Simple File Verification: Vérifier l'intégrité ou l'authenticité de fichiers

La "File Verification" est une procédure pour vérifier l'intégrité ou l'authenticité de fichiers à l'aide d'un algorithme correspondant. Lors de la "Simple File Verification" (SFV), on utilise pour cela une fonction de hachage.

1. Reconnaître les fichiers corrompus

Des fichiers peuvent être endommagés pour diverses raisons, par exemple à cause de supports d'enregistrement défectueux, d'erreurs dans la transmission des données, d'erreurs d'écriture pendant la copie, d'erreurs logicielles ou de virus. Le train de bits original ***i*** du fichier est modifié ou tronqué. On parle alors de fichiers corrompus.

Il y a plusieurs manières de reconnaître les fichiers corrompus:

La manière la plus simple est une comparaison bit par bit avec le fichier original non modifié. Cela ne permet toutefois pas de déterminer dans l'absolu lequel des fichiers est le fichier non modifié, ni si éventuellement les deux fichiers sont corrompus de manières différentes.

Un deuxième procédé consiste à calculer et à enregistrer une simple somme de contrôle ***ii*** à partir du train de bits. La somme de contrôle est conçue pour que sa valeur soit modifiée par les erreurs fortuites du train de bits. Cela permet avant tout de discerner sans doute possible des erreurs intervenues dans la transmission et dans l'enregistrement des données, mais non les modifications intentionnelles des données.

Si l'on utilise au lieu d'une somme de contrôle une valeur de hachage ***iii***, les modifications intentionnelles des données pourront elles aussi être discernées. Une valeur de hachage est une caractérisation quasiment univoque d'un flux de données, sous la forme d'une brève chaîne de caractères. Si une fonction de hachage cryptographique est utilisée ***iv*** , étant donné qu'il existe une correspondance univoque entre un train de bits et sa valeur de hachage, on peut être certain que si cette valeur de hachage est identique pour deux trains de bits, ils sont eux-mêmes identiques.

En conservant une valeur de hachage (ou une somme de contrôle) correspondante, on peut donc vérifier l'intégrité d'un fichier, c'est-à-dire l'absence de changements dans le train de bits. L'authenticité d'un fichier, par contre, c'est-à-dire la garantie qu'il a été produit à un certain moment par une certaine personne ou par un certain office, ne peut être garantie qu'au moyen d'une signature numérique ***v***.

2. Problématique pour les archives

L'état actuel de la technologie des matériels et des logiciels garantit dans une large mesure que les erreurs de transmission ou les défectuosités des supports d'enregistrement ne passent pas inaperçues. Une plateforme d'enregistrement suffisamment sûre ***vi*** garantit en outre généralement que les fichiers, une fois enregistrés, soient régulièrement vérifiés. Pour cela, on crée, pour chaque secteur du fichier, au niveau

i Terme désignant une séquence de bits qui représentent une information *(angl. bitstream)*.

ii Somme de contrôle *(angl. checksum)*. Une somme de contrôle sert principalement à détecter des erreurs de bits dans le train de bits. Le contrôle de redondance cyclique *(angl. CRC32)* est le plus connu des algorithmes de somme de contrôle.
Voir aussi: http://de.wikipedia.org/wiki/Checksum

iii Valeur de hachage *(angl. hash-code, hash-value)*. Une valeur de hachage est une reproduction aussi univoque que possible d'un grand flux de données sur une chaîne de caractères relativement courte. Une fonction bien connue pour générer des valeurs de hachage est la fonction MD5.
Voir aussi: http://de.wikipedia.org/wiki/Hashwert

iv Une fonction connue pour générer des valeurs de hachage cryptographique est le SHA-1 *(angl. secure hash algorithm)*.
Voir aussi: http://de.wikipedia.org/wiki/SHA-1

v Les signatures numériques sont basées sur des systèmes de cryptage asymétrique et utilisent une paire de clés, composée d'une clé privée (secrète) et d'une clé publique (non secrète).
Voir: http://de.wikipedia.org/wiki/Digitale_Signatur
ou: Niels Fromm, *Signatur und Zeitstempel zur Wahrung von Authentizität und Integrität*, 2009
http://edoc.hu-berlin.de/cmsj/32/fromm-niels-63/PDF/fromm.pdf

vi Voir à ce sujet le "Colloque de stockage" du CECO:
http://www.kost-ceco.ch/cms/index.php?id=95,82,0,0,1,0

interne, des sommes de contrôle que l'on recalcule régulièrement.

Cela signifie que dans l'exploitation actuelle d'archives électroniques, les phases critiques sont surtout le passage des fichiers de la source/provenance jusqu'à la plateforme d'enregistrement et le processus de copie aux archives. En copiant de grands fichiers ou de grandes collections de fichiers, on court par exemple le rique que le processus de copie s'arrête ou soit interrompu inopinément, que le support cible soit trop petit, etc.

Naturellement, le problème ne se pose pas lorsque les fichiers traités par les archives sont des conteneurs SIP ou AIP dotés en eux-mêmes d'une empreinte numérique (valeur de hachage). L'intégrité du conteneur peut alors à tout moment être vérifiée.

Si par contre les archives travaillent avec des fichiers individuels ou ne disposent pas d'une plateforme de stockage suffisamment sûre, il est également possible et nécessaire de garantir l'intégrité des données par des moyens simples.

3. Simple File Verification

La *Simple File Verification* (SFV) [vii] est une procédure pour vérifier l'intégrité des fichiers au moyen de sommes de contrôle CRC32. L'extension habituelle des fichiers SFV est .sfv. Ces fichiers contiennent des noms de fichiers et des sommes de contrôle correspondantes. Un programme spécial permet de créer des fichiers SFV et d'y aligner les sommes de contrôle qui y sont enregistrées.

Un fichier SFV pour les fichiers file1.dat à file4.dat se présente ainsi:

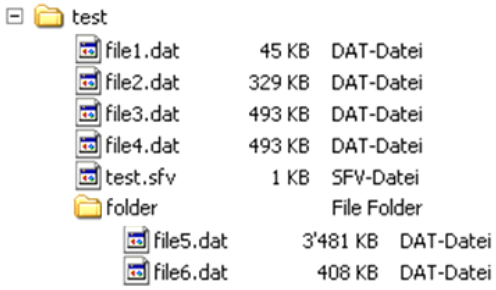
```
file1.dat  71f6aba0
file2.dat  96e67f7d
file3.dat  67f151f7
file4.dat  3094bcc6
```

La plupart des programmes de validation SFV peuvent générer et vérifier non seulement des sommes de contrôle CRC32, mais encore des valeurs de hachage MD5 et/ou SHA (extensions **.md5** et **.sha**).

La syntaxe d'un fichier MD5 se présente un peu différemment:

```
cb2ab3697c21950f0d5285f9a0d5ab91 *file1.dat
83cf7aaf9ee7088db45133b54bfd4dfc *file2.dat
dfe90465485b22cae193df23460ef766 *file3.dat
0cc107a0e37b541edfd13f32503d16e6 *file4.dat
```

Des structures entières de répertoire peuvent en outre être reproduites dans un fichier SFV, par exemple la structure de fichiers suivante:




```
file1.dat 67f151f7
file2.dat 3094bcc6
file3.dat 96e67f7d
file4.dat 17714f69
folder\file5.dat 71f6aba0
folder\file6.dat 68cb2916
```

Le fichier SFV portant le nom du dossier sélectionné (**test.sfv)** est créé dans le dossier correspondant.

4. Programmes recommandés

Les formats des fichiers de sommes de contrôle sont définis arbitrairement (SFV, MD5, SHA), de sorte que le choix du programme utilisé ne joue en général aucun rôle [viii]. Il est donc aussi possible de générer les fichiers de sommes de contrôle avec un programme et de faire la validation avec un autre programme. Avec certains programmes, un dossier entier est sélectionné pour la vérification (les fichiers de sommes de contrôle reçoivent alors le nom du dossier); pour d'autres programmes, les fichiers doivent être sélectionnés, et un nom doit être attribué aux fichiers de sommes de contrôle.

Le **hkSFV** est un favori, mais n'est malheureusement plus supporté. L'entreprise big-O Software n'existe apparemment plus. hkSFV peut créer et valider des fichiers de vérification SFV et MD5.

http://www.kost-ceco.ch/cms/download.php
http://www.big-o-software.com/

Le programme **QuickSfV** est très pratique parce qu'il s'intègre dans le menu contextuel (bouton droit de la souris). Mais il nécessite des droits d'administrateur pour être installé.

http://www.quicksfv.org/

L'utilisation de **xyChecksums** est un peu compliquée, mais ce programme est open source sous GPL et peut traiter des collec-

[vii] Voir: http://de.wikipedia.org/wiki/Simple_File_Verification

[viii] Une liste (incomplète) de logiciels de validation SFV sur Wikipedia: http://en.wikipedia.org/wiki/Comparison_of_file_verification_software

tions de fichiers vraiment volumineuses (> 500 Go).

http://wxchecksums.sourceforge.net/

Des collections également volumineuses à souhait peuvent être traitées par le *Command Line Tool* **md5sum**. Ce programme est disponible pratiquement sur tous les systèmes d'exploitation.

http://gnuwin32.sourceforge.net/packages/coreutils.htm
http://www.gnu.org/software/coreutils/
http://www.kost-ceco.ch/cms/download.php

Pour ne rien oublier, mentionnons encore **WIN-SFV32**, la forme d'origine de tous ces logiciels de validation.

http://www.kost-ceco.ch/cms/download.php

5. Liens utiles

- File verification
  http://en.wikipedia.org/wiki/File_verification
- Simple file verification (SFV)
  http://en.wikipedia.org/wiki/Simple_file_verification
- Somme de contrôle
  http://en.wikipedia.org/wiki/Checksum
- Fonction de hachage cryptographique
  http://en.wikipedia.org/wiki/Cryptographic_hash_function
- Comparaison de logiciels de vérification de fichiers
  http://en.wikipedia.org/wiki/Comparison_of_file_verification_software
- *Kryptographie "Hashfunktionen - Defintion und Eigenschaften"*, travail d'études de la Haute école de Würzburg
  http://www.fh-wuerzburg.de/fh/fb/all/personal/interper/WSCHNELL/Hash.pdf
- Niels Fromm, *Signatur und Zeitstempel zur Wahrung von Authentizität und Integrität*, 2009
  http://edoc.hu-berlin.de/cmsj/32/fromm-niels-63/PDF/fromm.pdf