Réseau - LengthRevealingEncryption - 150 points

Kévin DUVERGER

Table des matières

1 Résolution LengthRevealingEncryption :

Encore une fois, commençons par intéragir avec le serveur `146.59.227.136` sur le port `23413` :

```
Please make a choice :
        [1] : get the list of all pages on the server
        [2] : see your friend's navigation (20 pages)
        [3] : get a page of the website
        [4] : give the names of the pages visited by friend (separated by commas without spaces)
Your choice :
```

Comme dit dans le challenge, notre ami est le seul à pouvoir communiquer de manière chiffré avec le serveur, mais nous souhaitons casser tout le protocole pour savoir quelle page il visite.

La première chose à faire semble donc tout naturellement être de récupérer la liste des pages sur le serveur :

```
Here are the pages : index.html, 404.html, maths.html, info.html, physique.html, chimie.html, maths/
    arithmetique.html, maths/probabilites.html, maths/analyse.html, maths/suites.html, maths/
    fractales.html, info/bdd.html, info/programmation.html, info/algorithmique.html, info/
    demoeffects.html, info/graphes.html, physique/gravite.html, physique/pousseearchimede.html,
    physique/frottements.html, physique/ressorts.html, chimie/nommagemolecules.html, chimie/
    acidebase.html, chimie/nucleaire.html, chimie/equationsreaction.html
```

Nous pouvons également tenter de voir les pages visitées par notre ami (je ne donne que les 3 premières et les 2 dernières pour que ce document ne dépasse pas les 100 pages) :

```
#Page 1 :
83b271034d600ab953049799ca4f384e582091076335cf8a0822db97958f46be4fe37bdee55900796c9d1a734
8ded8a4e913343feb55914112da92a9b3e4da7fd80473bdfd0c8d5b9cbecdd17b5597cb794ff45397894a126f
3a939578f3a72ec2e152654db8c3f5f04b1c679c75fef17707af01dd7fb3de6945c6c9f7fb5944b1b9725c5d6
37a4ce8299bc410397eb9e58b302b06f277ed9d76b50fe7a72bc944f0ac357c8a4963ad13ac4f2f0d72e49fc6
072622c1f49e40568c581315a5dba8619034f37ff783b961cdf6647581e93011509b5905af9def52a2fee7905
df55aa390f2112ec91534bedea18561e21af9d9e6969979861c46bd9516b2276dd90e42db3f82144c16d0923a
d3f84885efe690a25c762ec1ae961d075459f1446ec1868a8eafea154efb074e01f2b5d247d025d6185aab1f7
09d31d1bfcc9275b4425b95e28a3e71c31721c0993ad7483f081e80114ffa3b2d6818d55f9a1e909500745960
7bd29506bdcd5994eb9ec3d8485289d784b1754583ddf69ddf4ff78b39df9286d7647caddfe632b7ccb110616
8daa24879daba7d88c937ed03fb548135b6ba5f15e1eaae2bdd07fe6c7b864337d15fabe2ab3b2256c4e4e91a
a67ee7e28e616dcb2bbf0cd01993a4080cef49c18b25ae237c7aa86f88901623c2ffd26f077ab441b13e8282d
6a3d5809159cafb5475df6c40d830019692b2709209871aafbd9b1601d5085b3d65952953f7f7e81c2eb88bf1
a69bcbf9db3daa840aeda166d3b995df377fac9fccf880e5e91f2caa1443ae17f4b87fd565ba2ad6eeecffb88
2709b4bfa6af09fc92e0115b3ca2d0c241b2a92571a5509bab61d08d3afe7697c33c93b4a7ebbedce95444b4a
f5a194eb0c7a4713edcf772f993d3b6f51015a4f95ed3adf8d689367715da1f40def341e895bd935fb1c9c2da
bd9ce6309c50ea3dc46422d83a5f1c2c4d4eef275d33ede27f609a6a3502ab62fe3553e3ce7fafdd7e18e7bd4
db1ea52acd4e6f2e7b56b93109aff8d875b48dc3daa9a2fc27c6b0da2af51c475e8ebb0d97f01aced904b827d
038380f0b09883c9b4a76ad24b9ba1d3cf18b23b3435d9655d6f8d324dc5f2e0689b9f64386676b1b78a659c8
aded6c67970cef0b660629c6e8ee80df06bae6152174ca58527e96f9cfc2dfa8b78bce591dec38b4c0465ce33
242113a70d13847e496a7ecb77c047ac2f2eddeea76725b06d71eb67e88e3e835d2ebc4fe2bcad2c2d447f7a3
ea7c2ff507
#Page 2 : fdff212be461cf5029250293dcfedd24e05d425ac850399f52031e46bc5719e773a456887cfd1bf
b430b588d25baf47cf48512d00dac08b5347ce77c9543fa0fb3329bba7cf6bd9f2e15a39403f9944377290736
f3ec75d3fd13488da25d63ff90491e0771992ceb37840645efdb8abfe0d2a3dc315938cdc088f7dc09f86ceaa
5b2baf9afe0195f03d862ea90c1f3d86fe1674b22d5cd9276062b324eeae7107816cebd4b01ba46ece6f2bf72
a2c38f57731de8aa223042b444deb5fde9bab5ae9ee78b1ea7032a81a3e0ce18c341ff0816e2b9adb84aa65d7
b57d92147f5c9237381747fa4920062f18337834b3c01d4ffb4adec1d4a07c2451d4d97c73fc87961a10ed493
b73bc3e0dfa43e22d12a610459b4c979fd9926ca3cf1a033af44df541edd98951a2a96e5974a174fbca33f6c2
f94320a92aaeaa1a159729a5a422abd14e85216bc0a7950976ef1203b4de37b93c07a06e60803c361b2e9a142
17a8b500289f6770f3a13ec5e75354f922e7785ff40c70c0a3d8f62838da4a6544d5d88b3d238dfea615680d4
c678fc43579243f0e22274247288be26fc47c16127a080bcc3cbbbfbae553a4567c24888ea85959d931369543
d31a62f4e82480082584975ad080c4728b51b7a1ad0e648139a45b67fdd9ea4db6eb4609619c51b029b19b941
eb5a482825648df0d1a8217205eee363fe7265707e453e466c56bf9c784644e6afe275d999f28463074c82c35
ccab5fde9336b4af2fa69e1c6e543bb975529d20ab8f596e5b0a7101777b04dcdcbcfc287db02589388801429
54603a7e211080fb228b897ecd4b0a0719356721597bbbe87cbbbdae84705d5311b1f6061624e5544606fd94e
55ec12b09f754bf858f159c804cf21d824f097321f0d7365030f79ed5af69a811e6334f2711aabb14c5261bef
33a2ad6e6805c76f34727f19dd4db6934779a1298a25ecae21b7613711e483be65330ee4d724b3bb3bb11030a
4c01db579bb30a52a8b3d9d3c4994cb6c26448a8667cc21f072f14521b3971e43950f35fb1ef57c8597916adb
4bbd08b
#Page 3 : f47c302ebf56920b0651c9f5be4f5e140547cac9ad1d22661b1ba10c29c6cee01d13c77b38a7bf8
8ce3c388ae12931cf0d5d8ac6698b6c5ef00d9e63359243e66b0ed3e06f0f302d6aa135bbac29254665a10ad7
4439d17493eb760f87f5daa2dd7291868f1f20e597a1dd98474c705e47814d35b74b47bb04c5ed504313e8d92
d4a74474b28dbefb9f12f208bccbfd93b2672812fac9927dc98df8a096e24fe15b37f090d618965a7fbb2fb22
69c72102088ec7c522ff5466eea2d73e4ec9724b3ff70c0b329cf1038881aaa0def0f211f9ca7f3c
...
#Page 19 : a5e10cc195b1423c6dbfd98c2bde4a317fd44950d6a90303c88dc8409d7dd59f6f7fb3444205ff
be937640c250b5396acc35ea6c79a95c9c2f9e33424c02b12d5343dc7c2e52f364057e41b7c89eee23f87a56b
1a0cdddfb871c50d55b3a9a0aa343156b84d577e159330c9c870121838586afe5d7c4980d0019f4f96df379cf
fe36d27b719eadcfcb9e8c2ad90a300729f4c5120e8a8d06fd0f303c551a08450137ad63b5319cf8077483644
b0891a24d87f70b6d18b7d6c8dd3fc43ed9a0a3e1823e11a39ffab6663b59302a829b17dd264f852838071ab7
918c16b2a45287a3de855999a6a75b59cb8be3146bdb8b49c7e04289c7a821b845d3864e39fd056bcb
```

```
#Page 20 : a0f762a5b76e477e8fbada57e8364ab2d0e0571f659ee6ed9555d4dac441676737629bbcbb2cab
054fd29d9aa171b0fd78e73c76f1d6ff5ad357d4f3dfbe06dcc6aa08b1fac179439e6c9226a3bace86e5eab46
5f021c885e2d6d8cba72efaa6e2d28155827fd3b4f12cbedf933c5eb27c9de868c536697e69439d9efef646f2
94363aa35142c3242cc81f98df104042abe9f4b393f9aa1cfd9469a7df8386ca4ba3c48035ca77a77d572cb30
9f17a335574ffb29b473e8c9fc2afaa23ad829ef0b8e015c8f9e7455c8a62e6b2eb27f881f333bfc46d76f387
ac644c2ed09045953a8291175ae10dca28e3781ce1c2f0da34c0385824ab0f2da6226f6195b7d0b328ff430c2
2f84fd5c3052a8ce07afef149e68dd9336717786dc2bb653946d268365eff2907e76a28d132a682ca57c6ffd3
b5b300052d6fe22b9c832d8b22c5850ff6c4666c70e367f7bcfa70401f2d398a95a62c38ac1b0ea8d62068613
b2b2051dc1af63dfc1eed0d35e0fd192b8d868815f0b77fa2c9bf1e4f77ca579c126de8701c9135e6a5c953c5
3ba937e971fdbff679bc711087747e79c2f6c67a1efc5b9c197ca3c24e05d5e43f3d841c4ab4d04749e6d8508
00ddce00b12ca4847560e8ba18d8225f3ef2b7b60aeb9327df5c653b489788068a7bdc63f2a36ca718b991ecb
4ee83fdba17facf3bd1675b54b56ff7bce00d25185bcde4bf76c17850e57ad423a4f35060ee3d861e24ed80f7
6499f9313c9939479ff0d9033d7a79666ee175c2566ff14858788f8164bd8fbae454368274602fc4af65a03dd
3644314d1f6cda2755993ea62faa553a1d355ae1bc3ba1b8f17e7e2975f46df1e54885e70a50bb355ac6e8c6a
9c36f21d079b44d2fe1133e1e2606bb417ee893f0d1a407c79e32ba85701e7f407116c73e04314bdfcf87f58b
c625d2c6c3d0a7894659c80a5f6840cf68734adab16234705c03cdb25a888f4791e94b6b1d29c80a66cca56bb
54bef1e7e1e2982cdac42c4aeff490f56c6878635ade2d2ada95fea1434564605e23e875b8608fb29113c2eef
87db16a6885402227e86d2828b26c29115c082ddfdf3ac1ceffeffa4a267bd3638c2cbc57b71a2fbdff4058d5
235025f69989f794b70feb5b955e31705ae4b81e9175fe4c4f1f369468f5edcc536409589efb2cab435b71245
76797cf593b34d06
```

Nous pouvons aussi tenter de récupérer juste une page du site :

```
Please give the page name : index.html
Here is the requested page :
<!DOCTYPE html []>
<html>
        <head>
                <meta charset="utf-8" />
        </head>
        <body>
                Here are the categories present on the website :
                        maths.html
                        info.html
                        physique.html
                        chimie.html
        </body>
        </head>
</html>
```

Maintenant que nous avons tout cela, il faut maintenant voir comment procéder. Pour rappel, nous voulons donc savoir quelle page notre ami a visité, et le challenge se nomme `LengthRevealingEncryption`. Possiblement, chaque page pourrait avoir une longueur différente, et le fait que le chiffrement révèle la longueur pourrait nous permettre de déterminer la page visitée juste avec la longueur (que nous avons) !
Nous pouvons donc lister les longueurs de toutes les pages :

```
'index.html' : 213
'maths.html' : 265
'info.html' : 261
'physique.html' : 258
'chimie.html' : 259
'maths/arithmetique.html' : 930
'maths/probabilites.html' : 689
'maths/analyse.html' : 849
'maths/suites.html' : 848
'maths/fractales.html' : 755
'info/bdd.html' : 724
'info/programmation.html' : 641
'info/algorithmique.html' : 880
'info/demoeffects.html' : 746
'info/graphes.html' : 934
'physique/gravite.html' : 613
'physique/pousseearchimede.html' : 980
'physique/frottements.html' : 697
'physique/ressorts.html' : 999
'chimie/nommagemolecules.html' : 851
'chimie/acidebase.html' : 732
'chimie/nucleaire.html' : 919
'chimie/equationsreaction.html' : 895
```

Et en récupérant la longueur des pages visitées par notre ami (en y divisant par 2), on obtient :

```
Page 1  : taille = 895 - 'chimie/equationsreaction.html'
Page 2  : taille = 755 - 'maths/fractales.html'
Page 3  : taille = 213 - 'index.html'
Page 4  : taille = 746 - 'info/demoeffects.html'
Page 5  : taille = 930 - 'maths/arithmetique.html'
Page 6  : taille = 849 - 'maths/analyse.html'
```

```
Page 7  : taille = 258 - 'physique.html'
Page 8  : taille = 641 - 'info/programmation.html'
Page 9  : taille = 213 - 'index.html'
Page 10 : taille = 697 - 'physique/frottements.html'
Page 11 : taille = 919 - 'chimie/nucleaire.html'
Page 12 : taille = 851 - 'chimie/nommagemolecules.html'
Page 13 : taille = 259 - 'chimie.html'
Page 14 : taille = 641 - 'info/programmation.html'
Page 15 : taille = 261 - 'info.html'
Page 16 : taille = 880 - 'info/algorithmique.html'
Page 17 : taille = 755 - 'maths/fractales.html'
Page 18 : taille = 849 - 'maths/analyse.html'
Page 19 : taille = 258 - 'physique.html'
Page 20 : taille = 848 - 'maths/suites.html'
```

Un programme pour le résoudre automatiquement serait le suivant :

```python
import math
import random
import os, sys, socket

server = "146.59.227.136"
port = 23413



# First, getting all of the pages on the website
pagesSeen = []
ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
try:
    ma_socket.connect((server, port))
except Exception as e:
    print("Problème de connexion", e.args)
    sys.exit(1)
# Getting data
data = ma_socket.recv(16384).decode("utf-8")
ma_socket.sendall(("1" + "\n").encode("utf-8"))
data = ma_socket.recv(16384).decode("utf-8")
pagesSeen = data.split(" : ")[1].split(", ")
pagesSeen[-1] = pagesSeen[-1].strip() # To remove \n\n at the end
print("List of pages on the website : " + str(pagesSeen))



# Now, getting the size of each page
pagesSizes = []
for page in pagesSeen :
    ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:
        ma_socket.connect((server, port))
    except Exception as e:
        print("Problème de connexion", e.args)
        sys.exit(1)
    # Getting data
    data = ma_socket.recv(16384).decode("utf-8")
    ma_socket.sendall(("3" + "\n").encode("utf-8"))
    # Getting data
    data = ma_socket.recv(16384).decode("utf-8")
    ma_socket.sendall((page + "\n").encode("utf-8"))
    # Getting data
    data = ma_socket.recv(16384).decode("utf-8")
    pagesSizes.append(data.find("</html>") + 7 - data.find("<!DOCTYPE"))
    # Closing the socket and adding the page seen
    ma_socket.close()
# Printing
print("Size of those pages : " + str(pagesSizes))



# Getting the pages seen by our friend
ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
try:
    ma_socket.connect((server, port))
except Exception as e:
    print("Problème de connexion", e.args)
    sys.exit(1)
# Getting data
data = ma_socket.recv(16384).decode("utf-8")
ma_socket.sendall(("2" + "\n").encode("utf-8"))
```

```
# Getting data
data = ma_socket.recv(16384).decode("utf-8")
while "#Page 20" not in data :
    data += ma_socket.recv(16384).decode("utf-8")
dataLines = data.split("\n")
pages = []
for i in range(0, len(dataLines)) :
    if dataLines[i][:5] == "#Page" :
        toStudy = dataLines[i].split(" : ")[1]
        length = len(toStudy) // 2
        found = False
        for j in range(0, len(pagesSizes)) :
            if pagesSizes[j] == length :
                found = True
                pages.append(pagesSeen[j])
                break
        if not found :
            print("One page was not found !")
# Printing the result
print("Pages found : " + str(pages))
ma_socket.close()



# Getting the flag
ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
try:
    ma_socket.connect((server, port))
except Exception as e:
    print("Problème de connexion", e.args)
    sys.exit(1)
# Getting data
data = ma_socket.recv(16384).decode("utf-8")
ma_socket.sendall(("4" + "\n").encode("utf-8"))
# Getting data
data = ma_socket.recv(16384).decode("utf-8")
ma_socket.sendall((",".join(pages) + "\n").encode("utf-8"))
# Getting data
data = ma_socket.recv(16384).decode("utf-8")
print(data)
# Closing connection
ma_socket.close()
```