

[Peter Jockisch](#)

Practical Application of Cryptographic Checksums
Checksum Formation with Free Software under MS-Windows, Unix/BSD,
GNU/Linux and MacOS X, with a Step-by-Step Introduction
to Signing and Encrypting E-mails and Files with OpenPGP
(One-page short version, [PDF file US Letter](#), [PDF file A4](#))

PETER JOCKISCH, Freiburg i. Br.
peterjockisch.de

1.1 Introduction

Computer files can be manipulated in many ways unnoticed. Cryptographic checksums, hash values, serve to protect your data: By forming an electronic fingerprint of a file, an always constant numerical value is created. If this value deviates at a later point in time, there is damage or manipulation. With a single mouse click, the integrity of a file can be checked at any time.

Cryptographic checksums form the basis for cryptographic signing and encryption, for website- and e-mail certificates, for the qualified electronic signature, and for the technical understanding of revision-proof e-mail archiving, to which all merchants are legally obliged.

This introduction presents two [free](#) graphical programs for checksum generation, *CyoHash* and *Jacksum*, for file manager operation.

Console-based programs are also described, they are available across operating system platforms and are pre-installed on MS-Windows 10 as well as most Unix/BSD and GNU/Linux systems ([see 2.3 for instructions](#)). This means that no programs need to be installed at all, the existing operating system resources are sufficient to calculate checksums.

1.2 Functional Principle

1.2.1 Electronic Fingerprints

Humans are complex creatures. In order to identify them quickly and easily, fingerprints are often created. Computer files can be identified according to the same principle: by generating an “electronic fingerprint”, the so-called cryptographic checksum, an always constant number.

By means of standardized procedures, a fast integrity and authenticity check of files of any kind can be carried out.

Human fingerprints are created with stamp pads, electronic fingerprints with a checksum program.

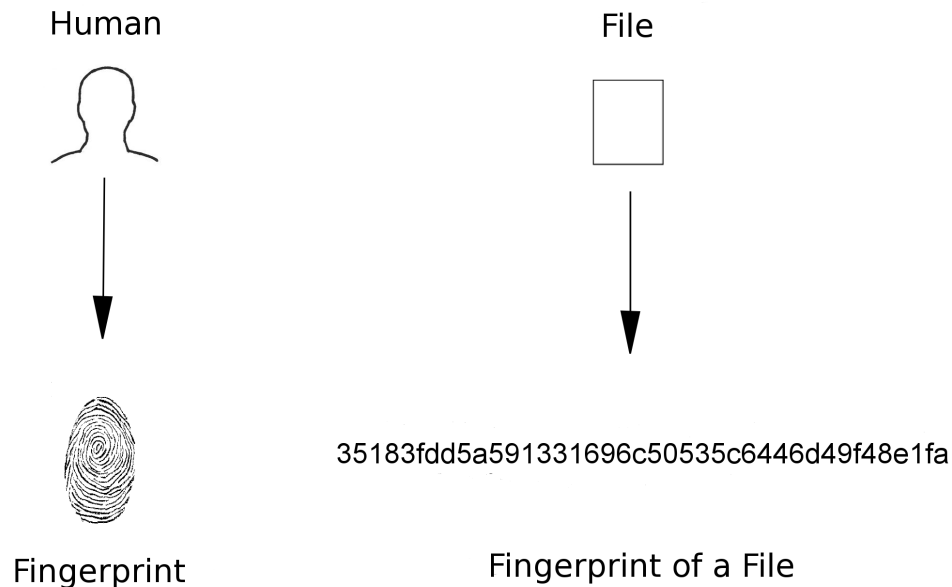


Fig.1: Proof of authenticity for human and computer files

1.2.2 Quality Criteria

We're looking at *cryptographic* checksums. They are based on [hash functions](#), which provide hash values as a result for any file. This value is also called hash code.

A file, as well as identical copies of it, always has the same hash checksum. However, if only a single bit or character changes due to damage or manipulation, a completely different hash code should be created.

A hash-function-checksum-procedure should therefore always return different values for different computer files. Depending on the method used, the calculated checksum always has the same length. Therefore, of course, only a limited number of numbers can be displayed: There are practically an infinite number of computer files, so that it is impossible to assign a different value to each of these files with a number of fixed length.

From a security point of view, there are various attack scenarios, including forgery of documents. An attacker

would like to create a fake version of a given original file, for example a business order, with a manipulated, increased order quantity that has the same hash value checksum. After making the changes to the document, he then tries to obtain a file version with a cryptographic checksum identical to that of the original file by trial and error, perhaps by inserting invisible control characters. Such an attack, of course, uses supporting computer programs.

If an attacker actually succeeds in creating a second file (*at a reasonable cost in terms of time*) containing the desired manipulations and which has the same cryptographic checksum of the original file, the hash function procedure in question is "broken". Once such a weakness becomes known, it should no longer be used. Due to continuous [research work](#), weaknesses are detected a long time in advance.

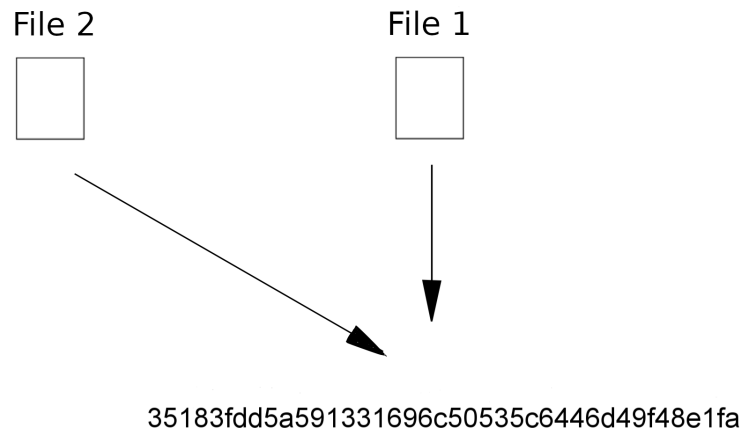


Fig.2: Checksum collision

If there were a computer with infinite computational power, theoretically any method could be broken by simply trying out all the possibilities ([brute force attack](#)). In practice, such an approach is not considered [practicable](#) in the majority of cases, since the necessary calculations are almost never feasible in a reasonable time.

Most hash functions have had a limited lifetime and have been replaced by successor functions for security

reasons. Computer generations with higher computing power contribute to shortening the service life. In addition to computational force-based attacks, however, there are also attacks with a different orientation, and it can never be ruled out that mathematical creativity can be used to launch practicable attacks today.

In the background a huge army of mathematicians works and researches, especially for intelligence services. Not all scientific findings are published.

1.2.3 Prevailing Standards in the West and Russia

Until 2016, the Western IT infrastructure was predominantly based on the [SHA-1 algorithm](#) (Secure Hash Algorithm 1). Since 2017, this algorithm has been regarded [as finally broken](#), and the computing time required to corrupt it has fallen drastically. Experts now recommend the SHA-2 variants SHA256, SHA384, or SHA512.

The recommended successor algorithm to SHA-2, [SHA-3 \[1\]](#), has been [officially established](#) since 2012.

In Russia and many other CIS states, [GOST R 34.11-94](#) resp. GOST 34.311-95 were the previous hash standards in authorities and various economic sectors. [\[2\]](#) As with SHA-1, [structural weaknesses](#) were also found in this standard.

1.3 Do Technologies exist that are blocked to the Public?

1.3.1 Obsolete Computer Systems

All computational power-related statements in this introduction refer to publicly available computer systems and research work released to the general public. The use of the latest, most advanced computer technology is probably currently still reserved for intelligence services in order to guarantee them a computing power advantage for an effective leveraging of established encryption technology.

The widely approved encryption procedures may not be readily breakable for lower levels of government. However, at the top of the hierarchy, at the intelligence level, there should be unrestricted access to the latest computer technology. In addition, all data transferred via the Internet will probably be archived for

automatic evaluation. Under this aspect, the resilience of files sent over the Internet that have been encrypted using publicly standardized technology is put into perspective.

For a long time there have been considerations that certain cryptographic algorithms raised to official standards

might have inherent mathematical weaknesses which are only known to the experts of the intelligence services. A possibly existing influence of the secret services on the design of security products (software and possibly hardware backdoor problems, open questions about standards, etc.) is the subject of numerous articles on computer security, for example in [“Did NSA Put a Secret Backdoor in New Encryption Standard?”](#). Several renowned companies have already directly or indirectly confirmed that they cooperate with intelligence services in their product development. One of the official reasons for this was the intention to optimize the technical safety of company products. It remains to be seen how much pressure was exerted on “cooperation”.

Corrupted electronics, known or unknown “advanced” hardware architectures with factory-built “remote maintenance” functions, [possibly even with a wireless system built into the processor](#), represent the other side of the problem.

1.4 Application examples: Business world, Internet, Archiving

The comprehensive requirements for the legally recognized personal authentication, which is almost always linked to the use of [proprietary](#) software and hardware, will not be discussed here. Section [1.5](#) contains further sources of information on the so-called qualified signature.

The creation of cryptographic [checksums](#) not only provides clues to the authenticity and integrity of files. It also enables acknowledgements of receipt and written confirmation of the last processing status of a file, as demonstrated by the following examples.

1.4.1 Preservation of File Integrity

You create a business balance sheet and then go on holiday. For quality control purposes, make a note of the cryptographic checksum of the completed balance sheet file before departure. After the vacation, you create the checksum again and verify whether the file is intact or whether it has been damaged or manipulated.

Unauthorized access, e.g. to an accounting file, can be detected in this way. In such cases, notify the system administrators and insist on a recovery of the original file version, which will of course only be successful if the file has the noted checksum again.

The file date specification and [version control systems](#) are no substitute for cryptographic checksums, since both can be manipulated directly or indirectly. Specialized programs can change both the creation and modification dates of files, even directory-wide.

Consistency checks using checksums usually work faster, more effectively and more securely. A separate checksum program should therefore always be available.

1.4.2 Indicator for the Processing Status of a File

When leaving a company you would like to have the last processing status of a computer file confirmed in writing, example text: *“Confirmation – This confirms that the file [file name] maintained by [first name, last name] has the SHA-512 checksum [concrete checksum] in its last processing status on [date]”*.

Two or three different cryptographic checksum methods are recommended for this purpose. This may be more secure in the long run, one of the methods may last longer in the future. In this way, the company secret is protected and you can still secure yourself to a certain extent. Should there ever be any queries, you will have a written indication of your latest processing status.

1.4.3 Armament against White-Collar Crime, Protection against Mobbing

In the context of general quality control and whenever corruption, lies, intrigues, bullying, sabotage and white-collar crime are likely, the use of cryptographic checksums, even before presentation dates (file checks), is recommended for your own protection. Signing software is often ruled out because it is naturally coupled with encryption functions and is therefore not tolerated on every

workstation computer. Company secrets could get out encrypted or malware could get in undetected. However, a [free](#) checksum program – not to be confused with freeware [3] – can be installed responsibly on company computers if the [system administrator](#) deems this appropriate.

1.4.4 File Reference in Contracts and Confirmations of Receipt

1.4.4.1 Contracts and Acknowledgements of Receipt

For written contracts and acknowledgements of receipt, electronic fingerprints facilitate the reference to computer files. Files of any kind can be uniquely identified by their hash value, e.g. text documents, video films, *recordings of conversations and interviews* (sound files in general), programs, CAD files. Services that have been rendered, which are finally available in the form of a data carrier, e.g. a CD-ROM or DVD to be delivered, can also be confirmed in writing in this way.

Electronic fingerprints can also be created from [archive files](#) [4]. Acknowledgements of receipt form a broad field of application.

1.4.4.2 Image Licensing

In image-forming occupational fields, photos go through many post-processing phases until they are published, and it is usually impractical to specify specific final files in advance. However, reference files can be binding for the publication or long-term exploitation of basic artistic/image design specifications: Selection of the image detail, basic contrast characteristics, implicit prohibition of (further) “cosmetic retouching”, etc. Image descriptions and miniature images of the originals are then included in the exploitation contract together with the corresponding file checksums. If possible, only use file formats that either work completely without compression or with lossless compression, formats that can also store color profile information. Of course, reference to reference color spaces requires proper color management and comprehensive preparation of the equipment pool.

Sometimes bindingly finished image files are possible. Provided there are enough resolution reserves for resizing (scaling), photos for a website can be exactly defined in all parameters. If a suitable standard is chosen (e.g. TIFF, JPEG, PNG, GIF), there is a high probability that the images will still be displayed by future webpage reading programs in the long term.

1.4.5 Indicator of the Authenticity of Documents, transmitted by Telephone

The recipient of a file or a data carrier sent by letter post can be informed by telephone of the hash function checksum for checking purposes. Although falsifying a voice has become much easier since the introduction of the “[Adobe Voco](#)” [5] software, it is still a time-consuming process. However, personal signing would be more convenient and secure.

1.4.6 Publication of Hash Values as alternative Proof of Authenticity

Some states only allow limited encryption and signing (personal electronic proof of authenticity). To a certain extent, cryptographic checksums can be used as an alternative stopgap:

1. First create the message or document separately as a computer file (text or PDF file, image, video, etc.). Attach the file to an e-mail and send the message.
2. Publish the hash codes of the sent documents on a webpage similar to a diary. Several reputable [free](#) (advertising-financed) [webhosters](#) are available for this purpose ([basic article on web hosting](#)). Even without (X)HTML knowledge you can create websites, e.g. with [free HTML editors](#).

Alternatively, free blog systems that do not require any technical design knowledge are recommended. A possible free service is [Blogger.com](#), another [Wordpress.com](#). Both are easy and uncomplicated to

use and can be used immediately, comment functions can be deactivated. When making your selection, make sure that the password you choose is SSL/TLS-secured during the login process; your user password should always be transported over the Internet via https.

Finally, create schematic entries, for example in the form *checksum procedure - file checksum*. There is no need for more information. For large daily entries, you could optionally add a file name abbreviation. The e-mail attachment “request.pdf” would then become “r...t.pdf” or simply “r...t”.

3. The recipient of the file can now perform a hash value comparison by calling up your webpage or blog and reading the corresponding hash code for the symbolic message.

1.4.7 Publish Documents with Hash Values

For the publication of documents on the Internet or in the Intranet (company network), it is advisable to specify hash values, possibly on a subpage, in the so-called download area. The use of a legally recognized SSL/TLS [certificate](#) [6] for the encrypted transmission of the websites with the published checksums increases secu-

urity. By comparing the hash values, users can be relatively sure that documents downloaded from reputable sources are free of malicious code (viruses, etc.), manipulation and transfer damage. Certificates issued by external instances, however, are possibly associated with [residual risks](#).

1.4.8 Archiving Files

When archiving files on CD-ROMs and DVDs, it is recommended that you write down the disk hash code. To check this, periodically compare the actual value with the originally noted hash value. This allows early damage to be detected. Unfortunately, regular copying

to new archiving specialist data carriers at relatively short intervals is still unavoidable at the moment.

Basic information on this topic can be found in the [FAQ section](#) of [Jacksum.net](#).

1.4.9 Increased Security for Password Storage

There are numerous other possible applications in information technology and electrical engineering, such as a variant for the safety-enhanced storage of user account data: A plain text password can also be stored exclusively in the form of its associated hash value. If the user enters his plaintext password again, the hash value is generated again and compared with the stored one. In the event of a data intrusion or data theft, no plaintext passwords are lost for the time being.

Good contemporary hash functions work like [one-way functions](#). They assign an individual hash value to a file. The reverse way, however, the calculation of the original file from the hash value, is not possible in a practicable time - according to the current publicly known state of knowledge, in which the publicly available procedures and technologies are used as a standard.

1.4.10 Legally recognized, audit-proof E-mail Archiving

In some industries, e-mails that lead to business transactions/orders are legally regarded as commercial letters. For their archiving, a simple saving or printing is no longer sufficient; instead, revision-proof archiving must be carried out in a technical manner that excludes subsequent, undetectable manipulation of the e-mail data.

The technical implementation of most programs probably takes place with the help of cryptographic checksums. Electronic fingerprints of all incoming and outgoing e-mails are created and stored in encrypted form. If, for example, an auditor wishes to view a particular

e-mail, the stored e-mail file is loaded into the corresponding e-mail archiving program, e.g. from a CD ROM. When reading in, the cryptographic checksum of the e-mail file is formed again and compared with the originally archived associated checksum for correspondence.

Only very specific software and/or hardware solutions from certain manufacturers are legally recognized within the framework of the legal requirements. There are numerous high-quality introductory articles on this subject available in the Internet, written by specialized lawyers and IT experts.

1.5 Cryptographic Signature, Website and E-mail Certificates

1.5.1 Cryptographic Signature and E-mail Certificates

In the past, people or offices provided their documents with an additional proof of authenticity by applying complex patterns to the documents with sealing lacquer or wax and [sealing stamps](#). Today, cryptographic keys or certificates assume the function of the seal stamp: For a document, e.g. an e-mail file, an accompanying, personal proof of authenticity, the so-called cryptographic signature, is calculated

with the help of a certificate. Upon receipt of the message, the recipient's e-mail program automatically determines (using this signature, among other things) whether the document was actually created by the specified sender (certificate holder/card holder).

E-mail and website certificates therefore have an identification function with which correspondence partners and webpages can prove their identity.

Classical identity cards are issued by state authorities. E-mail and website certificates are issued by certification authorities (CAs). And this is where the two decisive differences lie: classic ID cards are all equal and officially recognized, there is only one single issuer who also functions as a certification institution: the state.

E-mail and website certificates, on the other hand, exist in different quality classes, with different meaningfulness. Only certificates of the highest quality [issued by officially recognized certification bodies](#) are legally recognised, e.g. [e-mail certificates](#) for which an identity card or passport check is carried out to ensure that the personal details in the certificate match those in the identity card, or [EV website certificates](#).

The encryption standard [OpenPGP](#) works both with and without certification authorities, which means that you can also use self-created key certificates for signing and encryption:

1.5.1.2 Instructions: Signing and encrypting Files and E-mails with OpenPGP

The encryption standard [OpenPGP](#) enables signed e-mail communication both with and without certification authorities (CAs). For [signed](#) (and encrypted) communication, you create a (secret) [key](#) (certificate), the corresponding [public certificate](#) part of which you can send to your communication partners and publish on the Internet.

When signing documents or e-mails, you then enter your passphrase, whereupon a (companion) signature is created for your e-mail or file; the passphrase can optionally be stored temporarily. The [e-mail client](#) of the person you are corresponding with then verifies fully automatically in the background (with the help of your public certificate part) whether the attached signature was created with your private key (certificate).

If you want to send an encrypted document, you need the public OpenPGP certificate part of the file or e-mail recipient, also in order to check his/her attached e-mail signatures for authenticity.

Software Installation: An encryption software and the [command line](#) would be sufficient in itself, but most users prefer graphical interfaces. So you need two or three programs:

1. A free and open source encryption software: [Gnu Privacy Guard](#), [GnuPG \(WP article\)](#). GnuPG is [free](#) and [cross-platform](#) software and available 100 % free-of-charge, for Unix/BSD, GNU/Linux, MS-Windows and MacOS. The MS-Windows version is called [Gpg4win](#) (no donation required for download), the GPG4win suite already contains the [Kleopatra](#) graphical interface and the GpgEX program extension for Microsoft (file manager) [Windows File Explorer](#), for file encryption and signing with a right mouse click.
2. One of the dozens of free [graphical interfaces](#), for example the already mentioned [Kleopatra](#) for MS-Windows and Unix/BSD or GNU/Linux, to create keys (certificates) via graphical menu and optionally sign and encrypt files.
3. For the fully automatic operation with e-mail programs, you use one of the numerous free and charge-free [interfaces](#), for example [Enigmail](#) for Mozilla-[Thunderbird](#), or for [MS-Outlook](#) the [GpgOL](#) already included in the GPG4win suite.

Now generate your secret key, you can generate as many test keys as you like:



1. Open Kleopatra and create your key (note down a passphrase beforehand): “File” → “New Certificate” → “Create a personal OpenPGP key pair”. Then make a backup copy on USB stick and/or CD-ROM/DVD. To do this, use Kleopatra's export function (select key first): “File” → “Export Secret Keys”.

To export the secret key, you will be asked to enter your secret passphrase. You can open or drag the key in a file viewer or in a text editor. At the beginning or end of the file are respectively “-----BEGIN PGP PRIVATE KEY BLOCK-----” resp. “-----END PGP PRIVATE KEY BLOCK-----” (delete any cached or copied secret key copy thoroughly!)

2. You export your public key part via “File” → “Export Certificates” ([Menu Reference](#)). Check immediately that it is the public key part, save it on the desktop, for example, and then drag the file to a text editor or browser, for example Mozilla Firefox. At the end or at the beginning is “-----BEGIN PGP PUBLIC KEY BLOCK-----” respectively “-----END PGP PUBLIC KEY BLOCK-----”.

3. *Use in the e-mail software:* After installing the relevant e-mail reader interface, open your e-mail client. Under Thunderbird with the extension (AddOn) “[Enigmail](#)” go to “Local Folders”, select your e-mail account and then go to “OpenPGP Security”. There, place a check mark in “Enable OpenPGP support (Enigmail) for this identity” and select your previously created key under “Select key”.

From now on you can sign your e-mails by default. In order to verify the signatures of incoming e-mails (or in order to encrypt for third parties), you must have loaded the public key part of your respective correspondence partner into your program once, under Kleopatra via “File” → “Import”, or under “Enigmail” → “Key Management” → “Import keys [...]”, or via one of Enigmail's fully automatic functions which appear automatically.

Each key has a fingerprint, which can also be transmitted verbally on the telephone, for example. In order to determine it, select the key under Kleopatra, open the menu with a right click and select “Details”. Keys can also be authenticated by third parties. Read: [The Kleopatra Handbook](#).

Get familiar with the features of Enigmail, first of all write a signed test email addressed to yourself. Read further instructions and the [Enigmail User Manual](#).

Sign, encrypt and decrypt files: You can create signatures for your files, or encrypt them with your own key (or with public keys of third parties). Try it with any document, such as a text file. Under MS-Windows: Go to the document and press the right mouse button, the drop-down menu opens, select the GpgEX options (symbol with open lock) and choose the “Sign” function, whereupon the signature is created after entering your passphrase. Then go to this accompanying signature file and select the corresponding GpgEX option again, which allows you to carry out an automated check. Try out the numerous functions of the GpgEX options menu.

Password encryption: You can also encrypt a file with only a password, so that afterwards all those who know the password can open the file.

Optionally, you can also simply drag files into the open Kleopatra window, whereupon an action menu appears, or you can select a corresponding action in the Kleopatra file menu, whereupon the file manager opens.

If you want to encrypt a file for a specific recipient, you must select his public key. Create a test folder and try out all functions.

1.5.2 Website Certificates Industry in Criticism

Webpages can be called up in encrypted form ([https](#)) so that third parties under the law cannot read the contents of the pages called up and any correspondence that may have taken place (password transfer, data transfer, etc.). Institutions above the law (intelligence services at the highest level) can presumably read everything, if necessary by means of [pure computing power](#) (with computer technology blocked to the public). If one disregards the possibility of direct or indirect [backdoors in computer systems](#) and possible inherent mathematical weaknesses in algorithms, everything currently amounts to a [prime factorization](#), which in turn represents a pure computational power problem. Read the WP-article “[Post-quantum cryptography](#)” to get an overview of critical factors and aspects. A translated excerpt of a German article (2016/2017): “[...] *To emphasize once again: Quantum computers mean the end of all currently established public key procedures for digital signatures and key exchange, among other*

things. This means that a considerable part of the foundation of current crypto systems is completely broken away. Adequate replacement is not yet in

sight.[...]". (Jürgen Schmidt, "Kryptographie in der IT - Empfehlungen zu Verschlüsselung und Verfahren", sub-article "Elliptische Kurven Verschlüsselung", heise Security).

The encrypted transmission of webpages is carried out with the help of (website)certificates.

The international website certificate issuer industry as a whole is now facing severe criticism. On the one hand, because it happens again and again that individual CAs (Certification Authorities) award certificates to unauthorized persons due to inadequate verification procedures at the application stage. On the other hand, due to hacker break-ins by which third parties sometimes succeeded unnoticed in issuing formally recognized certificates for various popular websites or for companies. The criticism can be summed up as follows: the current technical basis of the certificate system is too vulnerable to such errors and attacks and cannot be used effectively and quickly enough to take countermeasures. Search terms such as "SSL debacle" lead to discussion posts and articles in the World Wide Web. Extensive sources of information: www.cabforum.org; "The EFF SSL Observatory": www.eff.org/observatory.

1.5.3 The Qualified Electronic Signature

Cryptographic checksums are part of the technical basis of the personal proof of authenticity of documents, the so-called *qualified* (= legally recognized) electronic signature as well as of numerous cryptographic procedures

in general, i.e. the encryption of (message) documents. Knowledge of the relevant contact points and sources of information is an advantage.

1.5.4 Central Information Sources on Applied Cryptography

Cryptography at the international business level: Bert-Jaap Koops' "Cryptography Law Survey" provides information on the basic legal situation regarding cryptography in the individual states and administrative structures of the world. Each entry is provided with a

comprehensive source collection: www.cryptolaw.org.

The history of cryptography: Wikipedia provides detailed information on the publicly known history of cryptography, among other things: "History of cryptography".

1.6 Further Application Possibilities

Checksums have also been used in electrical engineering for many decades, among other things to guarantee error-free data transmission.

1.7 Possibilities of Abuse

1.7.1 Fully automated Identification of consumed Content

Cryptographic checksums can also be used for questionable purposes. In the past, the media player of a software manufacturer is said to *have sent* unsolicited hash codes of the played files.

Theoretically, a fully automatic comparison with database tables could determine whether content used was licensed and which political films and sound files a user prefers to watch. Individual computers could be identified by a combination of features, and checksums could of course also be sent at operating system level. The same would also be possible with proprietary PDF programs.

Alternatively, [free PDF viewers](#) and [free media players](#) are available, such as the highly recommended [VLC media player](#). However, comprehensive, maximum security always requires a [free operating system basis](#).

In [computer forensics](#) as well as in countless other information technology areas, the formation or retrieval of cryptographic checksums is omnipresent. A thoroughly constructive application, especially with regard to the preservation of evidence in computer crimes, such as network break-ins.

In dictatorships or in occupied, foreign-controlled countries, there is a danger that "hard disk inspections" will be carried out locally or remotely. Through backdoors from software and hardware manufacturers, cryptographic checksums of all existing hard disk files can be routinely generated and

then automatically compared with checksums of “[indexed](#)” and [censored content](#), such as political educational films. In this way, it can be quickly and effectively checked whether citizens tend to cultivate their own opinions and whether they consume

political content that contradicts officially proclaimed dogmas. Freethinkers are easy to locate.

If all checksums of new and changed files created by the user are sent automatically on a regular basis, it could also be determined afterwards in which network or on which computer a document was created for the first time.

The amount of data is tiny and, if additionally encrypted, practically indecipherable. Changing the file name does not change the checksum. Other features, such as hardware and software configurations (including unlicensed programs), can also be analyzed and “reported” fully automatically.

It cannot be ruled out that computers from regime critics may be paralyzed from a distance. For example, the remote installation of a single [process](#) working permanently from the start of the computer, which consumes the entire computing power, a process which can possibly still be recognized via [ps -e](#) but can no longer be terminated, is sufficient (see also [pstree](#)). The installation of functional program errors is another possibility to partially or completely paralyze the computers of dissenters.

Sabotage measures could also be automated if statistical analyses in the background show that a high level of political educational films is consumed and that there is a “danger” of distributing or publishing critically questioning articles

1.7.2 Software Activation and Computer Identification via Electronic Fingerprints

The Free Software Foundation ([FSF](#)) lists computer features in an [article on privacy](#) that can be used to uniquely identify and recognize a computer. Such identification data will most likely be hashed and archived in a database. With some proprietary products, software activation ([product activation](#)) is linked to the determined hardware configuration. The attempt to install the purchased software on a second computer at the same time often fails.

1.8 Signing and Encryption of Files

1.8.1 Real-existing Protection with Publicly approved Encryption Procedures

It cannot be excluded that officially recommended cryptographic algorithms raised to standards have inherent mathematical weaknesses to facilitate decryption by intelligence services. Probably really advanced computer technology is kept out of the public eye or generally blocked in order to guarantee secret services a computational advantage. Under this aspect and in view of the highly probable influence on the design of company products (software and hardware back door problems), the effectiveness of the existing encryption practice is questionable; even if open, free IT infrastructure is used throughout.

Consistently applied signing and encryption, however, fend off at least some of the potential business attackers and prevent direct data access in the event of theft or loss of data media.

The reduction of complexity at program and operating system level is another central factor. In addition, old and very old computers on which still outdated proprietary operating systems and programs are installed can be modernized and security optimized with continuously updated specialized lightweight free operating system distributions.

2 Free checksum programs

[CyoHash](#) and [Jacksum](#) stand out from the large number of free graphical checksum programs. Jacksum, released under an [OSI-certified](#) Free Software license, the [GPL](#), listed in the [FSF](#) directory and based on [Java](#), runs on many operating system platforms ([software features](#)). It is therefore also suitable for [heterogeneous IT infrastructures](#) of company networks. Numerous internationally common checksum procedures are taken into account, the file manager integration ensures comfortable operation.

Jacksum file manager versions are available for [GNOME](#), [KDE](#), [ROX](#) and [Thunar](#) (Unix/BSD, GNU/Linux) as well as for the [Windows Explorer](#) of MS-Windows and the [Finder](#) of Apple Macintosh. The program author Johann Löfflmann maintains a webpage

with detailed information, [Jacksum.net](#). Suggestions for program extensions can be submitted, the exchange among users is also encouraged.

[CyoHash](#), an extension for MS-Windows File Manager [Explorer](#), offers only a fraction of the algorithms, but also supports the modern [SHA2 algorithms](#) and works without Java.

Among the many cross-platform text mode programs for BSD/Unix, GNU/Linux and MS-Windows, [sha1sum](#) and [shasum](#), respectively, is distinguished by its support of the modern SHA-2 and [SHA-3 algorithms](#).

A comparative overview of numerous free and proprietary checksum programs can be found in the WP article [“Comparison of file verification software”](#).

2.1 Cyohash

Cyohash is a free file manager extension for the [MS Windows Explorer](#). Download Cyohash either from the [official project page](#), or from a reputable program directory.

2.1.1 Checksum Formation

Point with the mouse arrow to the corresponding file and click on the right mouse button. In the menu that appears, press “Cyohash” and select a checksum algorithm, e.g. SHA-256.

A window appears showing the file name, directory path, checksum method, and cryptographic checksum, or a table listing this data. A separate entry appears for each checksum formed.

[Illustration to follow]

You now have the option of comparing the determined checksum with a target value. To do this, double-click on the respective table entry to open the corresponding window.

An application example: Program files offered on the Internet are almost always published together with their

checksums. After downloading such an [executing file](#), preferably from the official program project page, [copy](#) the corresponding cryptographic checksum published there, paste it into the input line (“Validate:”) at the bottom of the program window and press “OK”. If the values match, the input line turns green, otherwise red.

2.1.2 Create Checksums for Multiple Files

Point either to a table entry or to an empty table line and press the right mouse button to display further functions.

You can create checksums for several files at the same time. Select the function “Hash File(s)...”. A window opens in which you can select the desired checksum method. Then press “Browse...” and a Windows directory window opens. Select the files in the corresponding directory, [hold down the CTRL key](#) and press the mouse button to highlight the individual files. Finally click on “Open”. The Windows directory window then disappears, the CyoHash window appears in the foreground, in which you confirm your file selection with “OK”. All checksums are then displayed in the table window.

2.2 Jacksum

Jacksum can be called via the file manager or as a command line program. The file manager version does not require an installed command line version, it works independently.

In the following the application is described under [GNOME](#), [KDE](#) and [MS-Windows Explorer](#). The installation and usage under [ROX](#), [Thunar](#) and others as well as [Apple's Finder](#) is described in the [official installation section](#) and in the [official question and answer section](#) of the Jacksum website.

2.2.1 Installation of Java and Jacksum

2.2.1.1 Installation of Java

In Unix and [unix-like](#) operating system [distributions](#) Java is mostly already included, in the [free](#) variant [OpenJDK](#). As an MS Windows user, you call up a search engine page, for example Google, and type in “Java” or “JRE”, which is the abbreviation for “[Java Runtime Environment](#)”. This step is not necessary if a Java Runtime Environment already exists.

Example for a Java installation under MS-Windows, in the Anglosphere:

1. Go to [google.com](#) and enter “Java”
2. In the first place appear the entries of the official Java manufacturer Oracle ([Java.com](#)), for different operating system platforms. Go to one of these websites and follow the instructions there or alternatively:
3. Go back to the (Google) search menu and type “Java Runtime Environment” to go to the official download page

If the Java manufacturer offers a preselection for the (co-)installation of a Yahoo [toolbar](#), deactivate the checkbox by clicking on it, so that the checkbox remains empty (Google search for this topic: [Java installation Yahoo toolbar](#)). Such toolbars can be removed at any time. Google search for related articles: [uninstall toolbars](#).

2.2.1.2 Installation and Application of Jacksum under MS-Windows

Installation and use of Jacksum are described in detail in the [installation section](#) of the official program page as well as in the `readme.txt` files attached to the Jacksum download.

Select on the official web presence, [jacksum.net](#), the column “[Download](#)”. In the section “File browser integration (optional)”, download the “[...]windows-explorer-integration-[...]” file (file name varies with the [software version number](#)), a [ZIP file](#). Open it directly by double-clicking or right-clicking. Open the uncompressed folder, read the file `readme.txt` or start the installation by double-clicking on the [executable file](#) “Jacksum Windows Explorer Integration.exe”. A window appears prompting you to extract all files.

[Illustration to follow]

Now go to the fully extracted folder, there you will see the [.exe installation file](#), symbolized by a green circle.

[Illustration to follow]

Double-click to start the program installation. From then on you can right-click → “Send to” → [Procedure, e.g.: “Jacksum - 3) All algorithms”] to form cryptographic checksums. The checksum(s) appear in a separate text window.

[Illustration to follow]

2.2.2 Application under KDE Konqueror and KDE Dolphin

Open the file manager. To select the file, click the right mouse button → “Actions” → “Jacksum” → [select desired function].

2.2.3 GNOME Nautilus

Open the file manager. To select the file, right-click → “Scripts” → “Jacksum” → [select desired function].

The application possibilities of Jacksum are numerous. The [command line](#) version unfolds the full potential of this excellent software, including interaction with other programs. The provision of an open [application programming interface \(API\)](#) promotes broad acceptance.

2.3 Console-based Checksum Generation

In the [text mode \(console application\)](#) cryptographic checksums can be formed under any operating system, practically all manufacturers of [proprietary](#) and of [free](#) operating systems offer appropriate programs by default.

Since the introduction of Microsoft's [Windows PowerShell](#), MS Windows users have been able to use basic [file operation commands from the BSD/Unix world](#) by default ([comparison](#)) in addition to the classic [MS DOS commands](#) (extended in `cmd.exe`). The [GNU core Utilities](#) also make it possible to use other Unix standard programs under MS Windows ([List of Unix commands](#)).

2.3.1 On-board SHA Algorithms under MS-Windows, Unix/BSD and GNU/Linux Systems

SHA algorithms are pre-installed as standard on numerous [Unix and unix-like](#) and MS-Windows systems, in particular the widely used, platform-independent [shasum](#), also available for MS Windows, which supports SHA-2 as well as SHA-3 algorithms.

Creating cryptographic checksums under MS-Windows, with the on-board PowerShell: The [PowerShell](#) supports many basic commands of the [Unix shell](#); for an overview, skim the corresponding [comparison section in the WP-article on the PowerShell](#).

In addition, current SHA algorithms are also supported by using the “Get-FileHash” command. For official information, see the [Microsoft article](#) of the same name; the default is [SHA-256](#).

1. *Start program:* Go to the search field of the [Windows 10 taskbar](#) and enter: PowerShell. The command line environment opens. To adjust the font size, right-click on the upper window frame and select the “Properties” menu item. In the “Font” [tab](#), under “Size”, you now specify the font size, then press “OK”.

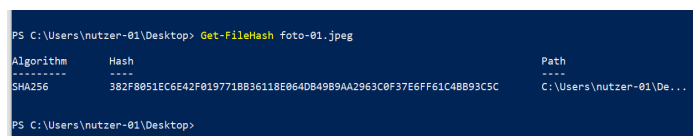
2. *Changing to the corresponding directory:* Now change to the [file directory](#) containing the file for which the checksum is to be formed. In the example [prompt](#) used ([see screenshot](#)), “PS C:\Users\nutzer-01>”, “Users” stands for the [Windows “Users” folder](#), “nutzer-01” in this example is the name of the [user account](#) of the logged-in user.

Enter [ls](#) to display a folder overview: “PS C:\Users\user-01>ls”. The following is a directory listing that includes the central folders “Desktop”, “Documents”, and “Downloads” you can move the contents up and down using the mouse wheel or sliders. In our example, we will now switch to the desktop folder, whose files are also displayed on the [desktop screen](#) by default; we will use the [cd](#) (change directory) command to do this:

“PS C:\Users\user-01> cd Desktop”, now we are in the Desktop folder: “PS C:\Users\user-01\Desktop>”; by typing [ls](#) again, you can view its contents.

3. *Forming the checksum of a file.* In our example, the image file foto-01.jpeg is located on the desktop, from which we want to create a checksum. Like the [Unix shell](#), the PowerShell also has an auto-complete function ([“command-line completion”](#)).

It is sufficient to type “get-f” in lower case at the command line (“PS C:\Users\user-01\Desktop> get-f”) and then press the [Tab key](#) (for auto-completion), whereupon the command appears in full length and in correct upper and lower case: “PS C:\Users\user-01\Desktop> Get-FileHash”. We add the file name (you can also use auto-complete for this): “PS C:\Users\user-01\Desktop> Get-FileHash foto-01.jpeg” and get the SHA-256 checksum ([see screenshot](#)).



```
PS C:\Users\nutzer-01\Desktop> Get-FileHash foto-01.jpeg

Algorithm Hash Path
-----
SHA256 382F8051EC6E42F019771B836118E064DB49B9AA2963C0F37E6FF61C4BB93C5C C:\Users\nutzer-01\De...
```

Abb.: Checksum generation with the PowerShell
(Click to enlarge)

Under unix-like systems:: Open a [command-line environment \(shell\)](#), type “sha”, and then press the tab key for the [command-line completion](#) to see all existing SHA procedures:


```
> sha
> sha1sum sha224sum sha256sum sha384sum
  sha512sum shasum
> sha
```

Go to the appropriate directory, select a procedure, add the name of the desired file, and press Enter. In the following example, the SHA256 checksum of the test.html file is formed:

```
> sha256sum test.html
> e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b
  934ca495991b7852b855 test.html
>
```

The SHA256 checksum value and the name of the associated file are displayed. Implementation and command names may vary. Virtually all Unix and unix-like systems and distributions have corresponding pre-installations. There are dozens of free programs for forming cryptographic checksums, both graphical and text-based, e.g. at sourceforge.net (WP-article).

2.3.2 Text Mode Programs for Professional Computer Use

The use of the [command line environment](#) enables highly effective work on the computer. Some applications are written exclusively for today's usually emulated [text mode](#), others offer a text-based [program interface](#) in addition to the graphical one.

Text-based programs enable the most effective use of computers. Andreas Poisel's www.automatisch.cc offers an excellent overview (German).

In addition to the graphical [“desktop environments”](#) and application programs, Linux distributions also contain [“text mode software”](#) with a [“text-based user interface”](#)

as standard. They are extremely powerful and belong to the preferred tools of many users, administrators, IT professionals and scientists.

Websites and recommended articles about text mode programs: [„automatisch.cc :: freie Textmode-Software”](#) • [Websites about text mode](#) • [“Console application”](#) • [„Text-based \(computing\)”](#) • [“Shell \(computing\)”](#) • [“Command-line interface”](#) • An excellent introduction: Floss Manual [“Introduction to the Command Line”](#) • [“GNU Screen”](#) • [“Computer terminal”](#).

Endnotes

[1] [“Announcing Approval of Federal Information Processing Standard \(FIPS\) 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, and Revision of the Applicability Clause of FIPS 180-4, Secure Hash Standard”](#), “A Notice by the National Institute of Standards and Technology on 08/05/2015”. [\[back\]](#)

[2] February 02, 2014, extract of the short information about the procedures supported by RHash, [“Hash Functions”](#), extract: “GOST is a hash function defined in Russian national standard [GOST R34.11-94](#). It has two widely used versions with testparameters and CryptoPro ones. It's relatively slow, but it is used for digital signature in Russian State banks and enterprises. Hash is a hexadecimal string of length 64.” [\[back\]](#)

[3] The term “freeware” is not clearly defined. It may or may not refer to [“Free Software”](#) (program text/[source code](#) is available, may be modified and distributed). Tendently prevailing, it refers to software distributed free of charge, but whose program text remains unpublished. There are several [categories of free and nonfree software](#). [\[back\]](#)

[4] “Packing programs” are used to create archive files or file archives, called “archives” for short. In particular, they offer the possibility of combining several individual files as well as nested (i.e. subfolder containing) file folders into a single file. Thus, for example, entire websites and extensive personal compilations of documents can be conveniently attached to an e-letter as individual files, or stored on a data carrier. WP-article [“Data compression”](#), packing program recommendation [“7-Zip”](#), free data compression software: [“Category:Free data compression software”](#). [\[back\]](#)

[5] Regarding the authenticity of spoken language, everyone is advised to read the sections “Technical details” and “Concerns” in the WP article “[Adobe Voco](#)”. [\[back\]](#)

[6] Certificates are IDs for the Internet (networks in general), mostly e-mail or website certificates. They can also be used to encrypt data transmission (“https://[...]”), which makes it relatively impossible for people and organizations subject to the law to read them. (WP-article: “[Public key certificate](#)”). [\[back\]](#)