

SelfLinux-0.10.0



Glossar

Autor: Mike Ashley ()
Formatierung: Matthias Hagedorn (matthias.hagedorn@selflinux.org)
Lizenz: GFDL

1 Glossar

1.1 3DES

Triple DES. ► [Symmetrischer](#) ► [Verschlüsselungsalgorithmus](#), der auf einer dreifachen Verschlüsselung mit ► [DES](#) basiert. Gilt nach heutigen Maßstäben als sicher. Wird vom ► [OpenPGP](#)-Standard zwingend vorgeschrieben und gilt daher als der kleinste gemeinsame Nenner unter den verwendeten Verschlüsselungsalgorithmen. 3DES hat eine Schlüssellänge von 168 Bit, die wirksame ► [Schlüssellänge](#) ist aber aufgrund des eingesetzten Verfahrens 112 Bit.

1.2 Algorithmus

Allgemein: (mathematisches) Verfahren, das in kleinste Teilschritte/Handlungsanweisungen unterteilt ist.

Siehe auch: ► [Verschlüsselungsalgorithmus](#).

1.3 ASCII

American Standard Code for Information Interchange. Standard-Zeichensatz für das Englische Alphabet. Besteht aus 128 Zeichen, jedes Zeichen wird durch eine 7 Bit lange Zahl dargestellt.

Siehe auch: ► [Binär](#).

1.4 ASCII-armor

Die Ausgabe erfolgt nicht in ► [binärer](#) Form, sondern in Form von am Bildschirm darstellbaren ► [ASCII](#)-Zeichen.

1.5 Asymmetrische Verschlüsselung

Im Gegensatz zur ► [Symmetrischer](#) Verschlüsselung wird bei asymmetrischen Verfahren zum Verschlüsseln ein anderer Schlüssel eingesetzt als zum Entschlüsseln. Zum Verschlüsseln und Überprüfen von ► [digitalen Signaturen](#) wird der ► [öffentliche Schlüssel](#) eingesetzt, zum Entschlüsseln und signieren der ► [geheime Schlüssel](#). Asymmetrische Verschlüsselung wird bei ► [Public-Key](#) Verfahren eingesetzt, um den eigentlichen symmetrischen Sitzungsschlüssel sicher auszutauschen.

1.6 Authentisierung

Das Beglaubigen der Identität durch die Eingabe eines ► [Mantra](#). Dadurch wird verhindert, dass sich eine andere Person als Urheber eines Dokumentes ausgeben kann oder ein für einen bestimmten Empfänger verschlüsseltes Dokument unbefugt lesen kann.

1.7 Benutzer-ID

(Engl. User-ID) Identifikation des Benutzers durch Name und Email-Adresse

1.8 Binär

Im Zweier-Zahlensystem (Binärsystem) dargestellt. Intern werden alle Daten in einem Computer binär dargestellt.

1.9 Blowfish

Von [Bruce Schneier](#) entwickelter, frei verwendbarer [symmetrischer](#) Verschlüsselungsalgorithmus mit 128 Bit [Schlüssellänge](#), der Teil der [OpenPGP-Spezifikation](#) ist.

1.10 Brute Force

Angriff auf verschlüsselte Daten, bei dem alle möglichen Schlüsselkombinationen durchprobiert werden. Brute-Force Verfahren sind extrem rechenaufwendig. Selbst wenn alle Rechner der Welt zusammengeschaltet wären, würde das Durchprobieren aller Kombinationen bei einem 128-Bit-Schlüssel einige Milliarden Jahre dauern.

1.11 CAST5

[Symmetrischer](#) [Verschlüsselungsalgorithmus](#) mit 128 Bit [Schlüssellänge](#). In der [OpenPGP-Spezifikation](#) vorgeschrieben.

1.12 Certification Authority

Zertifizierungsinstanz innerhalb eines hierarchischen Zertifizierungsmodells. Certification Authorities lassen sich auch problemlos in das von GnuPG favorisierte Modell des "[Web of Trust](#)" einbeziehen.

Siehe auch: [Web of Trust](#).

1.13 Cracker

Person, die vorsätzlich, unbefugterweise und oft mit bössartiger Absicht in fremde Rechnersysteme eindringt, im deutlichen Gegensatz zu "Hacker", worunter man allgemein einen gutmeinenden Computer-Freak versteht (siehe hierzu auch [RFC 1983](#)).

1.14 default

Standard, Standard-Einstellung, Voreinstellung.

1.15 DES

Digital Encryption Standard. [Symmetrischer](#) [Verschlüsselungsalgorithmus](#) mit einer [Schlüssellänge](#) von 56 Bit. Kann nach dem heutigen Stand der Technik - wenn auch mit erheblichem Aufwand - geknackt werden.

1.16 Diffie-Hellmann

Public-Key Algorithmus. Wird zum sicheren Austausch von Schlüsseln verwendet.

1.17 Digest Algorithmus

► [Hashalgorithmus](#).

1.18 Digitale Signatur

Auch: Digitale Unterschrift. Aus dem ► [geheimen Schlüssel](#) und einer Datei wird durch Anwendung einer ► [Hash-Funktion](#) eine eindeutige Zeichenfolge gebildet. Mit Hilfe des ► [öffentlichen Schlüssels](#) kann nun jeder überprüfen, ob die Datei tatsächlich von dem angegebenen Urheber bzw. Absender stammt und ob der Inhalt verfälscht wurde. Die digitale Signatur ermöglicht die Integrität und Authentizität eines elektronischen Dokumentes, unabhängig vom Datenformat zu verifizieren.

1.19 DSA

► [Digital Signature](#) Algorithm. Ein von der ► [NSA](#) entwickelter, sehr sicherer ► [Algorithmus](#) zum Signieren von Daten. DSA verwendet den Hash-Algorithmus SHA1.

1.20 Eigenbeglaubigung

Auch Selbstunterzeichnung. Indem der Benutzer seinen ► [öffentlichen Schlüssel](#) sowie die ► [Benutzer-ID](#) selbst mit seinem ► [geheimen Schlüssel](#) unterzeichnet, lassen sich Verfälschungen daran sehr leicht feststellen und bestätigt er deren Authentizität.

1.21 Einweg-Hash

Eine nicht umkehrbare ► [Hashfunktion](#). Es ist nicht möglich, aus der durch die ► [Hashfunktion](#) erzeugten eindeutigen Prüfsumme die ursprünglichen Daten wieder herzustellen oder auch nur Rückschlüsse darauf zu ziehen.

1.22 ElGamal

Auch ELG-E ► [Asymmetrischer](#) Verschlüsselungs-Algorithmus der sowohl zum Verschlüsseln als auch zum ► [Signieren](#) benutzt werden kann. Seit 1997 nicht mehr von Patenten gedeckt. Dieser Algorithmus gilt nach den heutigen Maßstäben als sicher und muß von jedem ► [OpenPGP](#)-System unterstützt werden.

1.23 Entropie

Begriff aus der Thermodynamik. Maß für die Unordnung eines Systems. Zum Erzeugen echter Zufallswerte benötigt GnuPG Entropie. Diese kann man beispielsweise durch (willkürliche) Festplattenzugriffe, Mausbewegungen oder Tastatureingaben erzeugen.

1.24 Falltür-Algorithmus

(Engl. Doortrap Algorithm) Ein ► [Algorithmus](#), der leicht zu berechnen ist, dessen Umkehrfunktion aber sehr schwer zu berechnen ist. So ist es z.B. leicht, zwei Primzahlen miteinander zu multiplizieren, um eine Nichtprimzahl zu erhalten, es ist aber schwer, eine Nichtprimzahl in ihre Primfaktoren zu zerlegen.

1.25 Fingerabdruck

(Engl. fingerprint) Eindeutige Prüfsumme (► [Hash](#)) des ► [öffentlichen Schlüssels](#); ist wesentlich kürzer als der Schlüssel selbst. Wird zum Überprüfen bzw. Verifizieren eines ► [öffentlichen Schlüssels](#) herangezogen.

1.26 Freie Software

Software, die allen Anwendern die Freiheit gibt, diese nach belieben - auch kommerziell - zu nutzen, den [Quellcode](#) einzusehen und nach eigenen Vorstellungen abzuändern, und die Software in veränderter oder unveränderter Form - ohne ihr allerdings eigene Einschränkungen aufzuerlegen - an andere weiterzugeben. Beispiele für Freie Software sind [Linux](#) und GnuPG. Siehe auch <http://www.gnu.org/philosophy/free-sw.html>.

1.27 Geheimschlüssel

(Engl. Secret Key, Private Key) Bei [asymmetrischen](#) Verfahren der Hauptschlüssel, der sowohl zum Entschlüsseln des [Geheimtextes](#), zum [digitalen Signieren](#) von Dokumenten als auch zur Generierung des [öffentlichen Schlüssels](#) und der [Widerrufurkunde](#) verwendet wird. Zum Verschlüsseln sowie zum Überprüfen einer [Signatur](#) genügt der [öffentliche Schlüssel](#).

1.28 Geheimtext

Die mit Hilfe eines [Verschlüsselungsverfahrens](#) verschlüsselten Daten.

Siehe auch: [Klartext](#).

1.29 GNU

Gnu's Not Unix. Abkürzung für das seit 1984 bestehende GNU-Projekt der [Free Software Foundation](#), dessen Ziel die Schaffung eines auf [freier Software](#) basierenden, [Unix-ähnlichen](#) Betriebssystems ist. [LINUX](#) beruht in großen Teilen auf GNU-Software. Siehe auch <http://www.gnu.org>

1.30 GNU-GPL

GNU General Public License. Eine Lizenz für [Freie Software](#), der auch GnuPG unterliegt. Jeder kann Software die unter der GPL steht frei benutzen, modifizieren und weitergeben; die einzigen Einschränkungen sind, dass man keine weiteren Einschränkungen bei der Weitergabe auferlegen darf und dass die Lizenz an sich nicht verändert werden darf.

1.31 GnuPG

GNU Privacy Guard. [Freie](#), dem [OpenPGP-Standard](#) entsprechende Verschlüsselungssoftware.

1.32 Hashfunktion

Auch kryptographische Prüfsumme oder Message Digest. Eine Hashfunktion ist eine Funktion, die aus einer Datei eine eindeutige Prüfsumme errechnet. Beispiele für Hashalgorithmen sind [SHA1](#) und [MD5](#).

1.33 Hybride Verschlüsselung

Verschlüsselungsverfahren, bei dem sowohl [symmetrische](#) als auch [unsymmetrische](#) [Verschlüsselungsalgorithmen](#) eingesetzt werden. Bei GnuPG (und [PGP](#)) werden beispielsweise die eigentlichen Daten mit einem zufällig erzeugten symmetrischen [Sitzungsschlüssel](#) verschlüsselt. Dieser [Sitzungsschlüssel](#) wird dann, um einen sicheren Schlüsseltausch zu ermöglichen, mit dem [öffentlichen](#)

Schlüssel des Empfängers verschlüsselt und von GnuPG zu einem Paket zusammengepackt. Auf der Empfängerseite entschlüsselt GnuPG zuerst mit dem ► **geheimen Schlüssel** des Empfängers den ► **Sitzungsschlüssel**, mit dem die ursprünglichen Daten wieder hergestellt werden können.

1.34 IDEA

International Data Encryption Standard. ► **Symmetrischer** ► **Verschlüsselungsalgorithmus** mit 128 Bit ► **Schlüssellänge**. Der IDEA-Algorithmus ist patentiert; kommerzieller Einsatz erfordert den Erwerb einer Lizenz. IDEA gilt nach den heutigen Maßstäben als sicher und kann von ► **OpenPGP**-Systemen optional unterstützt werden. Da man davon ausgehen kann, dass IDEA nicht von allen ► **OpenPGP**-Systemen unterstützt wird, ist von seiner Verwendung abzuraten.

1.35 IETF

Internet Engineering Task Force. Gremium das technische Standards für das Internet koordiniert. In den sogenannten Workinggroups der IETF, in denen sich Entwickler und Wissenschaftler aus eigenen Antrieb organisieren, werden die Standards erarbeitet und in RFC ("Request for Comments") genannten Dokumenten beschrieben. Siehe auch  <http://www.ietf.org>.

1.36 Key Escrow

Schlüssel hinterlegung. Maßnahme um beispielsweise ein Mitlesen verschlüsselter Nachrichten durch staatliche Stellen oder den Arbeitgeber zu ermöglichen.

1.37 Key Recovery

Die Möglichkeit durch eine in die Verschlüsselungssoftware eingebaute "Hintertür" für Unbefugte (staatliche Stellen, Arbeitgeber) das Wiederherstellen des ► **geheimen Schlüssels** zu vereinfachen. Key Recovery stellt einen Eingriff in das Grundrecht auf informationelle Selbstbestimmung dar und wird von GnuPG nicht unterstützt. Da GnuPG ► **freie Software** ist, kann der ► **Quelltext** daraufhin überprüft werden.

1.38 Keyserver

Zentrale Internet-Datenbanken zur Verwaltung und Veröffentlichung von ► **öffentlichen Schlüsseln**. Ein Keyserver führt keine Zertifizierung der verwalteten Schlüssel durch, hierfür bedarf es zusätzlicher Maßnahmen wie des ► **Web of Trust**.

1.39 key space

Die Anzahl an möglichen Schlüsseln. Je länger der Schlüssel ist, desto größer ist auch die mögliche Anzahl an verschiedenen Schlüsseln und desto schwieriger ist es, den Schlüssel durch Raten oder ► **Ausprobieren** herauszubekommen.

1.40 Klartext

Allgemein: unverschlüsselte Daten.

Siehe auch: ► **Geheimtext**.

1.41 Kompromittierung

Unbeabsichtigte oder unautorisierte Offenlegung des (Geheim-)Schlüssels bzw. der verschlüsselten Daten.

1.42 Kryptographie

Wissenschaft von der ► [Verschlüsselung](#) von Daten und deren Anwendung.

1.43 LINUX

Betriebssystem der ► [Unix](#)-Familie, das als ► [freie Software](#) für verschiedene Rechner-Plattformen verfügbar ist. Benannt nach dem Initiator Linus Torvalds, der 1991 die erste Version im Internet veröffentlicht hat, wird es von einem weltweiten Netzwerk von Programmierern ständig weiterentwickelt. Linux basiert in großen Teilen auf dem ► [GNU-Projekt](#) und ist der ► [GNU GPL](#) unterstellt. LINUX ist derzeit das meistverwendete Betriebssystem auf dem Gebiet der Internetserver und gewinnt immer mehr Marktanteile im Desktop-Bereich.

1.44 Mantra

Ein Passwort-Satz. Der ► [geheime](#) Schlüssel ist bei GnuPG noch einmal selbst mit einem Mantra geschützt. Ohne das Mantra kann man den ► [geheimen](#) Schlüssel weder zum Entschlüsseln noch zum ► [Signieren](#) verwenden. Ein sicheres Mantra sollte möglichst lang sein, möglichst wenig Wörter aus dem Wörterbuch/Lexikon enthalten und trotzdem leicht zu behalten sein. Um sich das Mantra zu merken, sollte man es niemals aufschreiben, sondern quasi wie ein "Mantra" still in sich "hineinmurmeln".

1.45 MD5

Message Digest 5. Ein kryptographisch sicherer ► [Hashalgorithmus](#).

1.46 NSA

 [National Security Agency](#). Amerikanischer Geheimdienst, der sich vorrangig mit ► [Kryptographie](#) und dem weltweiten gezielten Abhören der elektronischen Kommunikation beschäftigt.

1.47 Öffentlicher Schlüssel

(Engl. public key) Bei ► [asymmetrischen](#) und ► [hybriden](#) Verschlüsselungsverfahren der frei zugängliche Schlüssel. Mit dem öffentlichen Schlüssel können Daten verschlüsselt und ► [Signaturen](#) überprüft werden. Zum ► [Signieren](#) und Entschlüsseln ist der ► [geheime Schlüssel](#) erforderlich.

1.48 OpenPGP

Protokoll, das den Austausch von verschlüsselten Daten, ► [Signaturen](#) und Schlüsseln regelt. Spezifiziert in RFC 2440.

1.49 PGP

Pretty Good Privacy. Eine von Philip Zimmermann in den USA entwickelte, weit verbreitete Verschlüsselungssoftware. PGP benutzt den patentierten ► [Algorithmus](#) ► [IDEA](#) und fordert für kommerzielle

Anwender den Erwerb einer Lizenz. Der Quellcode von PGP ist öffentlich nicht verfügbar, die Integrität der Software wird von Experten in Frage gestellt.

1.50 Primzahl

Auch Primfaktor. Zahl, die nur durch sich selbst oder die Zahl 1 teilbar ist. Die Zerlegung in Primfaktoren spielt bei vielen Verschlüsselungsalgorithmen eine zentrale Rolle.

1.51 Privatsphäre

(Engl. Privacy) Im Rahmen dieses Handbuches das Schützen vertraulicher Informationen vor dem Zugriff oder der Manipulation durch Dritte. Da Daten, die über das Internet oder ein lokales Netzwerk verschickt werden, leicht mitgelesen, abgefangen oder manipuliert werden können, und Daten auf einem nicht vernetzten Einzelplatzrechner in der Regel auch nicht sicher vor unbefugten Zugriffen sind, ist die einzige Möglichkeit, seine Privatsphäre zu schützen, eine wirkungsvolle ► [Verschlüsselung](#).

1.52 Public-Key-Verschlüsselung

► [Hybrides](#) oder ► [asymmetrisches](#) Verschlüsselungsverfahren mit einem ► [öffentlichen](#) (engl. public key) und einem ► [geheimen](#) Schlüssel (engl. secret key). Der ► [öffentliche](#) Schlüssel wird zum Verschlüsseln und Überprüfen von ► [Signaturen](#) benötigt, der ► [geheime](#) zum Entschlüsseln und ► [Signieren](#).

1.53 Quelltext

(Engl. source code) Ein Computer-Programm in seiner ursprünglichen, vom Menschen lesbaren Textform; kann nicht direkt vom Prozessor ausgeführt werden, sondern muß vorher von einem Compiler oder Interpreter in ► [Binärcode](#) übersetzt werden. Kann von Programmierern oder Systemadministratoren verändert, den eigenen Anforderungen angepaßt und auf eventuelle Sicherheitsrisiken geprüft werden. Der Quelltext der meisten gebräuchlichen kommerziellen Software ist nur dem Hersteller zugänglich; erst in letzter Zeit gewinnt ► [freie Software](#), bei welcher der Benutzer Zugriff auf die Quelltexte hat, zunehmend an Bedeutung. Der Quellcode von GnuPG, welches der ► [GNU GPL](#) unterliegt ist jedermann frei zugänglich.

1.54 RIPE-MD-160

Ein ► [Hash-Algorithmus](#).

1.55 RFC

Request For Comments. Dokumente, die (unter anderem) technische Standards für das Internet beschreiben. Die RFCs werden von den sog. Workinggroups der ► [IETF](#) erarbeitet. Der Standard für OpenPGP beispielsweise ist in RFC 2440 spezifiziert. Weitere Informationen sowie alle RFCs finden Sie unter <http://www.rfc-editor.org>.

1.56 RSA

Ein ► [Algorithmus](#) zum ► [Signieren](#) und ► [asymmetrischen](#) Verschlüsseln von Daten. RSA steht für Rivest, Shamir, und Adelman, die Erfinder des ► [Algorithmus](#). Die Spezifikation von ► [OpenPGP](#) unterstützt die optionale Verwendung von RSA. Dieser Algorithmus ist von Patenten geschützt und daher nicht frei verwendbar.

1.57 Schlüssel

Datensequenz, die benutzt wird, um mit einer Verschlüsselungssoftware aus dem ► [Klartext](#) ► [Geheimtext](#) zu erzeugen (Verschlüsselung) und um aus dem ► [Geheimtext](#) den ► [Klartext](#) wieder herzustellen (Entschlüsselung). Auch zum ► [Signieren](#) und Überprüfen einer digitalen ► [Signatur](#) wird ein Schlüssel benötigt.

1.58 Schlüssel ID

(Engl. key ID) Eindeutige Kennzeichnung eines Schlüssels. Bestehend aus ► [Schlüssellänge](#), verwendetem ► [Algorithmus](#), den letzten 8 Stellen des ► [Fingerabdrucks](#), dem Erzeugungsdatum und der ► [Benutzer-ID](#).

1.59 Schlüsselbund

(Engl. key ring) Eine Sammlung ► [öffentlicher](#) und ► [geheimer](#) Schlüssel, wird als Schlüsselbund bezeichnet, bei GnuPG die Dateien pubring.gpg und secring.gpg. Da ein Teilnehmer für jeden Gesprächspartner, dem er verschlüsselte Email schicken will, dessen ► [öffentlichen](#) Schlüssel seinem öffentlichen Schlüsselbund hinzufügt, kann dieser recht groß werden.

1.60 Schlüsseleditor

Der GnuPG-Schlüsseleditor ist ein interaktives Kommandozeilen-Interface zum Bearbeiten und Anzeigen der Schlüssel. Es wird mit der Option `--edit-key` "Schlüssel-ID» aufgerufen.

1.61 Schlüssellänge

Wie bei guten symmetrischen Verschlüsselungen beruht die Sicherheit auch bei einer Public-Key Verschlüsselung ganz und gar auf dem Schlüssel. Deshalb ist die Schlüsselgröße ein Maß für die Sicherheit des Systems, doch kann man die Größe eines symmetrischen Schlüssels nicht mit der eines Schlüssels einer Public-Key-Verschlüsselung vergleichen, um Rückschlüsse auf ihre relative Sicherheit ziehen zu können.

1.62 Schlüsselpaar

Bei ► [Public-Key](#) Verfahren: der ► [geheime](#) und der dazugehörige ► [öffentliche](#) Schlüssel.

1.63 Schlüssel-Server

Siehe: ► [Keyserver](#)

1.64 SHA1

Secure Hash Algorithm One. Von der ► [NSA](#) entwickelter ► [Hashalgorithmus](#) mit einer ► [Schlüssellänge](#) von 160 Bit.

1.65 Selbstunterzeichnung

(Engl. self signature)

Siehe auch: ► [Eigenbeglaubigung](#).

1.66 Signatur

Auch digitale Signatur. Aus der zu signierenden Datei und dem ► [Geheimsschlüssel](#) wird mittels eines ► [Einweg-Hashalgorithmus](#) eine digitale Signatur erzeugt, deren Echtheit man mit dem ► [öffentlichen](#) Schlüssel überprüfen kann. Wird die Datei oder die Signatur verändert, ergibt sich bei der Überprüfung der Signatur eine Fehlermeldung. Mit digitalen Signaturen kann man die Echtheit von digitalen Dokumenten wie beispielsweise Texten, Fotografien, Quellcode bestätigen.

1.67 Sitzungsschlüssel

Der ► [symmetrische](#) Schlüssel mit dem bei ► [OpenPGP](#)-Verfahren die eigentlichen Daten verschlüsselt werden. Der Sitzungsschlüssel selbst wird dann mit dem ► [asymmetrischen](#) ► [öffentlichen](#) Schlüssel verschlüsselt. Auf diese Weise kann der Sitzungsschlüssel sicher übertragen werden.

1.68 Symmetrische Verschlüsselung

Symmetrisch verschlüsselte Daten müssen mit dem gleichen Schlüssel entschlüsselt werden, mit dem sie auch verschlüsselt worden sind. Das heisst, Absender und Empfänger müssen sich auf einen Schlüssel einigen bzw. den Schlüssel miteinander tauschen. Das Sicherheitsrisiko bei symmetrischen Verfahren ist die Überbringung des Schlüssels an den Empfänger. Dieses Problem wird bei ► [Public-Key](#) Verfahren durch die Kombination mit ► [asymmetrischen](#) Verfahren gelöst.

1.69 Trust-Datenbank

Datei, in der die ► [Vertrauensstufen](#), die man den verschiedenen Schlüsselbesitzern zuordnet, verwaltet werden.

1.70 Twofish

Von  [Bruce Schneier](#) entwickelter, ► [symmetrischer](#) ► [Verschlüsselungsalgorithmus](#) mit wahlweise 128 oder 256 Bit ► [Schlüssellänge](#). Der ► [OpenPGP](#)-Standard spezifiziert 256 Bit.

1.71 Unix

Familie von Multi-User-Multi-Tasking-Multi-Platform-Betriebssystemen. Während Unix früher ausschliesslich auf mittelgroßen und großen Rechenanlagen eingesetzt wurde, gewinnt es heute in Form von ► [LINUX](#) einen wachsenden Markt im Internet, in der Industrie und im Privatbereich.

1.72 Verschlüsselung

Das Verändern eines Textes, Bildes bzw. allgemein einer Datei unter Verwendung eines ► [Schlüssels](#) und nach einem festgelegten Verfahren (► [Verschlüsselungsalgorithmus](#)), mit dem Ziel, dessen Inhalte für andere unkenntlich zu machen, wobei der Vorgang unter Verwendung des Schlüssels wieder umkehrbar ist.

1.73 Verschlüsselungsalgorithmus

Methode nach der aus dem ► [Klartext](#) der ► [Geheimtext](#) erzeugt wird. Es wird unterschieden zwischen ► [symmetrischen](#) und ► [asymmetrischen](#) Verschlüsselungsalgorithmen. Beispiele für

Verschlüsselungsalgorithmen: ► [3DES](#), ► [Blowfish](#), ► [ElGamal](#) und ► [Twofish](#).

Siehe auch: ► [Algorithmus](#).

1.74 Vertrauensstufen

(Engl. trustlevel) Maß für das Vertrauen in die Fähigkeit des Schlüsselbesitzers Schlüssel sorgfältig zu authentifizieren und zu ► [signieren](#). Die vier Vertrauensstufen werden folgendermaßen abgekürzt:

q	Unbekannt
n	kein Vertrauen
m	teilweises Vertrauen
f	volles Vertrauen

GnuPG entscheidet ausgehend von dieser Einstufung, ob es von anderen Kommunikationspartnern ► [signierte](#) und authentifizierte Schlüssel als echt anerkennt. Für ein sicheres ► [Web of Trust](#) ist es entscheidend, dass man den einzelnen Benutzer-IDs in seinem Schlüsselbund wohlüberlegte Vertrauensstufen zuweist.

1.75 Web of Trust

("Netzwerk gegenseitigen Vertrauens") Schlüsselunterschriften werden auch in einem als Web of Trust bekannten Schema benutzt, um die Gültigkeit auch auf Schlüssel auszudehnen, die nicht direkt von Ihnen selbst, sondern von anderen Personen ► [signiert](#) worden sind. Dabei ist nicht das Vertrauen in die andere Person, sondern das Vertrauen in deren Fähigkeit, Schlüssel sorgfältig zu authentifizieren und richtig zu ► [signieren](#) entscheidend. Verantwortungsbewußte Benutzer, die eine gute Schlüsselverwaltung praktizieren, können das Verfälschen des Schlüssels als einen praktischen Angriff auf sichere Kommunikation mit Hilfe von GnuPG abwehren.

1.76 Widerrufsurkunde

Wenn ein ► [geheimer](#) Schlüssel ► [kompromittiert](#) worden ist, sollte man - um Schäden und Missbrauch zu vermeiden - die davon abgeleiteten öffentlichen Schlüssel für ungültig erklären. Dies geschieht durch das veröffentlichen einer aus dem geheimen Schlüssel erzeugten Widerrufs-Urkunde.