Compte rendue du TP 2 Crypto

Introduction au empreinte cryptographique MD5 et SHA-1

1,2,3,4- création du fichier1 puis fichier 2 et leur calcul md5 respectif

```
administrateur@Debian-12-Bookworm:~$ echo bonjour > fichier2
administrateur@Debian-12-Bookworm:~$ md5sum fichier1
94baaad4d1347ec6e15ae35c88ee8bc8  fichier1
administrateur@Debian-12-Bookworm:~$ md5sum fichier2
94baaad4d1347ec6e15ae35c88ee8bc8  fichier2
administrateur@Debian-12-Bookworm:~$
```

5- le résumé du fichier1 et fichier2 sont identiques parce que ce sont peut être des enveloppes différentes mais le message qu'ils contiennent (bonjour), est identique.

6- création et résumé du fichier3 avec le message bonjour1

```
administrateur@Debian-12-Bookworm:~$ echo bonjour1 > fichier3
administrateur@Debian-12-Bookworm:~$ md5sum fichier3
f5acb92e2ac2c8403f8503c552a1d659  fichier3
```

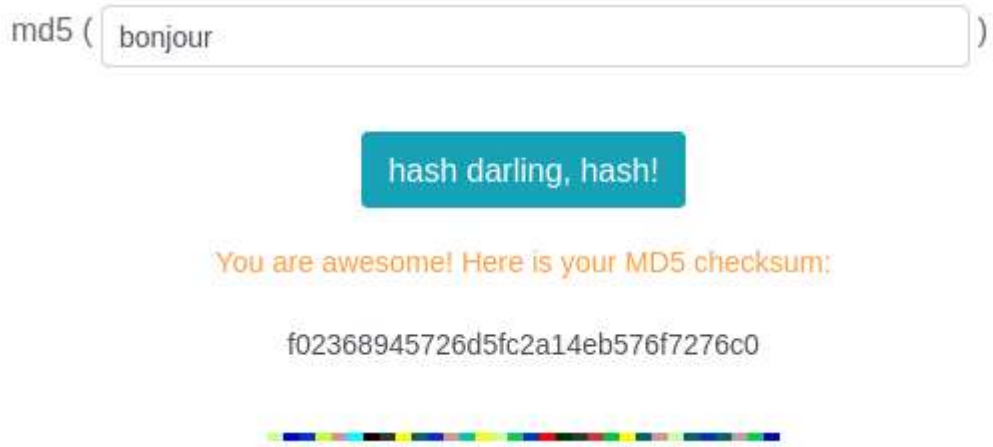
On remarque un résultat différent des deux autres fichier1 et fichier2 parce que leur message était simplement "bonjour", cela correspond donc à mes attentes par rapport à la question 5.

7- Exécution de la commande : echo bonjour | md5sum

```
administrateur@Debian-12-Bookworm:~$ echo bonjour | md5sum
94baaad4d1347ec6e15ae35c88ee8bc8  -
```

J'en conclus donc que cette commande permet l'obtention de l'empreinte MD5 d'un mot sans pour autant qu'il soit contenu dans un fichier.

8- Calcul du MD5 sur le site proposer



La réponse n'est pas identique, ce qui va à l'encontre de ce que je pensais.

9- Exécution de la commande : echo -n bonjour | md5sum

```
administrateur@Debian-12-Bookworm:~$ echo -n bonjour | md5sum
f02368945726d5fc2a14eb576f7276c0  -
```

10- le résultat est bien identique grâce à l'option -n, qui a pour utilité de retirer le saut de ligne et ainsi donnée le hash du string “bonjour” sans retour à la ligne.

Empreinte SHA1

11,12- Exécution des différentes commandes:

```
administrateur@Debian-12-Bookworm:~$ sha1sum fichier4
e7bc546316d2d0ec13a2d3117b13468f5e939f95  fichier4
```

13,14- Exécution des commandes “echo bonjour > fichier4” et “sha1sum fichier4”

```
administrateur@Debian-12-Bookworm:~$ echo bonjour > fichier5
administrateur@Debian-12-Bookworm:~$ sha1sum fichier5
e7bc546316d2d0ec13a2d3117b13468f5e939f95  fichier5
```

15- Ces deux fichiers ont une empreinte identique car, comme pour md5, sha1 convertis les mots contenu et/ou demander et non les enveloppes (ici, “fichier4” et “fichier5”)

16- Contenu de l’expérience précédente sous md5 et de la logique, le résultat correspond à mes attentes.

```
administrateur@Debian-12-Bookworm:~$ echo bonjour1 > fichier6
administrateur@Debian-12-Bookworm:~$ sha1sum fichier6
c83904636c6d95cd84e2e298e1d7298e966aed98  fichier6
```

Comparez les résumés de “sum”, “md5sum”, “sha1” et “sh512sum”.

```
administrateur@Debian-12-Bookworm:~$ sum fichier3
55386     1 fichier3
administrateur@Debian-12-Bookworm:~$ md5sum fichier3
f5acb92e2ac2c8403f8503c552a1d659  fichier3
administrateur@Debian-12-Bookworm:~$ sha1sum fichier3
c83904636c6d95cd84e2e298e1d7298e966aed98  fichier3
administrateur@Debian-12-Bookworm:~$ sha512sum fichier3
b137e593a9bc3632f2d963dd1105e5ccf072b119aaab2b7e7e04e6cd2e806031d8f820d207bb7420916a106f1b427508b5d2be60
5bc723706ea367e9d8c7e780  fichier3
```

17- J’en conclus que selon la commande choisie, la longueur de l’empreinte sera différente et donc également la complexité de son déchiffrage.

20- Par rapport à la longueur du condensé je dirais qu’il s’agit du sha256, mais je n’ai malheureusement pas réussi à trouver la méthode pour obtenir le résultat demandé.