Simple File Verification: Integrität oder Authentizität von Dateien überprüfen

"File Verification" ist ein Verfahren zur Überprüfung der Integrität oder Authentizität von Dateien mit Hilfe eines entsprechenden Algorithmus. Bei der "simple file verification" (SFV) wird dafür eine Hash-Funktion verwendet.

1. Korrupte Dateien erkennen

Dateien können durch eine Vielzahl von Ursachen beschädigt werden, etwa durch defekte Speichermedien, Fehler bei der Datenübertragung, Schreibfehler beim Kopieren, Softwarefehler oder Viren. Dabei wird der ursprüngliche Bitstrom ***i*** der Datei verändert oder verkürzt. Man spricht in diesem Fall von korrupten Dateien.

Korrupte Dateien können auf mehrere Arten erkannt werden:

Die einfachste Art ist der bitweise Vergleich mit der unveränderten Originaldatei. Damit kann allerdings nicht abschliessend geklärt werden, welche der Dateien die unveränderte ist, oder ob allenfalls beide Dateien auf unterschiedliche Art korrupt sind.

Bei einem zweiten Verfahren wird eine einfache Prüfsumme ***ii*** über den Bitstrom berechnet und gespeichert. Die Prüfsumme ist so gestaltet, dass zufällige Fehler im Bitstrom den Wert der Prüfsumme verändern. Damit können vor allem Fehler bei der Datenübertragung und der Datenspeicherung sicher erkannt werden, nicht aber absichtliche Datenänderungen.

Wird statt einer Prüfsumme ein Hashwert ***iii*** verwendet, können auch absichtliche Datenänderungen erkannt werden. Ein Hashwert ist eine nahezu eindeutige Kennzeichnung eines Datenstromes in Form einer kurzen Zeichenkette. Wird eine kryptographische Hashfunktion ***iv*** eingesetzt, ist eine eindeutige Zuordnung Bitstrom → Hashwert garantiert.

Durch das Aufbewahren eines entsprechenden Hashwertes (bzw. einer Prüfsumme) kann also die Integrität einer Datei, das heisst das Fehlen von Veränderungen im Bitstrom, überprüft werden. Die Authentizität einer Datei hingegen, das heisst die Garantie, dass eine Datei zu einem bestimmten Zeitpunkt von einer bestimmten Person oder Stelle erzeugt worden ist, kann nur durch eine digitale Signatur ***v*** gewährleistet werden.

2. Problemstellung im Archiv

Der heutige Stand der Hard- und Softwaretechnik garantiert weitgehend, dass keine Übertragungsfehler oder Speichermedienfehler unerkannt stattfinden können. Eine hinreichend sichere Speicherplattform ***vi*** garantiert in der Regel auch, dass die einmal gespeicherten Dateien regelmässig überprüft werden. Dazu werden intern Prüfsummen für jeden Dateiblock angelegt und regelmässig wieder neu berechnet.

Das bedeutet, dass im heutigen Betrieb eines Digitalen Archivs vor allem der Weg der Dateien von der Quelle/Provenienz zur Speicherplattform und Kopiervorgänge im Archiv kritisch sind. Mögliche Risiken beim Kopieren von grossen Dateien oder Dateisammlungen sind, dass der Kopiervorgang

i Bezeichnung für eine Folge von Bits, die eine Information repräsentieren *(engl. bitstream)*.

ii Prüfsumme *(engl. checksum)*. Eine Prüfsumme dient vornehmlich dazu Bitfehler im Datenstrom zu erkennen. Die zyklische Redundanzprüfung *(engl. CRC32)* ist der bekannteste Prüfsummenalgorithmus.
Siehe auch: http://de.wikipedia.org/wiki/Checksum

iii Hashwert *(engl. hash-code, hash-value)*. Ein Hashwert ist eine möglichst eindeutige Abbildung eines grossen Datenstromes auf einen relativ kurze Zeichenkette. Eine bekannte Funktion zum Erzeugen von Hashwerten ist die MD5-Funktion.
Siehe auch: http://de.wikipedia.org/wiki/Hashwert

iv Eine bekannte Funktion zum Erzeugen von kryptographischen Hashwerten *(engl. secure hash algorithm)* ist SHA-1.
Siehe auch: http://de.wikipedia.org/wiki/SHA-1

v Digitale Signaturen basieren auf asymmetrischen Kryptosystemen und verwenden ein Schlüsselpaar, das aus einem privaten (geheimen) und einem öffentlichen (nicht geheimen) Schlüssel besteht.
Siehe dazu: http://de.wikipedia.org/wiki/Digitale_Signatur
oder: 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 Siehe dazu das KOST-Kolloquium "Archivtaugliche Speicherinfrastruktur":
http://www.kost-ceco.ch/cms/index.php?id=95,82,0,0,1,0

unbeabsichtigt abbricht oder unterbrochen wird, dass das Zielmedium zu klein ist, etc.

Das Problem stellt sich natürlich nicht, wenn die vom Archiv gehandhabten Dateien SIP- oder AIP-Container sind, die selber mit einem entsprechenden Container-Hashwert versehen sind. Hier kann die Integrität des Containers jederzeit überprüft werden.

Wird im Archiv hingegen mit einzelnen Dateien gearbeitet, oder steht keine hinreichend sichere Speicherplattform zur Verfügung, so soll und kann auch mit einfachen Mitteln die Datenintegrität garantiert werden.

3. Simple File Verification

Simple File Verification (SFV) [vii] ist ein Verfahren zur Überprüfung der Integrität von Dateien mit CRC32-Prüfsummen. Die übliche Dateiendung für SFV-Dateien ist **.sfv**. SFV-Dateien enthalten Namen und zugehörige Prüfsummen anderer Dateien. Mit einem entsprechenden Programm können SFV-Dateien erstellt und die darin gespeicherten Prüfsummen abgeglichen werden.

Eine SFV-Datei für die Dateien file1.dat bis file4.dat sieht folgendermassen aus:

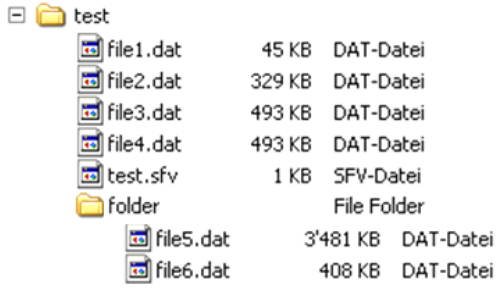
```
file1.dat  71f6aba0
file2.dat  96e67f7d
file3.dat  67f151f7
file4.dat  3094bcc6
```

Die meisten SFV-Validator-Programme können neben CRC32-Prüfsummen auch MD5- und/oder SHA-Hashwerte generieren und prüfen (mit den Endungen **.md5** und **.sha**).

Die Syntax einer MD5 Datei sieht etwas anders aus:

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

Es können auch ganze Verzeichnisstrukturen in einer SFV-Datei abgebildet werden, wie zum Beispiel die folgende Dateistruktur:




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

Die SFV-Datei mit dem Namen des ausgewählten Folders (**test.sfv)** wird im entsprechenden Folder angelegt.

4. Empfehlenswerte Programme

Die Dateiformate für die Prüfsummendateien sind vorgegeben (SFV, MD5, SHA), so dass die Wahl des verwendeten Programms in der Regel keine Rolle spielt [viii]. Die Prüfsummendateien kann so auch mit dem einen Programm erzeugt und mit einem anderen Programm validiert werden. Bei gewissen Programmen wird ein ganzer Ordner zur Prüfung gewählt (die Prüfsummendateien erhält dann den Ordnernamen), bei anderen Programmen müssen die Dateien ausgewählt und für die Prüfsummendateien ein Name vergeben werden.

Favorit ist **hkSFV**, leider wird es nicht mehr unterstützt. Offenbar gibt es die Firma *big-O Software* nicht mehr. hkSFV kann SFV- und MD5-Prüfdateien anlegen und validieren.

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

Recht einfach zu bedienen ist **QuickSfV**, weil es sich in das Kontextmenü (rechte Maustaste) integriert. Es ist aber nur mit Admin-Rechten installierbar.

http://www.quicksfv.org/

Etwas umständlich in der Bedienung ist **xyChecksums**, aber das Programm ist *open source* unter GPL und kann auch wirklich grosse Dateisammlungen verarbeiten (> 500 GB).

http://wxchecksums.sourceforge.net/

Ebenfalls beliebig grosse Dateisammlungen können mit dem *Command Line Tool* **md5sum** verarbeitet werden. Das Programm ist praktisch auf allen Betriebssystemen verfügbar.

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

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

[viii] Eine nicht vollständige Liste von SFV-Validator Programmen bei Wikipedia :
http://en.wikipedia.org/wiki/Comparison_of_file_verification_software

Der Vollständigkeit halber sei auch **WIN-SFV32** erwähnt, die Urform aller dieser Validator-Programme.
http://www.kost-ceco.ch/cms/download.php

5. Weiterführende Links

- File verification
  http://en.wikipedia.org/wiki/File_verification
- Simple file verification (SFV)
  http://en.wikipedia.org/wiki/Simple_file_verification
- Checksum
  http://en.wikipedia.org/wiki/Checksum
- Cryptographic hash function
  http://en.wikipedia.org/wiki/Cryptographic_hash_function
- Comparison of file verification software
  http://en.wikipedia.org/wiki/Comparison_of_file_verification_software
- *Kryptographie "Hashfunktionen - Defintion und Eigenschaften"*, Studienarbeit der FH 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