TP1

Mathys Domergue

Début réaliser avec Askel Caubel jusqu'à la parti 5

1. Fonction de hachage

1.1 Calcul d'un condensé à l'aide de MD5

Exercie 1

Le hash est un est une phrase en haxadécimal :

```
b10a8db164e0754105b7a99be72e3fe5
```

Exercice 2

On obtient 16 condensés différents

Exercice 3

Il est possible d'avoir du confilt avec MD5 car les possbilités sont moindre par rapport au SHa256 ou SHa512.

1.2 Verification des propriétés du condeasé

Exercice 4

Le hash reste le meme car c'est le contennu du fichier qui est hachée.

Exercice 5

Le hash diffère du premier. La simple modification d'un caractère change totalement le hache

```
b9be3ef4018be19f248f6f8e63b9e006
```

Exercice 6

Un fichier binaire n'est qu'un fichier remplit de 1 et de 0. Il est donc possible de faire un hache d'un fichier binaire.

Test effectuer sur un fichier binaire d'un code c++ :

```
aksel@RedArch ~/Documents/GitHub/R401-Architectures_securisee/TP-1/file-to-hash main md5sum V0.2_ESP-Fenetre-rainSensor.ino.bin

>>> 7a3d7d2be3c6bf8c84e211786a130a5b  V0.2_ESP-Fenetre-rainSensor.ino.bin
```

2. Clefs de chiffrement

2.1 Génération de clefs

Exercice 7 :

```
aksel@RedArch ~/.gnupg gpg --list-keys /home/aksel/.gnupg/pubring.kbx
```

pub rsa1024 2023-02-27 [SC] F54AB41B512674472ACDB54136D043ED57514426 uid [ultimate] AkselCaubel (A Network & Telecom Student) <aksel.caubel@etu.umontpellier.fr> sub rsa1024 2023-02-27 [E]

Les différents champs observables sont :

```
pub -> Il donne l'algorithme utilisé, la taile de bits utilisé avec la date création de la clef publique.
uid -> Le UID Permet de donner l'identifiant de la personne avec l'éventuelle description rentré.
sub -> Il donne l'algorithme utilisé, la taile de bits utilisé avec la date création de l'empreinte de création.
```

Exercice 8

Pour regarder les différentes clef que l'on possède sur notre ordinateur, on peut utiliser les différentes options ci dessous (source : gpg --help)

```
-k, --list-keys             list keys
     --list-signatures       list keys and signatures
     --check-signatures      list and check key signatures
     --fingerprint           list keys and fingerprints
-K, --list-secret-keys      list secret keys
```

Pour lister les clefs privé il faut donc faire un ***gpg -K*** :

```
aksel@RedArch ~/.gnupg gpg -K
/home/aksel/.gnupg/pubring.kbx
```

sec rsa1024 2023-02-27 [SC] F54AB41B512674472ACDB54136D043ED57514426 uid [ultimate] AkselCaubel (A Network & Telecom Student) <aksel.caubel@etu.umontpellier.fr> ssb rsa1024 2023-02-27 [E]

Ce qui permet de faire le lien entre les différentes clefs est l'empreinte

2.2 Diffusion de la clef publique

Exercice 9

le contenu du fichier est :

```
aksel@RedArch ~/.gnupg cat akselCaubel.key
-----BEGIN PGP PUBLIC KEY BLOCK-----

mI0EY/ygUgEEAK+h9c8qyl9CtQnNj3LxPBsuDVK1aIDrvdSCyEImaeoqmxZKVdjo
3gxudnvcQlxCnjrryyvS8YtOjpLoBpoSfCXhEb08vOWf8NeQm/N/1Tccqiwz3riZ
GnvL0HEIVjPP3v7MJvib/w5NERaEaVTDQIisMmcpdvGR4GICU9cmQRaNABEBAAG0
TEFrc2VsQ2F1YmVsIChBIE5ldHdvcmsgJiBUZWxlY29tIFN0dWRlbnQpIDxha3Nl
bC5jYXViZWxAZXR1LnVtb250cGVsbGlci5mcj6IzgQTAQoAOBYhBPVKtBtRJnRH
Ks21QTbQQ+1XUUQmBQJj/KBSAhsDBQsJCAcCBhUKCQgLAgQWAgMBAh4BAheAAAoJ
EDbQQ+1XUUQm8oAD/A865cwH0MjNswfXp1+BpqpN6jbYzz3F4EjepXgpfT1ijlB8
oT6Duuld5dEgRr49bz8LJRYfiql4cIoXUxEwVXO7PurMbul+jBPtX/due69l/G9X
XqQUbDG/5VQsqZ5pPdorUiQsz5o79UhNrcw06j96rBciDqQAq79JOGghZCv5uI0E
Y/ygUgEEAJ8yjx2a4Z9EQmUcEj2W+DwveAOKQyr4dRMKC0OrT2OEjsY8J0jzrSpg
W4+S7M7cy/5ePPvWDkyXvoQ9yxokSlZF+moy7uLHztBsTarUfwru+6froQ8RiuFU
lrtATuUu9th58A2IDJ4Rb/dBBa4iZUdkRsbIQ9sgKsG+nDyClpphABEBAAGItgQY
AQoAIBYhBPVKtBtRJnRHKs21QTbQQ+1XUUQmBQJj/KBSAhsMAAoJEDbQQ+1XUUQm
ZkgD/A4tpmq77Pa8VCOatxj2+4QSVp35tGMvqa77TIWzgbGcZ+R5wHKL8zyy8Irx
h+LK9Qd039Svx3V5BCfyCRsbha6cyOIyArNO1Uxc0C1jwAmz/ykA09a5b74ClH0r
JOBYnxrTnQQqw4KSh4xFbOxzXAyxsfG2P2lg3FpRcUcFZ3cHmI0EY/y19gEEAKoI
m8JCYiRfu24OsfseVVwOJI64FN0oKYpqkDgrXfgCUMT2x0NxUxrDAksOqssjei4F
0tNR+5TIyjmhBpNr1P8TKcYGLGUNwU5PTkd0BaypJMdCHS2BDIV+04x67KFwwxRl
6DFgzls3HAuuvzEsYn65EWEBeXOwsX5aW7vJFBt3ABEBAAG0TEFrc2VsQ2F1YmVs
IChBIE5ldHdvcmsgJiBUZWxlY29tIHN0dWRlbnQpIDxha3NlbC5jYXViZWxAZXR1
LnVtb250cGVsbGlci5mcj6I1AQTAQoAPhYhBNB2LEQz04V5+kYjUic/+iJecsyI
BQJj/LX2AhsDBQkAAVGABQsJCAcCBhUKCQgLAgQWAgMBAh4BAheAAAoJECc/+iJe
csyIi3AD/i43dPgNKJ2gRhjR7Wb0B8CMOvdCojAmE/jFbspXZEr72vDmEYMl1l0T
YSqAvXDv3AqU+zYBQXAlMen6W1bKsQ0yNajxDHzHOVyZubsyH8TJOBtP64WUy8Xu
BNB2LEQz04V5+kYjUic/+iJecsyIBQJj/LX2AhsMBQkAAVGAAAoJECc/+iJecsyI
WmUD/jigvK9/rcVN8OOLY3ItHUxXqeo2tGoqH4P1mNWD0MEk6pHftdCWG/sNzxPx
bLm/qPze8JJ9g+6hzNFhOTHAESXUDl5fcwzlGs7I+KWUSUq+brFGfocgQUk1Lyol
YqGQmnROwj4YeFOyz1FU2HbeMy/eXOf7oxH//1WQ4j3G7bQ1
=vhyg
-----END PGP PUBLIC KEY BLOCK-----
```

Exercice 10 :

```
aksel@RedArch  ~/.gnupg  gpg --export-secret-keys --armor >
akselCaubelPriv.key
aksel@RedArch  ~/.gnupg  l

total 44K

drwx------  4 aksel aksel 4,0K 27 févr. 15:12  .

drwxr-x--- 33 aksel aksel 4,0K 27 févr. 15:12  ..

-rw-rw-r--  1 aksel aksel 2,1K 27 févr. 15:02  akselCaubel.key

-rw-rw-r--  1 aksel aksel 4,1K 27 févr. 15:12  akselCaubelPriv.key
```

3. Chiffrage d'un fichier

Exercice 11 :

Pour pouvoir envoyer un message à mon binôme, il faut avoirla clé publique de mon binôme.

```
aksel@RedArch  ~/.gnupg  gpg --import mathysDomergues.key

gpg: key 9FD5FA5848CD98B1: public key "testt <mail@bonjour.com>" imported

gpg: key 10B47788530F4DB6: public key "Mathys Domergue
<mathys.domergue@etu.umontpellier.fr>" imported
gpg: Total number processed: 2

gpg:               imported: 2
```

Exercice 12

```
pub   rsa3072 2023-02-27 [SC] [expires: 2025-02-26]

      3DD0B9ABE9D72E08BDAC7B0610B47788530F4DB6

uid           [ unknown] Mathys Domergue
<mathys.domergue@etu.umontpellier.fr>
sub   rsa3072 2023-02-27 [E] [expires: 2025-02-26]



aksel@RedArch  ~/.gnupg
```

Exercice 13 :

```
aksel@RedArch ~/.gnupg gpg --decrypt toto.txt.asc > toto.txt

gpg: encrypted with 1024-bit RSA key, ID FBEA4C286A0D7DAE, created 2023-
02-27
      "AkselCaubel (A Network & Telecom student)
<aksel.caubel@etu.umontpellier.fr>"
aksel@RedArch ~/.gnupg cat toto.txt

hello word !
```

Exercice 14

Il est bien l'équivalent du fichier initial

4 Signature numérique

4.1 Signature d'un fichier

Exercice 15

```
aksel@RedArch ~/.gnupg gpg --detach-sign --clearsign Mytoto.txt

File 'Mytoto.txt.asc' exists. Overwrite? (y/N) y

aksel@RedArch ~/.gnupg cat Mytoto.txt.asc

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA512



Hello Mate !

-----BEGIN PGP SIGNATURE-----



iLMEAQEKAB0WIQT1SrQbUSZ0RyrNtUE20EPtV1FEJgUCY/y+oAAKCRA20EPtV1FE

JilHA/sHNZ3qcMfvXMt2nCia6ia9UTqxLCVl3KlAL1uttJ+BtFIPQV70YvnyXKpm

FHPmV82EruGE7yG+wLOe6ZEUuwZd/fJC64VTCEh7bCZ57VELH8c7mP/kPsUZWL9M

vFFQiFgwuKG6EMh5YAEyaaA2N491O4lT1fgcToFqE4lczQJCsg==

=cX9l
```

```
-----END PGP SIGNATURE-----
```

Exercice 16

```
aksel@RedArch  ~/.gnupg  gpg --verify Mytoto.txt.asc
gpg: Signature made lun. 27 févr. 2023 15:30:56 CET
gpg:                using RSA key F54AB41B512674472ACDB54136D043ED57514426
gpg: Good signature from "AkselCaubel (A Network & Telecom Student)
<aksel.caubel@etu.umontpellier.fr
>" [ultimate]
```

4.2 Signature d'une clef publique

Exercice 17

```
aksel@RedArch  ~/.gnupg  gpg --sign-key
3DD0B9ABE9D72E08BDAC7B0610B47788530F4DB6

pub  rsa3072/10B47788530F4DB6
     created: 2023-02-27  expires: 2025-02-26  usage: SC
     trust: unknown       validity: unknown
sub  rsa3072/7A5AAC8D2BA6FBBA
     created: 2023-02-27  expires: 2025-02-26  usage: E
[ unknown] (1). Mathys Domergue <mathys.domergue@etu.umontpellier.fr>



pub  rsa3072/10B47788530F4DB6
     created: 2023-02-27  expires: 2025-02-26  usage: SC
     trust: unknown       validity: unknown
 Primary key fingerprint: 3DD0 B9AB E9D7 2E08 BDAC  7B06 10B4 7788 530F
4DB6
```

```
     Mathys Domergue <mathys.domergue@etu.umontpellier.fr>



This key is due to expire on 2025-02-26.

Are you sure that you want to sign this key with your

key "AkselCaubel (A Network & Telecom Student)
<aksel.caubel@etu.umontpellier.fr>" (36D043ED57514426)


Really sign? (y/N) y


 aksel@RedArch  ~/.gnupg 
```

5. Utilisation d'un certificat

Exercice 18

1378	4.735553202	10.214.0.198	18.159.99.25	TCP	66	56016 → 443 [ACK] Seq=11487 Ack=1368 Win=501 Len=0 TSval=2674262373 TSecr=2123789827
1379	4.739522138	52.84.45.24	10.214.0.198	TLSv1.2	645	Application Data
1380	4.739549925	10.214.0.198	52.84.45.24	TCP	66	40814 → 443 [ACK] Seq=3458 Ack=1160 Win=501 Len=0 TSval=3743348857 TSecr=3680930003
1381	4.760188756	10.214.0.198	178.250.1.25	TCP	74	53184 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=737283567 TSecr=0 WS=128
1382	4.784760707	178.250.1.25	10.214.0.198	TCP	74	443 → 53184 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3221620290 TSecr=737283567 WS=1024
1383	4.784857023	10.214.0.198	178.250.1.25	TCP	66	53184 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=737283592 TSecr=3221620290
1384	4.792273656	10.214.0.198	178.250.1.25	TLSv1.3	749	Client Hello
1385	4.792533068	10.214.0.198	178.250.1.25	TLSv1.3	72	Change Cipher Spec
1386	4.792695639	10.214.0.198	178.250.1.25	TLSv1.3	236	Application Data
1387	4.817244048	178.250.1.25	10.214.0.198	TCP	66	443 → 53184 [ACK] Seq=1 Ack=860 Win=64512 Len=0 TSval=3221620322 TSecr=737283599
1388	4.817509567	178.250.1.25	10.214.0.198	TLSv1.3	308	Server Hello, Change Cipher Spec, Application Data, Application Data
1389	4.817523603	10.214.0.198	178.250.1.25	TCP	66	53184 → 443 [ACK] Seq=860 Ack=243 Win=64128 Len=0 TSval=737283625 TSecr=3221620323
1390	4.818022018	10.214.0.198	178.250.1.25	TLSv1.3	150	Application Data, Application Data
1391	4.818404522	10.214.0.198	178.250.1.25	TLSv1.3	608	Application Data
1392	4.818431250	10.214.0.198	178.250.1.25	TLSv1.3	132	Application Data

Exercice 19

Le problème pour les admin est qu'ils n'ont plus accés au données