1.1 Kryptografie

1.1.1 Symmetrische Verschlüsselung

Bei einer **symmetrischen** Verschlüsselung benutzen beide Kommunikationspartner den **identischen** (privaten) **Schlüssel** und einen allgemein bekannten (**öffentlichen**) **Algorithmus** zur Verschlüsselung der Daten. Der Namenszusatz „symmetrisch" ergibt sich aus dem **identischen (symmetrischen) Schlüssel** für beiden Richtungen.

Name	Bedeutung	Block-größe	Schlüssellänge	Sicherheit[1]	Bemerkungen
DES	Data Encryption Standard	64 Bit	56 Bit Schlüssel + 8 Bit Parität	Unsicher	Innerhalb von 22 Std. knackbar
3DES	Triple DES	64 Bit	112 Bit o. 168 Bit	Sicher	Zeitaufwendig, da dreifach DES
IDEA	International Data Encryption Algorithm	64 Bit	128 Bit	Sicher	Ähnlich 3DES, aber schneller; Patent bei Ascom Systec AG; daher selten eingesetzt
RC4/RC5 /RC6	„Ron Rivest"-Code	Variable	Variable	Sicher	Beliebige Schleifenanzahl; gilt ab 6 (besser 12) Schleifendurchläufen als sicher
Blowfish	-	-	Variable bis zu 448 Bit	Sicher	Schneller als DES; keine Patente
Twofish	-	128 Bit	Bis zu 256 Bit	Sicher	In AES Endausscheidung
AES	Advanced Encryption Standard[2]	128, 192, 256 Bit	128, 192, 256 Bit	Sicher	Ablösung des DES; Algorithmus: Rijndael[3]; keine Patente; häufig in Ipsec eingesetzt

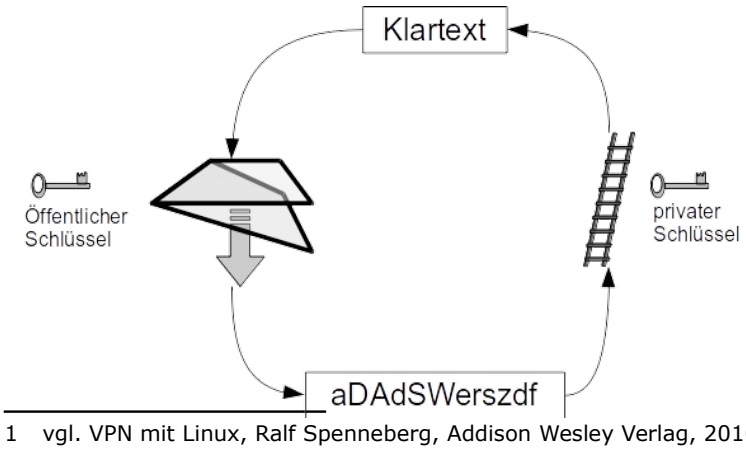
Tabelle 1: Symmetrische Verschlüsselungsverfahren

Weitere Verfahren MARS, CAST, CAST-128, CAST-256, CBC (Cipher Block Chaining).

1.1.2 Asymmetrische Verschlüsselung

Bei asymmetrischen Verschlüsselungsverfahren werden **zwei unterschiedliche Schlüssel** verwendet:

- public Key Verschlüsselung
- private Key Entschlüsselung



Als Grundlage für ein asymmetrisches Verschlüsselungsverfahren dient ein schwieriges mathematisches Problem, welches in eine Richtung leicht berechnet werden kann, aber die Rückrichtung „unmöglich"[4] zu ermitteln ist. Für den Einsatz eines solchen mathematischen Problems muss also eine sogenannte „Falltür"-Funktion[5] möglich sein. Beispiele für asymmetrische Verschlüsselungsverfahren sind das RSA[6]- und DSA[7]-Verfahren.

1 vgl. VPN mit Linux, Ralf Spenneberg, Addison Wesley Verlag, 2010, 2. Auflage
2 Wettbewerb des NIST (National Institute of Standards and Technology) zur Festlegung eines neues Verschlüsselungsstandards
3 Rijndael: von den Belgiern Joan Daemen und Vincent Rijmen
4 Als „unmöglich" gilt eine Rechenzeit von mehreren Tausend bis Millionen Jahren.
5 Man kann zwar alles durch die Falltür (öffentlicher Schlüssel u. Algorithmus) werfen, aber nur über die richtige Treppe (privater Schlüssel) gelangt man wieder an den ursprünglichen Ort (Klartext).
6 RSA: benannt nach **R**ivest, **S**hamir und **A**dleman; kann zur Verschlüsselung sowie für digitale Signaturen verwendet werden; asymmetrisches Verfahren
7 DSA: **d**igital **S**ignature **A**lgorithm; wird für digitale Signaturen verwendet; asymmetrisches Verfahren

Das folgende Diagramm zeigt den prinzipiellen Ablauf beim Einsatz eines asymmetrischen Verschlüsselungsverfahren. Es wird davon ausgegangen, dass A etwas Geheimes an B versenden möchte. Dazu muss B einen **private** und einen **public Key generieren**. Den public Key kann er über einen unverschlüsselten Kanal versenden. A nutzt den public Key von B um seine Nachricht zu verschlüsseln. Mittels des private Keys von B kann dieser die verschlüsselte Nachricht entziffern.

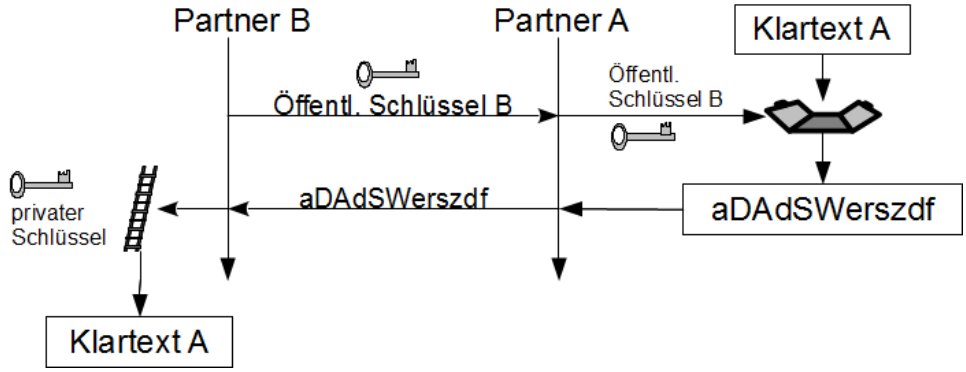


Abbildung 1: prinzipieller Ablauf asymmetrisches Verschlüsselungsverfahren

1.1.2.1 Diffie Hellmann (DH)-Verfahren

Das DH-Verfahren sorgt dafür, dass beide Kommunikationspartner über die korrekten private bzw. public Keys verfügen. **WICHTIG**: Die **vorherige Authentifizierung** ist beim Einsatz des DH-Verfahren von größter Bedeutung, da ansonsten ein Man-in-the-Middle die Schlüsselgenerierung vortäuschen könnte.

Beide Kommunikationspartner (Amy und Bernd) wählen eine sehr große **Primzahl p** und eine zusätzliche **Zufallszahl z**. Weiterhin wählen beide je eine **persönliche** und **geheime Zufallszahl** (**a** bzw. **b**).

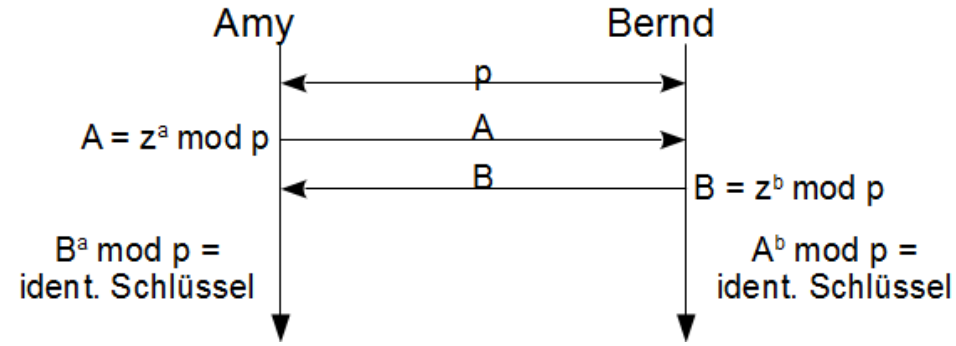


Abbildung 2: Schlüsselaustausch nach Diffie-Hellmann

Mathematisch gilt: $B^a \bmod p = (z^b)^a \bmod p = z^{(ab)} \bmod p = (z^a)^b \bmod p = A^b \bmod p$

Das heißt nach dem Austausch von A und B verfügen beide über den identischen Schlüssel. Im IKE sind p (große Primzahl) und z (Generator) fix. Beides ist standardisiert und muss nicht übertragen werden. Amy und Bernd erzeugen nur noch MODP-Gruppen[8] mit festgelegten Längen (768, 1024, usw.)

1.1.2.2 Hash

Ein Hash-Wert ist eine zufällige Zahl (Ausgabewert), die per Algorithmus[9] aus einer Eingabe (abzusichernder Wert) ermittelt wird. Eine beliebig lange Eingabe führt zu einem über den möglichen Wertebereich gleichverteilten Ausgabewert mit fester Länge. Ein Rückschluss auf die Eingabe ist so nicht mehr möglich. Anforderungen an einen Hash-Algorithmus sind:

- Schnelligkeit
- Umkehrung muss unmöglich sein
- zwei ähnliche Eingaben müssen zu vollkommen verschiedenen Ausgaben führen

Hash-Werte werden zu **Sicherstellung der Integrität bei IPsec** eingesetzt. MD5 und SHA-1 bzw.

8 MODP: **mod**ulo **p**
9 z.B.: CRC16, MD5, SHA

SHA-2 sind die hierfür verwendeten Algorithmen. MD5 ist nicht kollisionsfrei[10]. 2004/2008 wurden Methoden veröffentlicht, mit denen Kollisionen berechnet werden konnten (das X.509 Zertifikatssystem gilt daher nicht mehr als sicher).Mittelfristig sollten VPNs daher nicht mehr mit MD5 betrieben werden.

SHA[11] wird bei DSA eingesetzt und kann mit 160 Bit-Hash (SHA-1) bzw. 256/384/512 Bit-Hash (SHA-2) betrieben werden. 2005 wurden erste erfolgreiche Angriffe aus SHA-1 dokumentiert. SHA-2 gilt derzeit (Stand: 2010) als sicher.

1.2 Weitere Tunnel-Protokolle

Neben IPsec gibt es eine Reihe weiterer Protokolle mit deren Hilfe sich gesicherte Verbindungen zu entfernten Systemen aufbauen lassen. Im folgenden werden diese kurz vorgestellt.

SSL[12] wurde ursprünglich zur Absicherung von HTTP-Verbindungen erdacht und kann **nur TCP-Verbindungen gesichert** übertragen. SSH[13] kann ASCII-Verbindungen aufbauen und darüber einzelne TCP-Verbindungen tunneln. Es stellt daher noch kein vollwertiges VPN dar.

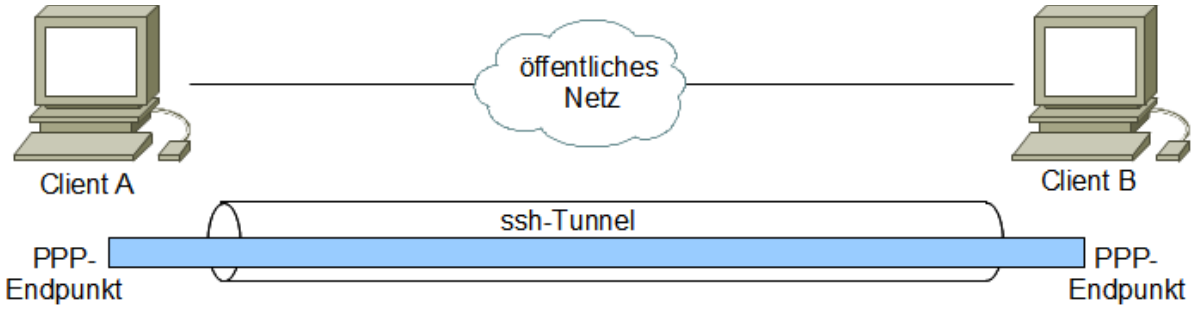


Abbildung 3: "PPP über ssh-Tunnel"-Verbindung

In Mircosoft-Systemen wird **PPTP**[14] ähnlich wie SSH/PPP eingesetzt. Hierzu wird zunächst ein verschlüsselnder GRE-Tunnel[15] aufgebaut und darin die PPTP Übertragung vorgenommen. PPTP übernimmt dabei selbst weder Verschlüsselung noch Authentifizierung, weshalb PPTP als unsicher gilt.

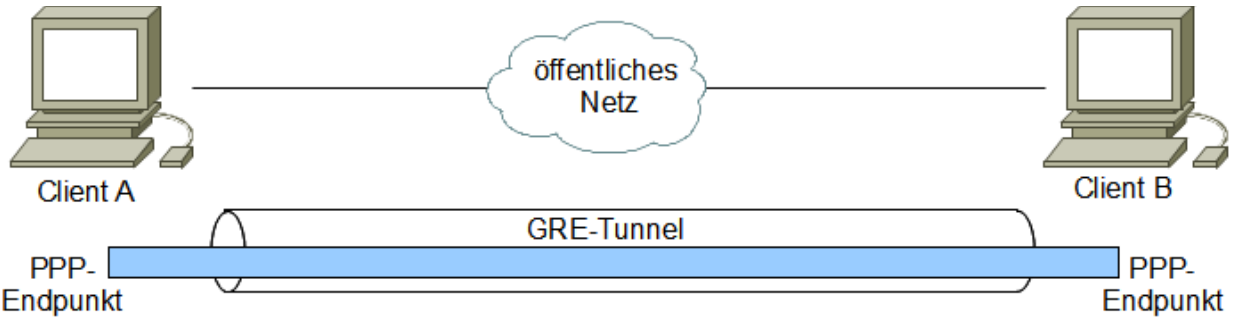


Abbildung 4: PPTP-Verbindung (PPP über GRE-Tunnel)

Mit Einführung von **MS-CHAPv2**[16] wurde dieses System um die **Vertraulichkeit** erweitert und **MPPE**[17] benannt. Die Verschlüsselung wird mit 40 bis 128 Bit Schlüsseln, die aus dem Kennwort generiert werden, vorgenommen. Genau hierin liegt auch die Schwachstelle des Systems. Es ist einfacher ein Kennwort zu erraten als eine Brut-Force-Attacke durchzuführen, da Kennwörter normalerweise nicht echt zufällig sind. Daher gelten PPTP und MPPE als grundsätzlich unsicher.

Quelle für dieses Arbeitsblatt: R. Spenneberg, VPN mit Linux

10 zwei Eingaben können zu identischer Ausgabe führen und ermöglichen so Manipulationenen
11 SHA: **S**ecure **H**ash **A**lgorithm
12 SSL: **S**ecure **S**ocket **L**ayer
13 SSH: **S**ecure **SH**ell
14 PPTP: **P**oint to **P**oint **T**unneling **P**rotocol
15 GRE: **G**eneric **R**outing **E**ncapsulation
16 MS-CHAPv2: **M**icro**S**oft **Ch**allenge/Response **A**uthentication **P**rotocol **V**ersion **2**
17 MPPE: **M**icrosoft **P**oint to **P**oint **E**ncryption Protocol