:7f:3e:a1:2a:04:
                42:7f:5d:a0:49:01:05:36:ae:3d:c6:ba:da:96:a6:
                37:4b:44:a0:51:06:63:2d:f0:98:07:58:77:3a:55:
                a0:11:4a:f0:00:ae:13:4d:37:08:e4:f5:00:ba:c2:
                92:5f:09:2a:0e:52:b2:07:2d:0a:71:0f:fe:61:ae:
                6a:39:24:13:03:05:c2:4c:0a:31:03:34:54:06:c2:
                70:07:20:9e:c7:2f:0d:05:9f:30:a8:cc:02:f9:09:
                ab:31:2c:46:25:cf:c9:9c:70:44:21:00:0b:08:4c:
                e0:1d:36:a2:6e:73:d3:fc:69:ab:94:26:ba:21:32:
                9f:ae:44:6d:f3:0a:00:77:00:04:c0:cc:0d:6d:ac:
                ca:03:70:f0:11:7f:8c:90:5b:65:e3:00:d1:c1:77:
                3a:51:a0:a2:0e:00:04:0d:c2:f0:e6:7a:01:00:00:
                c3:4c:09:05:12:07:75:69:00:00:a0:a2:a5:00:93:
                74:7f:d7:15:06:07:f2:07:20:a5:05:1d:c0:1e:4a:
                a7:10:20:2e:a5:00:1c:20:37:76:57:79:54:3a:0c:
                06:95:24:2c:a7:26:0d:00:24:2e:bc:59:16:2d:c3:
                75:5e:04:0f:29:1a:ba:3f:53:9f:c0:15:b3:22:05:
                e2:cf:34:c6:f5:0d:48:e3:15:f0:a5:6c:4a:08:05:
                e7:0d:00:30:09:f0:01:03:14:e2:91:c0:30:77:
                aa:04:3b:29:e0:56:7c:52:06:63:4a:26:c5:57:32:
                7a:c1:c8:03:c0:00:0a:06:77:c9:44:30:11:65:03:
                54:05:c3:c6:07:06:3f:f2:20:ec:a5:16:0e:99:31:
                51:08:22:0c:07:29:b4:a2:c5:35:0d:27:ca:69:ac:
                a3:01:05:e8:ac:00:70:00:02:1c:0a:0b:03:57:06:
                ca:62:35:09:13:36:83:70:70:13:71:52:ee:f0:26:
                72:08:07:b0:2c:a1:0c:30:e0:70:1a:f0:33:a3:ff:
                dc:a6:c1:74:1f:09:07:00:70:ab:49:04:c0:ca:4d:
                a3:f5:a0:79:4b:73:2a:70:07:2b:0d:ac:00:76:a2:
                df:e4:96:9e:55:12:d1:c5:08:1f:b0:09:01:5f:c7:
                4b:0e:0e:4a:41:72:00:2a:c0:70:39:02:c0:0d:00:
                24:40:3e:c2:0f:a1:03:cc:54:51:00:2f:f9:07:
                db:00:4a:e3:3e:bc:94:c9:ce:00:48:22:47:50:c9:
                af:23:0b:
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Key Usage: critical
```

```
Certificate Sign, CRL Sign
X509v3 Certificate Policies:
    Policy: X509v3 Any Policy
    CPS: http://www.imprimerienationale.fr/GIN/PC
X509v3 Basic Constraints: critical
    CA:TRUE, pathlen:0
X509v3 CRL Distribution Points:
    Full Name:
        URI:http://www.imprimerienationale.fr/GIN/CRL/ACR.crl
    Full Name:
        URI:http://crl.imprimerienationale.fr/GIN/ACR.crl
X509v3 Subject Key Identifier:
    C9:AA:06:97:35:09:05:DD:47:CB:9F:54:0E:7F:DE:28:29:CE:3A:15
X509v3 Authority Key Identifier:
    keyid:CD:0F:07:0E:AB:6F:21:04:0A:77:3A:56:BB:44:50:7F:E8:F8:94:E9
Signature Algorithm: sha384WithRSAEncryption
b1:3c:df:17:22:fa:7b:74:d4:42:04:5dc8:ef:08:58:0d:8e:
bd:05:8c:fc:1f:42:8c:5f:e1:a0:bc:de:19:30:2d:37:53:2c:
83:6e:45:30:21:61:d2:61:bb:e4:b6:ae:9c:b0:b2:38:06:2b:
13:03:09:74:2d:df:a6:92:0f:3b:37:a5:58:2b:16:5a:2e:09:
2c:10:2e:92:c4:be:a2:69:f7:c4:48:a7:95:55:ee:fb:5a:2f:
cd:1b:e2:5d:01:1c:46:44:2f:fc:c5:0f:a6:32:1b:18:48:84:
46:fc:0f:fe:56:09:48:07:05:0e:09:1a:72:6b:b3:aa:f3:0b:
5b:bd:9c:aa:22:43:ff:3f:3c:29:bc:07:d4:2a:e2:bff:f8:0f:
c6:a7:c4:c0:9a:39:60:f3:f2:03:08:06:10:c7:4e:a4:66:08:
ab:db:c8:9f:22:eb:a2:64:8f:a2:fc:0f:4a:7a:ee:15:0e:08:
73:03:20:63:c3:ae:0f:37:20:fb:af:0f:24:95:fb:78:90:c6:
c9:c1:fa:e2:97:b0:0b:12:08:a9:08:07:00:2c:2b:75:3c:e0:
6a:ef:c0:cd:42:b3:2d:5c:b7:d1:42:58:e1:18:45:2b:ff:9c:
00:7c:a5:2d:0d:e0:05:e0:3d:1f:8c:b2:10:44:0d:08:47:36:
71:ee:0a:73:a8:33:95:a1:b0:16:53:c8:05:16:0f:ca:0e:24:
0b:47:03:ab:2d:20:3f:55:0b:b3:f0:71:04:c2:77:c3:56:07:
fc:0c:12:0e:ff:a9:2e:09:51:05:01:fc:05:0c:00:7a:2c:0f:
b1:77:62:1b:c9:d2:c9:63:05:a2:c5:7f:09:e2:02:4d:18:9c:
79:d6:4c:b6:3b:c0:46:29:27:58:c7:58:5c:c4:77:f5:a4:17:
9c:6e:7c:46:ab:bc:08:74:36:e0:df:0f:ba:0d:f4:4d:0a:0a:
29:af:ca:4c:ee:cb:e9:23:5f:19:db:a6:01:35:93:f4:18:40:
6a:32:20:8a:c7:c3:ae:0a:da:47:3d:47:24:3a:9a:27:b0:25:
e0:f3:79:ba:fe:3b:9c:ca:08:02:77:ba:1c:4c:3f:ce:4b:04:
4c:92:00:25:de:2c:e0:35:7e:07:2b:fa:76:3b:07:c0:78:21:
0b:f5:c4:33:29:28:00:e7:09:46:b7:52:3b:02:40:0b:19:05:
bd:b2:72:3d:01:c1:03:99:c1:a7:45:6f:01:74:e7:00:0a:05:
58:f3:48:bd:bd:d1:15:4c:ac:37:0f:11:b3:46:44:18:0b:7b:
c9:92:01:aa:a4:ae:aa:03:cc:77:02:c8:05:41:2d:c3:ae:0b:
9a:3b:02:a4:33:08:fc:9b
```

A seguinte tabela mostra um breve resumo dos resultados obtidos através da análise das imagens acima de cada uma das Entidade de Certificação.

	Tamanho da chave	Algoritmo
<i>Ministère de la Justice</i>	2048 bits	RSA
<i>Imprimerie Nationale</i>	4096 bits	RSA

Assim, podemos concluir que os certificados do Imprimerie Nationale como têm uma chave de 4096 bits são mais seguros que os certificados do Ministère de la Justice que apenas possuem uma chave com 2048 bits.