Principe

Objet : disposer de moyen pour vérifier la bonne "*identité*" d'un fichier
Contexte : réception d'un fichier dont on souhaite être rassuré en terme d'ouverture et/ou d'exécution
Modalité : utilisation des outils natifs et/ou de logiciels adaptés.

Le Hash

Ce type de fonction est très utilisé en cryptographie, principalement dans le but de réduire la taille des données à traiter par la fonction de cryptage. En effet, la caractéristique principale d'une fonction de *hachage* est de produire un *haché* des données, c'est-à-dire un condensé de ces données. Ce condensé est de taille fixe, dont la valeur diffère suivant la fonction utilisée.

Hein ?

=

```
79054025
255fb1a2
6e4bc422
aef54eb4
```

Exemple avec les **empreintes digitales** : dans la perception que nous en avons à l'heure actuelle, une empreinte digitale est unique et représente un individu d'une façon si certaine que nous pouvons la qualifier de sure.

Pourtant la connaissance de cette empreinte ne permet pas **à elle-seule** de remonter à l'individu, ni de reconstituer cet individu. Il faut que la **correspondance ait été préalablement établie dans une base de données** pour que l'identification puisse avoir lieu par comparaison

C'est exactement ce genre de propriété que présente une fonction de hashage. Il s'agit bien d'un condensé de données pour un fichier. On parle alors abusivement de son empreinte digitale, en ce sens qu'un seul hash ne peut exister pour un fichier.

Disposer du hash permet de vérifier avec la source officielle (si digne de ce nom) que le fichier est bien celui qu'il prétend être.

Les différents hash

MD4 et **MD5** (Message Digest) développées par Ron Rivest. MD5 produit des hachés de 128 bits en travaillant les données originales par blocs de 512 bits. Il tend à disparaître pour des raisons de sécurité.

SHA-1 (Secure Hash Algorithm 1), - mis au point par la NSA- est basé sur MD4. Il

fonctionne également à partir de blocs de 512 bits de données et produit par contre des condensés de 160 bits en sortie. Il nécessite donc plus de ressources que MD5.

SHA-2 (Secure Hash Algorithm 2) est destiné à remplacer SHA-1. Les différences principales résident dans les tailles de hachés possibles : 256, 384 ou 512 bits. Il est la nouvelle référence en termes de fonction de hachage.

Et ça évolue encore ?
Oui donc à chaque fois, différents contrôles sont à faire selon les données sources transmises.
A noter qu'en février 2016 MD5 est moins utilisé au profit de SHA-1 et demain SHA-2 et après demain SHA-3 etc..

Modalités

Prennons l'exemple d'un fichier à télécharger. Le site de l'éditeur publie son *hash* et la méthgode utilisée. Une fois le fichier rappatrié, il suffit de vérifier le *hash* du fichier avec ce qu'annonce l'éditeur ou la source fiable.

Selon les sytèmes d'exploitation les outils sont plus ou moins intégrés (sous forme de ligne de commande) ou à récupérer via différents logiciels en open source ou sous licence payante.

Sous OS X

Pour un contrôle **SHA-1**, on ouvre **Terminal** et on saisit la commande comme suit :

```
openssl sha1 /chemin/complet/vers/le/fichier
```

Soit si le fichier est paint.dmg et qu'il se trouve dans mes téléchargements

```
openssl sha1 ~/Downloads/Paint.dmg
```

Cette commande renvoie alors le résultat suivant :

```
SHA1(/Users/username/Downloads/Paint.dmg)
=07272d863ab77113e38e6ce3878c2162feb4893e

07272d863ab77113e38e6ce3878c2162feb4893e est l'identité du fichier concerné.
```

Pour un contrôle **MD5 :**

```
openssl md5 /chemin/complet/vers/le/fichier
```

Pour un contrôle **SHA256** :

```
openssl dgst -sha256 /chemin/complet/vers/le/fichier
```

A noter qu'il existe des applications permettant de disposer de ces informations. Je préconise FileListExport qui permet de faire des exportations des caractéristiques de fichiers vers un fichier .CSV ou Excel.

Sous Windows

Il faut récupérer l'outil de vérification en ligne de Microsoft. Puis en mode ligne de commande (touche Windows + R) on frappe :

```
FCIV -md5 -sha1 Chemin\Nomfichier.ext
```

On obtient le calcul du hash en MD5 et en SHA-1. On peut utiliser un outil en partie gratuit comme celui disponible ICI.

Sous IOS

Il existe différentes applications. Une application comme Filecheck permet de vérifier tous fichiers sur IOS.

Sous Androïd

Il existe différentes applications. AFV me semble une bonne solution pour les smartphones et tablettes de chez Google.

Sous Linux/Unix.

Il s'agit d'une ligne de commande normalement intégrée à l'OS.

```
md5sum Chemin\Nomfichier.ext
```

```
sda1sum Chemin\Nomfichier.ext
```

A noter via **FileStation** d'un Synology (DSM v5.x) le **calcul MD5 est disponible en propriété**.

En ligne

On peut faire un calcul de fichier (max 4 Giga) en allant sur ce site : http://onlinemd5.com

Utilisation courante

- **Source sur le Net** : le hash est affiché (pas toujours la cas hélas), je télécharge, je calcule je vérifie.

- source **personnelle** : je calcule le hash j'envoie le fichier en indiquant son hash. La personne le réceptionne et fait le clacul de son côté sur la PJ.[*technique souvent utilisée pour les transferts monétiques via un ERP*]

- vérification **locale** : je lance une vérification de plusieurs fichiers systèmes importants ou majeurs pour moi, j'isole ces informations et je vérifie de temps à autre, en cas de doute. [*Attention s'il s'agit de fichiers systèmes les mises à jour de l'OS peuvent les modifier.*] Cette technique permet de vérifier qu'il n'ya pas de corruption sur les fichiers concernés.

Étiquettes : Bureautique IOS Mac osX SYNO WEB Windows