Devoir Maison – Authentification et GPG

Exercice 1 :

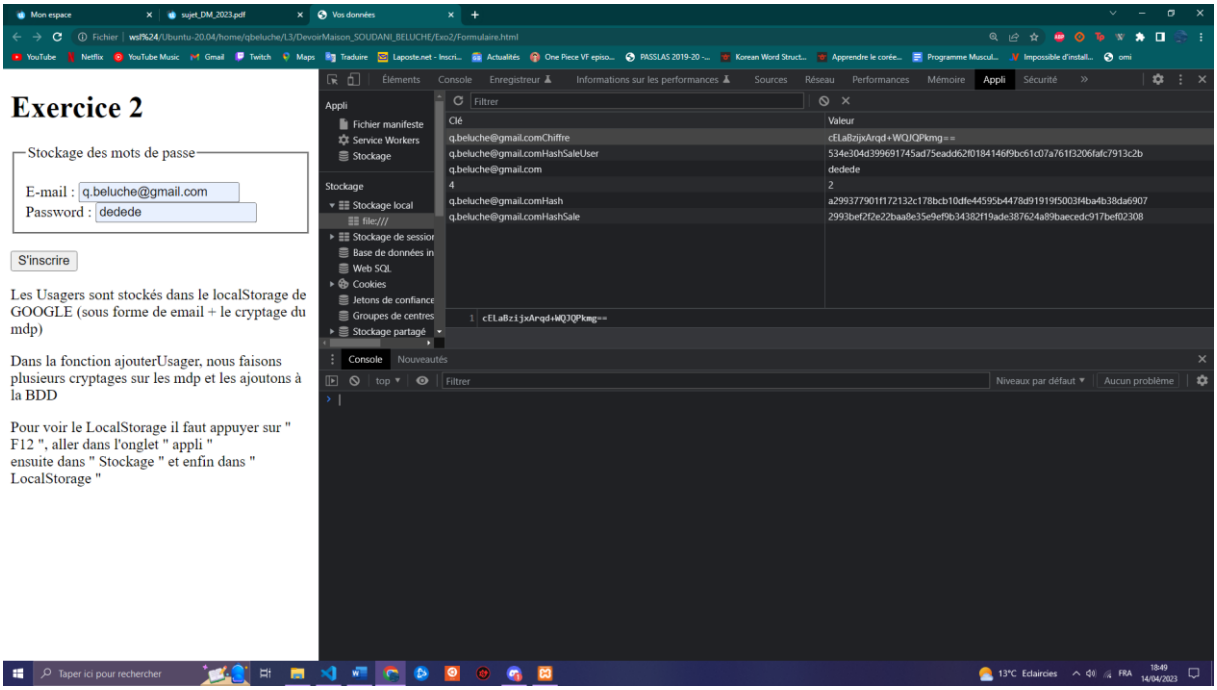
1)

- SHA1(beach.jpg)= f41023e03192f9cb9306ffa662ee5e7e7db89c5d
- SHA1(neach.jpg)= d6c725e161d23afb3957434f109bf5f1382e9fd9

2)

- SHA1(a.pdf)= e2081af21ac70cbca367e0608090de050427f286
- SHA1(b.pdf)= e2081af21ac70cbca367e0608090de050427f286

Exercice 2 :



Exercice 3 :

1)

L'algorithme de hachage utilisé est « MD5 » car il y a « 1 » et son sel est « AAAA » vu après le 1

Je trouve le même mot de passe :

openssl passwd -1 -salt AAAA ->

password : !!1331xxx ->

1AAAA$H9wXcd/WaaomJUgWKFspy

2)

```
#!/bin/bash

file='phpbb.txt'

myString='$1$CACA$XLWo4OqFFCYICqYrZ0y5i/'
echo "hash = $myString"
i=1
while read line; do

#Reading each line
hachage=$(openssl passwd -1 -salt CACA "$line")
if [ $hachage == $myString ]
then
    echo "MDP trouvé : $line"
fi
i=$((i+1))
if [ $((i%5000)) = 0 ]
then
    echo "i = $i"
fi
done < $file
```

Voici le script bash créé pour parcourir le fichier et tester chaque mot de passe et voir s'il correspond au mot de passe de myString rentré préalablement.

Pour 1BABA$DOzBWHNx08SgVSX/YuYvC/ :

« batman » est le mot de passe trouvé.

Pour 1CACA$XLWo4OqFFCYICqYrZ0y5i/ :

« enigma » est le mot de passe trouvé.

Exercice 4 :

1)

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --full-generate-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection? 2
DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 10d
Key expires at Sun Apr 23 16:23:57 2023 CEST
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: KeyQuentin
Email address: q.beluche@gmail.com
Comment: DevoirMaison
You selected this USER-ID:
    "KeyQuentin (DevoirMaison) <q.beluche@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: WARNING: some OpenPGP programs can't handle a DSA key with this digest size
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 50E6D9996B53692C marked as ultimately trusted
gpg: revocation certificate stored as '/home/qbeluche/.gnupg/openpgp-revocs.d/C0C83A1AB87E06ABD606B35050E6D9996B53692
C.rev'
public and secret key created and signed.

pub   dsa2048 2023-04-13 [SC] [expires: 2023-04-23]
      C0C83A1AB87E06ABD606B35050E6D9996B53692C
uid                      KeyQuentin (DevoirMaison) <q.beluche@gmail.com>
sub   elg2048 2023-04-13 [E] [expires: 2023-04-23]
```

2)

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --list-key
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   3  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: next trustdb check due at 2023-04-23
/home/qbeluche/.gnupg/pubring.kbx
---------------------------------
pub   rsa3072 2023-04-13 [SC] [expires: 2023-04-23]
      4F2DF71A72779F39E3EE34E05B3131D7C36E3612
uid           [ultimate] PhotoIdentite (https://pbs.twimg.com/profile_images/1275225080398860288/U8JO-80i_400x400.jpg
) <q.beluche@gmail.com>
uid           [ultimate] Onino <q.beluche@gmail.com>
uid           [ultimate] Quentin (Photo d'identité numérique) <q.beluche@gmail.com>
sub   rsa3072 2023-04-13 [E] [expires: 2025-04-12]

pub   rsa3072 2023-04-13 [SC] [expires: 2025-04-12]
      559F86A6A688E935E224DC2DE0ED0F652867134E
uid           [ultimate] RSA4096
sub   rsa3072 2023-04-13 [E]

pub   dsa2048 2023-04-13 [SC] [expires: 2023-04-23]
      C0C83A1AB87E06ABD606B35050E6D9996B53692C
uid           [ultimate] KeyQuentin (DevoirMaison) <q.beluche@gmail.com>
sub   elg2048 2023-04-13 [E] [expires: 2023-04-23]
```

(a) Voici l'empreinte de ma clé :

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --fingerprint C0C83A1AB87E06ABD606B35050E6D9996B53692
C
pub   dsa2048 2023-04-13 [SC] [expires: 2023-04-23]
      C0C8 3A1A B87E 06AB D606  B350 50E6 D999 6B53 692C
uid           [ultimate] KeyQuentin (DevoirMaison) <q.beluche@gmail.com>
sub   elg2048 2023-04-13 [E] [expires: 2023-04-23]
```

(b) Voici le changement de date d'expiration :

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --edit-key C0C83A1AB87E06ABD606B35050E6D9996B53692C
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

sec  dsa2048/50E6D9996B53692C
     created: 2023-04-13  expires: 2023-04-23  usage: SC
     trust: ultimate      validity: ultimate
ssb  elg2048/3D9B2FCD9E678021
     created: 2023-04-13  expires: 2023-04-23  usage: E
[ultimate] (1). KeyQuentin (DevoirMaison) <q.beluche@gmail.com>

gpg> expire 10d
Changing expiration time for the primary key.
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 10d
Key expires at Sun Apr 23 16:28:00 2023 CEST
Is this correct? (y/N) y

sec  dsa2048/50E6D9996B53692C
     created: 2023-04-13  expires: 2023-04-23  usage: SC
     trust: ultimate      validity: ultimate
ssb  elg2048/3D9B2FCD9E678021
     created: 2023-04-13  expires: 2023-04-23  usage: E
[ultimate] (1). KeyQuentin (DevoirMaison) <q.beluche@gmail.com>

gpg: WARNING: Your encryption subkey expires soon.
gpg: You may want to change its expiration date too.
gpg> save
```

(c) Voici l’ajout d’une photo d’identité numérique :

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --edit-key C0C83A1AB87E06ABD606B35050E6D9996B53692C
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   3  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: next trustdb check due at 2023-04-23
sec  dsa2048/50E6D9996B53692C
     created: 2023-04-13  expires: 2023-04-23  usage: SC
     trust: ultimate      validity: ultimate
ssb  elg2048/3D9B2FCD9E678021
     created: 2023-04-13  expires: 2023-04-23  usage: E
[ultimate] (1). KeyQuentin (DevoirMaison) <q.beluche@gmail.com>

gpg> adduid
Real name: PhotoIdentite
Email address: q.beluche@gmail.com
Comment: https://pbs.twimg.com/profile_images/1275225080398860288/U8JO-80i_400x400.jpg
You selected this USER-ID:
    "PhotoIdentite (https://pbs.twimg.com/profile_images/1275225080398860288/U8JO-80i_400x400.jpg) <q.beluche@gmail.c
om>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o

sec  dsa2048/50E6D9996B53692C
     created: 2023-04-13  expires: 2023-04-23  usage: SC
     trust: ultimate      validity: ultimate
ssb  elg2048/3D9B2FCD9E678021
     created: 2023-04-13  expires: 2023-04-23  usage: E
[ultimate] (1)  KeyQuentin (DevoirMaison) <q.beluche@gmail.com>
[ unknown] (2). PhotoIdentite (https://pbs.twimg.com/profile_images/1275225080398860288/U8JO-80i_400x400.jpg) <q.belu
che@gmail.com>

gpg> save
```

(d) Génération d'une sous clé RSA4096 avec une date d'expiration de 4 jours :

```
gpg> addkey
Please select what kind of key you want:
   (3) DSA (sign only)
   (4) RSA (sign only)
   (5) Elgamal (encrypt only)
   (6) RSA (encrypt only)
  (14) Existing key from card
Your selection? 4
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 4d
Key expires at Mon Apr 17 16:39:20 2023 CEST
Is this correct? (y/N) y
Really create? (y/N) y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

sec  rsa3072/5B3131D7C36E3612
     created: 2023-04-13  expires: 2023-04-23  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa3072/38AC371211E8BFBB
     created: 2023-04-13  expires: 2025-04-12  usage: E
ssb  rsa4096/1C3039FF40A9B1B9
     created: 2023-04-13  expires: 2023-04-17  usage: S
[ultimate] (1). PhotoIdentite (https://pbs.twimg.com/profile_images/1275225080398860288/U8JO-80i_400x400.jpg) <q.belu
che@gmail.com>
[ultimate] (2)  Onino <q.beluche@gmail.com>
[ultimate] (3)  Quentin (Photo d'identité numérique) <q.beluche@gmail.com>
```

(e) Modification des préférences en SHA256 et AES192 :

```
gpg> setpref SHA256 AES192
Set preference list to:
     Cipher: AES192, 3DES
     Digest: SHA256, SHA1
     Compression: ZIP, Uncompressed
     Features: MDC, Keyserver no-modify
Really update the preferences? (y/N) y
gpg: WARNING: no user ID has been marked as primary.  This command may
             cause a different user ID to become the assumed primary.
*
sec  rsa3072/5B3131D7C36E3612
     created: 2023-04-13  expires: 2023-04-23  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa3072/38AC371211E8BFBB
     created: 2023-04-13  expires: 2025-04-12  usage: E
ssb  rsa4096/1C3039FF40A9B1B9
     created: 2023-04-13  expires: 2023-04-17  usage: S
[ultimate] (1)  PhotoIdentite (https://pbs.twimg.com/profile_images/1275225080398860288/U8JO-80i_400x400.jpg) <q.belu
che@gmail.com>
[ultimate] (2)  Onino <q.beluche@gmail.com>
[ultimate] (3). Quentin (Photo d'identité numérique) <q.beluche@gmail.com>
```

(f) faire « save ».

4) Clé associée à l'e-mail de l'UL :

```
Real name: Quent
Email address: quentin.beluche6@etu.univ-lorraine.fr
Comment: Key associée à l'e-mail UL
```

5) Création d'un certificat de révocation :

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --gen-revoke 77D8456B1794AC1CC31DDF39BEBA1B2A13A13856

sec  dsa2048/BEBA1B2A13A13856 2023-04-13 Quent (Key associée à l'e-mail UL) <quentin.beluche6@etu.univ-lorraine.fr>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 0
Enter an optional description; end it with an empty line:
> Création d'un certificat de révocation
>
Reason for revocation: No reason specified
Création d'un certificat de révocation
Is this okay? (y/N) y
ASCII armored output forced.
-----BEGIN PGP PUBLIC KEY BLOCK-----
Comment: This is a revocation certificate

iKAEIBEIAEgWIQR32EVrF5SsHMMd3zm+uhsqE6E4VgUCZDgXKiodAENyw6lhdGlv
biBkJ3VuIGNlcnRpZmljYXQgZGUgcsOpdm9jYXRpb24ACgkQvrobKhOhOFZeCwD+
I8fr7gCMaFYgm0PsG6gy010pmQJnwWPOrKWRTLijFRIBAJWDhqrRSCyd7Dnyv5BM
NghtFXx5Rbzf5SOUGscMINBg
=jhhs
-----END PGP PUBLIC KEY BLOCK-----
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data and make it available to others!
```

6)Pour format binaire :

gpg --export --output nomDeLaClé.gpg --armor

Pour format texte :

gpg --export --output nomDeLaClé.asc --armor

7)

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --list-key
/home/qbeluche/.gnupg/pubring.kbx
---------------------------------
pub   dsa2048 2023-04-13 [SC] [expires: 2023-04-23]
      77D8456B1794AC1CC31DDF39BEBA1B2A13A13856
uid           [ultimate] Quent (Key associée à l'e-mail UL) <quentin.beluche6@etu.univ-lorraine.fr>
sub   elg2048 2023-04-13 [E] [expires: 2023-04-23]

pub   dsa2048 2023-04-14 [SC] [expires: 2023-04-24]
      C395377C964022B874109A724DC0648E015BCA33
uid           [ultimate] Younes (clePourDevoirMaison) <younes.soudani@gmail.com>
sub   elg2048 2023-04-14 [E] [expires: 2023-04-24]

qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg -e -r C395377C964022B874109A724DC0648E015BCA33 message.txt
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg -s message.txt.gpg
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$
```

8) Le chiffrement du fichier mp3 donne un fichier mp3.asc plus gros, tandis que le chiffrement d'un fichier texte va donner un fichier texte crypté moins gros.

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --encrypt --armor Leonard.mp3
You did not specify a user ID. (you may use "-r")

Current recipients:

Enter the user ID.  End with an empty line: 77D8456B1794AC1CC31DDF39BEBA1B2A13A13856

Current recipients:
elg2048/D7DA102435454DD6 2023-04-13 "Quent (Key associée à l'e-mail UL) <quentin.beluche6@etu.univ-lorraine.fr>"

Enter the user ID.  End with an empty line:
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ ls -lh Leonard*
-rw-r--r-- 1 qbeluche qbeluche 5.4M Apr 14 16:48 Leonard.gpg
-rw-r--r-- 1 qbeluche qbeluche 6.5M Apr 14 16:48 Leonard.mp3
-rw-r--r-- 1 qbeluche qbeluche 7.3M Apr 14 16:51 Leonard.mp3.asc
-rw-r--r-- 1 qbeluche qbeluche  158 Apr 14 16:48 LeonardRichter-CrimeTime.mp3:Zone.Identifier
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --output Leonard.decrypted --decrypt Leonard.mp3.asc
gpg: encrypted with 2048-bit ELG key, ID D7DA102435454DD6, created 2023-04-13
      "Quent (Key associée à l'e-mail UL) <quentin.beluche6@etu.univ-lorraine.fr>"
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --sign Leonard.mp3
```

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --verify Leonard.mp3.gpg
gpg: Signature made Fri Apr 14 16:53:07 2023 CEST
gpg:                using DSA key 77D8456B1794AC1CC31DDF39BEBA1B2A13A13856
gpg: Good signature from "Quent (Key associée à l'e-mail UL) <quentin.beluche6@etu.univ-lorraine.fr>" [ultimate]
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$
```

9) Le message chiffré est plus gros à chaque fois, quand il est déchiffrable par les deux destinataires il est encore plus gros.

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --encrypt --armor -r younes.soudani@gmail.com MemeMessag
eAChiffre.txt
```

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --encrypt --armor -r quentin.beluche6@etu.univ-lorraine.
fr MemeMessageAChiffreYounes.asc
```

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --encrypt --armor -r quentin.beluche6@etu.univ-lorraine.
fr MemeMessageAChiffreYounes.asc
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ ls -lh MemeMessage*
-rw-r--r-- 1 qbeluche qbeluche    4 Apr 14 17:02 MemeMessageAChiffre.txt
-rw-r--r-- 1 qbeluche qbeluche 1.9K Apr 14 17:06 MemeMessageAChiffreYounes+Quentin.asc
-rw-r--r-- 1 qbeluche qbeluche  898 Apr 14 17:02 MemeMessageAChiffreYounes.asc
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --output MemeMessageAChiffre.decrypted --decrypt MemeMes
sageAChiffreYounes+Quentin.asc
gpg: encrypted with 2048-bit ELG key, ID D7DA102435454DD6, created 2023-04-13
      "Quent (Key associée à l'e-mail UL) <quentin.beluche6@etu.univ-lorraine.fr>"
```

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --symmetric --cipher-algo AES256 MemeMessageAChiffre.txt
```

```
qbeluche@DESKTOP-JI5R194:~/L3/DevoirMaison_SOUDANI_BELUCHE$ gpg --symmetric --cipher-algo AES256 MemeMessageAChiffreYoun
es+Quentin.asc
```

On peut constater que les deux fichiers vont réduire de taille.

Exercice 5 :

1) b) Université : TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 .

RSA avec Clé de 256 bits.

TLS 1.2

Emis le Lundi 3 avril 2023 à 2:00:00

Expire le Mercredi 3 avril 2024 à 1:59:59

Algo de chiffrement : SHA-256, SHA-1, SHA-256 ECDSA, SHA-256 avec chiffrement RSA version 3

Numéro de série : 31

Outlook : TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

RSA avec Clé de 256 bits.

TLS 1.2

Emis le mardi 26 Juillet 2023 à 2:00:00

Expire le mercredi 26 Juillet 2023 à 01:59:59

Algo de chiffrement : SHA-256, SHA-1, SHA-256 ECDSA, SHA-256 avec chiffrement RSA version 3

Numéro de série : 16

gmail : TLS_AES_128_GCM_SHA256.

Elliptic Curve Clé de 128 bits .

TLS 1.3.

Emis le mardi 28 Mars 2023 à 18:47:33

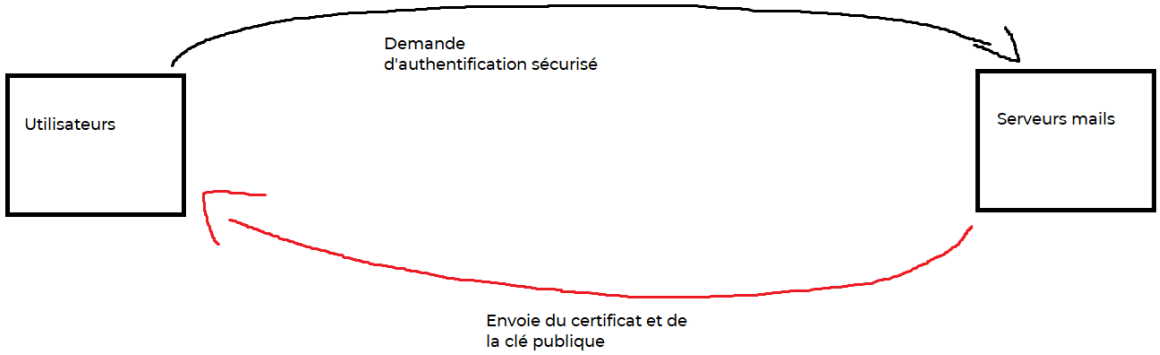
Expire le mardi 20 Juin 2023 à 18:47:33

Algo de chiffrement : SHA-256, SHA-1, SHA-256 ECDSA, SHA-256 avec chiffrement RSA version 3

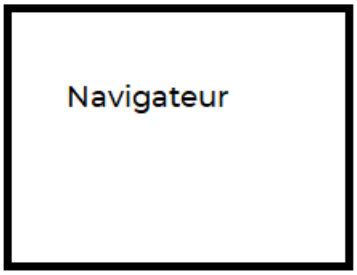
Numéro de série : 6

2.c)

1)



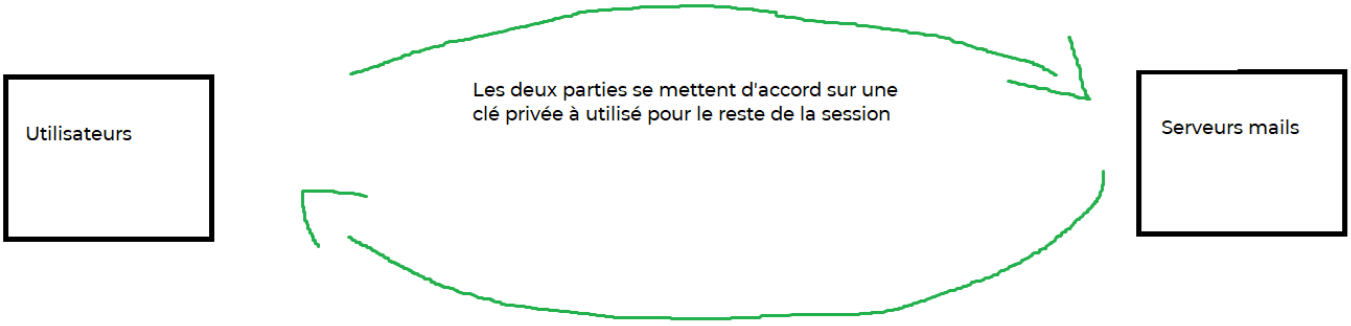
2)



Le navigateur vérifie ensuite dans sa BDD si :
- Le certificat est valide ?
- La signature du certificat

Le navigateur va ensuite chercher la clé publique de l'autorité qui a certifié le certificat.

3)



b) Dans le cas d'une connexion à un service de wifi publique, le navigateur envoie une demande d'authentification qui est accepté ou non. Il ne se passe rien par rapport a la clé privée qui pourrait être échangé entre les deux étant donnée qu'elle n'existe pas vue l'absence de certificats.

QUESTION 3

a) L'intérêt de protéger une connexion par https est de rendre quasi-impossible le déchiffrement des données pendant leur itinéraire entre l'ordinateur de l'utilisateur et le serveur du site ou l'on souhaite se connecter.

De plus même si le chiffrement était remis en question, certaine propriété cryptographique garantissent la confidentialité des échanges passés

b) On cherche donc à protéger les données (entre autre nos identifiants) qui transitent entre notre ordinateur et le serveur de connexion. Un attaquant qui écouterait les communications pourrait, sans le chiffrement, les récupérer et ainsi avoir accès à nos comptes.

Il pourrait aussi modifier ou simplement récupérer les informations qui transitent.

Il s'agirait d'une attaque de type « l'homme du milieu ».

c) Le fournisseur mail en HTTPS ne reçoit donc qu'une demande d'authentification puis après un accord entre les deux, la clé privée.

Tant dis qu'en HTTP la connexion n'étant pas confidentiel, il n'y a pas de clé privée vue qu'il n'y a pas de chiffrement ni de certificats.