

# Vom Schein des Unscheinbaren

Till Hoke | till.hoke@cms.hu-berlin.de

## Einleitung

Es gibt Begriffe, die in ihren verschiedenen Anwendungsbereichen Gegensätzliches bedeuten. Ein solcher Begriff ist der der Transparenz, des Durchscheinens. Im Felde des Politischen etwa bezeichnet er eine öffentliche Durchschaubarkeit der Verhältnisse oder Entscheidungen hinsichtlich ihrer Bedingungen und Hintergründe.

In arbeitsteiligen Umgebungen geht für den Einzelnen in den Poren des Gesamtzusammenhanges der Blick für die Details des Ganzen oft verloren, ja er ist sogar nicht erwünscht. Hier bedeutet das Durchscheinens Unsichtbarkeit, das Absehen von den Einzelheiten des Zusammenhanges, der Mechanik, die in seinen Tiefen verborgen liegt. Diese soll funktionieren und in ihrem Funktionieren unbemerkt im Hintergrund bleiben. Erst aber, wenn das Funktionsgefüge Risse aufweist, wenn der Drucker fehlt, wenn die Anwendung über das Netzwerk nicht startet oder das Passwort plötzlich nicht mehr greift, fällt der Hintergrund aus seinem transparenten Dabei lärmend in den Vordergrund des Arbeitsalltags ein. Dann scheint im Durchscheinens nicht mehr die Durchsichtigkeit, sondern das Undurchdringliche eines Anderen: zum Beispiel das eines Virtual Private Network (VPN).

Sicherheitsanwendungen wie das VPN bieten überhaupt einen ausgezeichneten Zugang zum Konzept der Transparenz, denn gerade die Sicherungen sollen funktionieren, auch ohne dass man diese ständig bedenken und durchschauen muss. Denn was erst noch und immerzu auch bedacht werden soll, kann dem Unbedachtsein verfallen.

## Praktische Hintergründe

### Von den sieben Geißlein, des Wolfes Kreide und dem VPN

Der Zugriff auf vertrauliche Daten bedarf der Vorsicht und gewisser Zurüstungen vor allem dann, wenn er über ein Medium erfolgt, welches man nicht kontrolliert und über dessen Vertraulichkeit man keine Aussage treffen kann. Darf man dem Übertragungsweg aber nicht vertrauen, muss die Vertraulichkeit in das Übertragene selbst verlegt werden. Das heißt, an den Endpunkten der Übertragung muss es Vorrichtungen geben, die das Übertragene dahingehend verpacken, dass es für Dritte auf dem Wege der Übertragung nicht ausgespäht werden kann (*Vertraulichkeit – Confidentiality*). Wahrscheinlich wird auf einem der Endpunkte irgendein geheimer Schlüssel nach einer kryptographischen Funktion auf das zu übertragene Datum angewendet, welches dadurch für Dritte unleserlich wird. Das andere Ende entfernt die ganze Geheimwissenschaft wieder von dem aus der Ferne Erhaltenen und gibt diesem seine Lesbarkeit zurück.

Allein der Vorsicht ist damit noch nicht Genüge getan. Es mag nämlich sein, dass der eine Endpunkt der Übertragung gar nicht der ist, der zu sein er vorgibt, wie etwa der böse Wolf den sieben Geißlein weismachen will, dass es die Mutter sei, welche Einlass begehrt. Also verlangen wir einen Ausweis, den die Daten bei sich führen, und den allein die Daten unseres erwarteten und vertrauten Gegenübers bei sich führen können, weil dieser Ausweis nicht wie die

*Seit etwa 10 Jahren existiert eine VPN-Lösung, die ihren Ursprung in einem vom DFN geförderten Drittmittel-Projekt hatte. Inzwischen nutzen ca. 70 Mitarbeiter in den Bereichen Prüfungsverwaltung und Haushalt das damals entwickelte Verfahren. Dieses befindet sich gegenwärtig sowohl, was die verwendeten Technologien anbetrifft, als auch in Rücksicht auf die eingesetzte Technik in einer weitgehenden Umgestaltung, um es den kommenden Anforderungen gegenüber aufzuschließen. Der Artikel gibt einen Überblick über das neue Verfahren auf Basis von OpenVPN und dessen technische Voraussetzungen.*

Kreide<sup>1</sup> des Wolfes in irgendeinem Kaufmannsladen für jeden Dahergelaufenen feilgeboten wird (Authentifizierung – Authentication).

Ferner mag es sein, dass üble Zeitgenossen unseren Daten auf dem Wege über das Medium auflauern und, da sie diese nicht zu lesen vermögen, ihnen Schaden antun, bevor sie sie auf die Weiterreise schicken. Wir wollen also unseren wertvollen Vertraulichkeiten ein Merkmal mitgeben, an dem wir ablesen, ob diese unterwegs aufs Geratewohl manipuliert worden sind (Integrität – Integrity).

Zu guter Letzt soll unser Gegenüber, dem wir vertrauten, zu keiner Zeit abstreiten können, dass das Gesendete von ihm selbst stammt (Unleugbarkeit – Non-repudiation).

Es bleibt in Rücksicht auf das Thema zu ergänzen, dass wir diese vier Anwendungen nicht jedes Mal extra und auch noch persönlich an unserer vertraulichen Post anbringen möchten. Wir mögen das getrost Agenten überlassen, denen wir vertrauen, wie etwa einem Rechtsanwalt<sup>2</sup>. Und eigentlich wollen wir auch gar nicht entscheiden, welches unserer Datenpakete vertraulich ist (und damit der Vorsorge bedarf) und welches nicht. Und wir sollten das auch nicht dürfen. Unsere Agenten sollen sicherstellen, dass nur Daten, denen unsere Vorsicht in den oben genannten Weisen obwaltete, unser sicheres Heim verlassen. Denn was hätten wir gewonnen, wenn wir die geheime Schatzkarte nach all dem Aufwand in unsere gute Stube geborgen hätten und ein unbemerkter Gast sich des Nachts an unseren Schreibtisch schliche und mit einer Kopie der Karte durch das ungesicherte Kellerfenster verschwände. Also überlassen wir den Agenten gleich noch die Verantwortung für unsere Türschlösser – oder allgemeiner, die Sorge für die Sicherheit der Endpunkte unseres vertraulichen Datenverkehrs. Selbstverständlich sollen die Agenten all das im Sinne der Einleitung unsichtbar und im Hintergrund erledigen.

<sup>1</sup> Für diejenigen, die mit den Märchen der Gebrüder Grimm nicht vertraut sind, gibt es hier Hilfe: [http://de.wikipedia.org/wiki/Der\\_Wolf\\_und\\_die\\_sieben\\_jungen\\_Geißlein](http://de.wikipedia.org/wiki/Der_Wolf_und_die_sieben_jungen_Geißlein)

<sup>2</sup> Aber auch für den Agenten besteht die Gefahr der Korruption. Siehe [5].

Und während wir den sieben Geißlein nachhängend unsere Kellerfenster versperren, haben wir insgeheim schon mal abgehandelt, was ein VPN tun soll.

Ersetzen wir nun in dem Bild das sichere Heim durch einen PC, die Agenten durch eine Software oder ein Netzwerkgerät und die vertrauliche Fracht durch den Datenverkehr zwischen einer Datenbank und einem Anwendungsprogramm, so haben wir zu den besprochenen Zurüstungen, welche die Vorsicht gebot, auch schon ein lebendiges Szenario, wie es sich im universitären Alltag abspielt.

## Von den Anwendungen und den Spielarten des VPN

An der Humboldt-Universität ergeben sich die VPN-Anwendungsfälle für das Verwaltungsnetz mehrheitlich aus zwei Entwicklungen:

- 1. Zentrale Datenhaltung/dezentrale Verarbeitung:** Dies ist das klassische VPN-Problem. Die dezentrale Organisation auf der einen und die gemeinsame Nutzung von Ressourcen auf der anderen Seite erfordern einen sicheren Zugriff auf Datenbestände, welche zentral vorgehalten werden müssen. Dabei handelt es sich um die großen Datenbanken der Universitätsverwaltung, insbesondere die Haushalts- und die Studierendendatenbank.
- 2. Administration:** Die bedingt durch eine steigende Zahl von Online-Anwendungen wachsende Komplexität der Serverinfrastruktur sowie die nahezu ständige Verfügbarkeit dieser Anwendungen verlangen eine flexible Administration, d. h. die Administration der Server muss zu jedem Zeitpunkt und von jedem Ort aus abgesichert möglich sein.

Im Einzelnen lassen sich die VPN-Anwendungen derzeit nach folgenden Arbeitsgebieten unterscheiden:

- der Zugriff auf die Haushaltsdatenbank aus den Fakultätsverwaltungen;
- der Zugriff auf die Studierendendatenbank aus den Prüfungsämtern;
- der Zugriff der Administratoren von außerhalb des Verwaltungsnetzes auf

ihre Administrationskonsolen;

- die Heimarbeitsplätze mit Zugang zum Verwaltungsnetz.

Die beiden erstgenannten Dienste werden schon seit mehreren Jahren angeboten. Die Versorgung der Prüfungsämter ist dabei so gut wie abgeschlossen, die Zahl der Anwender wird sich also nicht wesentlich erhöhen. Im Bereich Haushalt erwarten wir allerdings noch steigende Nutzerzahlen. Beide Fallgruppen unterscheiden sich jeweils noch einmal danach, ob sie noch am älteren IPsec-VPN teilnehmen oder schon in das OpenVPN-Netz eingebunden sind.

| Anwendung             | IPsec | OpenVPN |
|-----------------------|-------|---------|
| Haushaltsdatenbank    | 7     | 11      |
| Studierendendatenbank | 30    | 15      |

Tabelle 1: Anzahl der Arbeitsplätze bei den Hauptanwendungen im VPN des Verwaltungsnetzes

Die zuletzt genannten Dienste (administrativer Zugriff und Heimarbeitsplätze) befinden sich noch in einem Pilotstadium.

Gegenwärtig werden zwei VPN-Technologien parallel eingesetzt: IPsec (implementiert von F-Secure VPN+) und OpenVPN. Mit IPsec begann die VPN-Entwicklung im Verwaltungsnetz. Inzwischen wurde jedoch die verwendete kommerzielle Software durch den Hersteller verkauft. Die Nachfolgefirma entwickelte das Produkt nicht mehr weiter – zumindest nicht als eigenständige Anwendung. Das ist zwar gerade im Sicherheitsbereich, wo die Aktualität der Software enorm wichtig ist, schon ein genügender Anlass, das Produkt zu wechseln. Allein, es waren für uns andere Kriterien ausschlaggebend, denn die verwendeten Sicherheitstechniken und -parameter (Hashfunktionen, Verschlüsselungsalgorithmen, Schlüssellängen) sind durchaus noch aktuell (und unterscheiden sich wenig von denen, welche OpenVPN einsetzt). Für alle anfallenden Probleme mit der Software selbst (die in einigen Bereichen durchaus noch nicht ausgereift war) waren wir aber auf uns allein gestellt. So bereitet die Installation von F-Secure VPN+ auf

Windows-Systemen in der Nachfolge von Windows 2000 mitunter Probleme, z. B. im Zusammenhang mit Hyperthreading unter Windows XP. Unter Windows Server 2003 ist die Installation von VPN+ eine Hürde, die man nur mit Tricks überwinden kann, wobei keine Versuche mehr unternommen wurden, solche auf Biegen und Brechen zustande gekommene Installation auch noch einem Stabilitätstest zu unterziehen. Irgendwann heißt es halt Abschied nehmen.

OpenVPN wird IPsec beim Zugang zum Verwaltungsnetz ersetzen. Damit wird aber nicht nur die Software, sondern die komplette Infrastruktur ausgetauscht. Die folgende Tabelle liefert eine Übersicht über die beteiligten Komponenten.

Die Abteilung „DV in der Verwaltung“ des CMS der Humboldt-Universität stellt für die Arbeitsplätze folgende Dinge bereit:

- den Thin Client;
- einen 2-fach KVM-Umschalter mit DVI-Ports, USB-Konsole und USB-HUB;

- den Linksys VPN-Router (1 Uplink, 4 LAN-Ports);
- bei Bedarf: Adapter PS/2-Tastatur/Maus auf USB;
- bei Bedarf: SSL-fähige Printboxen (inklusive Adapter Centronix auf USB);
- bei Bedarf: Ethernet-Switch.

Auf der Fakultätsseite werden benötigt:

- eine freie IP-Adresse für die VPN-Box;
  - eine Dose, die in das lokale Netz gepatcht ist;
  - einen Netzwerkdrucker mit Linux-Unterstützung<sup>4</sup>;
  - einen Monitor mit DVI-Eingang.
- Wünschenswert sind überdies USB-Tastatur und -Maus, da es beim Adapterbetrieb eher zu Problemen<sup>5</sup> kommt.

Leider benötigen die drei Standardgeräte auf der Client-Seite der OpenVPN-Hardware (Thin Client, VPN-Box, KVM-Switch) mehr Ressourcen, als beim Einsatz von IPsec notwendig waren. Kommen dann noch eine Printbox und ein Ethernet-Switch hinzu, erhöht sich der Bedarf in ganz unterschiedlichen Bereichen. Das betrifft zunächst einmal die IP-Adressen, aber auch so banale Dinge wie Steckdosen für die Stromversorgung. Oft kommen Platzprobleme hinzu, denn auch wenn ein leichtgewichtiger Thin Client einen mitunter sperrigeren PC ersetzt, müssen Kabel und Netzteile untergebracht werden. Wo bereits VPN-Clients auf IPsec-Basis bestehen, sind selbstverständlich keine neuen IP-Adressen bzw. Dosen erforderlich. Dadurch, dass an eine VPN-Box bis zu vier Thin Clients angeschlossen werden können, reduziert sich der Bedarf an IP-Adressen dort, wo mehrere Sachbearbeiter in einem Raum sitzen, denn die Thin Clients werden über den bestehenden VPN-Tunnel virtuell mit dem geschützten Netzwerkbereich verbunden und beziehen aus diesem ihre IP-Adressen.

<sup>4</sup> Gemeint ist, dass der Hersteller nicht auf proprietäre Standards setzt, sondern gängige Druckersprachen wie PCL oder Postscript bzw. für die Netzwerkübertragung IPP oder Application Socket unterstützt.

<sup>5</sup> Diese Probleme führen an einigen Arbeitsplätzen bisweilen zum Komplettausfall der Eingabegeräte, die nur durch geduldiges Ab-, An- und Umstecken wieder zur Zusammenarbeit zu bewegen sind. Die Hintergründe dieser Blackouts liegen ebenso im Dunkeln, wie sich die Wiederbelebung jeder Systematik entzieht.

### IPsec und OpenVPN – ein historischer Abriss

Vor dem Aufkommen des Internets mussten Firmen hohe Geldsummen investieren, um ihre Zweigstellen mit der Zentrale über dedizierte Kommunikationsleitungen zu verbinden. Mit dem Aufstieg des Internets Anfang der 90er Jahre gab es in wachsendem Maße Bandbreite für den Datenverkehr. Diese war zwar kostengünstiger, das Medium aber öffentlich und damit unsicher. Es entstand das Problem, dedizierte, vertrauliche Verbindungen über dieses öffentliche Medium zu realisieren.

Das erste größere Projekt dazu war IPsec – ein Ableger der IPv6-Arbeitsgruppe der Internet Engineering Task Force (IETF) –, dessen erste Version 1995 zur Verfügung stand. Die für die Kryptofunktionen noch zu geringe Prozessorleistung in den Routern, wo sie implementiert werden sollten, behinderte allerdings zunächst die Verbreitung. Dazu kam, dass einige der Komponenten von IPsec, wie z. B. das IKE-Protokoll, eine lange Entwicklungszeit beanspruchten. Schließlich erwies sich IPsec als ein recht komplexes Bündel von Standards, welches vom Anwender gerade in der frühen Phase relativ hohen Lernaufwand erforderte.

Komplexität und langsamer Fortschritt führten Mitte der 90er Jahre zu einer Zersplitterung der Bestrebungen. Ein wichtiger Ableger ist das Secure Socket Layer-Protokoll (SSL, 1994 von Netscape entwickelt). Der Fokus der Entwicklung von SSL lag dabei in der Sicherheit auf Anwendungsebene, nicht auf Netzwerkebene wie bei IPsec. Aufgrund des weitläufigen Gebrauchs im World Wide Web entwickelte sich SSL im Gegensatz zu IPsec recht schnell. Allerdings waren sogenannte SSL-VPNs nur geschützte Client-Server-Verbindungen einzelner (Web-)Applikationen, keine VPNs im eigentlichen Sinne<sup>3</sup>.

Ende der 90er Jahre bot die Reifung der Linux-Systeme ausgezeichnete Bedingungen für den Test experimenteller Netzwerkkonzepte. Eine dieser Neuerungen war das tun/tap-Device (1999/2000). Tun und tap simulieren Netzwerkgeräte über Software. Hinter diesen Geräten steht keine Hardware, sondern ein User-Space-Programm. Das ermöglicht eine spezielle Bearbeitung von Netzwerkpaketen, ohne das Netzwerksubsystem des Betriebssystemkerns zu manipulieren (wie das bei IPsec der Fall ist).

Die kryptographischen Entwicklungen, welche in die OpenSSL-Bibliothek eingegangen sind, einerseits und virtuelle Netzwerkgeräte andererseits bildeten die Kernkomponenten der Entwicklung von OpenVPN durch James Yonan (erstes Release 2001). (nach [2], Folie 3 – 7)

Kasten 1: Historisches (nach [2], Folie 3 – 7)

<sup>3</sup> VPNs im eigentlichen Sinne bestehen nur in der Kombination von Verbindungsvertraulichkeit und Netzwerksicherheit.

|                        | IPsec-basiertes VPN   | OpenVPN   |
|------------------------|---|---|
| Rechner-Hardware       | dedizierter PC mit KVM-Switch   | dedizierter Thin Client mit KVM-Switch  |
| Rechner-Betriebssystem | Windows 2000/XP   | Windows XPe/Embedded Standard   |
| VPN-Agent              | F-Secure VPN+-Instanz auf dem PC  | OpenVPN unter OpenWrt auf Linksys-Router  |
| VPN-Gateway            | Windows 2000 Server mit F-Secure VPN+   | Debian Linux mit OpenVPN  |
| Datenbankzugriff über  | Frontend auf Citrix Metaframe Terminalserver  | Frontend auf Citrix Metaframe Terminalserver  |
| Proxy                  | Socks   | Socks   |
| Drucker                | lokal oder im Netz: die Treiber müssen auf dem PC und dem Terminalserver installiert sein | ausschließlich im Netz via CUPS-Server: die Treiber müssen auf dem CUPS-Server installiert werden, auf den Terminalservern werden die Drucker jeweils mit demselben allgemeinen Treiber eingebunden |
| sonstige Hilfsdienste  | DNS, NTP, CA, Policy Manager (Konfigurationsservice für die VPN-Policies)                 | DHCP, DNS, NTP, CA, CURC (Konfigurationsservice für OpenWrt), WDM (Konfigurationsservice für Wyse Thin Clients)   |

Tabelle 2: Unterschiede derzeitiger VPNs im Verwaltungsnetz

## Technische Hintergründe

Jede ernsthafte VPN-Technologie muss zumindest zwei Bereiche berücksichtigen. Der eine ist augenscheinlich der Bereich der Kryptographie, der Methoden des Verbergens und des Offenlegens. Der andere Bereich ist das minder offenbare Feld der Netzwerkinfrastruktur. Beide Bereiche greifen selbstverständlich ineinander. So bedient sich der kryptographische Schlüsselaustausch<sup>6</sup> gewisser Kommunikationsstandards, wie zum Beispiel des IKE-Protokolls für IPsec oder des TLS<sup>7</sup>-Handshakes für OpenVPN.

### Von den Geheimnissen ...

Sowohl IPsec als auch OpenVPN nutzen dieselben kryptographischen Verfahren. Diese wurden in den 70er Jahren des vorigen Jahrhunderts entwickelt und beruhen auf dem mathematischen Sachverhalt, dass sich Gleichungen in einer Richtung sehr schnell lösen lassen, in umgekehrter Richtung aber – ohne Zusatzinformation – nur mit sehr großen Aufwand. Es geht beim Ver- und Entbergen also um Komplexität, um den Res-

ourcenverbrauch des optimalen Algorithmus' zur Lösung eines mathematischen Problems. Für spezielle mathematische Probleme gibt es (noch) keinen effizienten Algorithmus, so dass die Auflösung bei genügend großen Zahlen ähnlich komplex ist wie das Durchprobieren aller möglichen Lösungen.

Die Sicherheit nach solchen Problemen formulierter Kryptoverfahren hängt also an zwei Bedingungen:

- erstens am mathematischen Fortschritt, denn eine wissenschaftliche Leistung, die zweifelsohne den Nobelpreis verdient hätte (wenn es denn für Mathematik einen gäbe), könnte einen effizienteren, weniger aufwändigen Algorithmus formulieren;
- zweitens an der technischen Entwicklung neuartiger, leistungsfähiger Computer, welche komplexe Aufgaben in Bruchteilen der auf Basis der gegenwärtigen Prinzipien der Rechentechnik benötigten Zeit erledigen.

Sicherheit ist somit relativ oder historisch. Aber auch auf der Gegenseite gibt es ökonomische Rücksichten. So mag wohl kaum jemand für riesige Geldsummen Rechner und Personal aufreiben, um über einen großen Zeitraum nur sehr wenig Text zu entschlüsseln. Auch beim Spionieren spielen Zeit und Geld natürlich eine Rolle. Schließlich haben Informationen eine recht niedrige Halbwertszeit. Wenn wir also unterstellen, dass es keine absolute Sicherheit

gibt, so darf man ergänzen, dass diese auch nicht nötig ist, weil in der Welt nun einmal alles in der Zeit und in Verhältnissen geschieht. Auch das Übel wirkt nur in der Abhängigkeit.

Die eingesetzten Kryptoverfahren sind dennoch auf dem höchsten verfügbaren Entwicklungsstand für solche Technologien. Sie lassen sich anhand der eingangs genannten vier Prinzipien für sichere Datenübertragung darstellen. Dabei handelt es sich um das Diffie-Hellman-Verfahren zur gemeinsamen Schlüsselberechnung, das asymmetrische RSA-Kryptosystem, Einweg-Funktionen für Hashes (Message Digests<sup>8</sup>) und symmetrische Verschlüsselungsalgorithmen. Dazu kommen dann noch Protokolle, die festlegen, wie der auf die Erzeugung und Erneuerung des Schlüsselmaterials bezogene Datenaustausch über das Internet kommuniziert werden soll.

1. Schlüsselgenerierung und symmetrische Verschlüsselung (Vertraulichkeit)  
Aus Geschwindigkeitsgründen wird der Datenverkehr über das Netz nicht nach dem Public-Key-Verfahren verschlüsselt, sondern man bedient sich schneller

<sup>6</sup> Schlüsselaustausch: Genaugenommen handelt es sich nicht um einen Schlüsselaustausch, sondern um einen Austausch von Material zur gemeinsamen, aber getrennten Schlüsselgenerierung.

<sup>7</sup> Internet Key Exchange (IKE) und Transport Layer Security (TLS) sind Protokolle, die u. a. den initialen Aufbau einer abgesicherten Netzwerkverbindung (den Handshake) regeln.

<sup>8</sup> Message Digest ist eine Zeichenfolge fester Länge, welche einen Text (nahezu) eindeutig in kondensierter Form repräsentiert. Das heißt, die Möglichkeit, dass ein zweiter Text nach Anwendung einer Hashfunktion dieselbe Repräsentation besitzt (Kollision), soll verschwindend gering sein. Das Digest wird durch die Anwendung sogenannter Einweg-Funktionen (Hashfunktionen) erzeugt.

Algorithmen, die mit symmetrischen Schlüsseln arbeiten. Standardmäßig verwendet OpenVPN den Algorithmus Blowfish CBC<sup>9</sup> mit 128 Bit Schlüssel-länge. Da die Schlüssel, mit denen die symmetrischen Verfahren arbeiten, auf beiden Seiten gleich sein müssen<sup>10</sup>, erhebt sich das Problem, wie man über unsichere Wege einen gemeinsamen geheimen Schlüssel kommuniziert. Eine Methode wäre das Verteilen von sogenannten Preshared Keys auf jenen Rechnern, welche abgesichert miteinander kommunizieren. In großen, weit verteilten Netzen kann dies sehr unbequem sein, vor allem deshalb, weil man die Schlüssel in gewissen Zeitabständen erneuern muss. Das Diffie-Hellman-Verfahren<sup>11</sup> beschreibt eine Methode, nach der zwei Systeme je für sich selbst aus öffentlich bekanntem und aus geheimen Material das gleiche Geheimnis ableiten.

## 2. Die RSA-Kryptographie<sup>12</sup> (Authentizität des Senders)

Für sich genommen weist das Diffie-Hellman-Verfahren eine Schwachstelle auf. Die besteht in der Anfälligkeit für Man-in-the-middle-Angriffe. Der Mann in der Mitte kann nämlich in den Besitz des privaten Geheimnisses und damit zum Klartext des damit verschlüsselten Materials kommen, indem er sich einfach in die zur Schlüsselgenerierung notwendige Kommunikation zwischen zwei Parteien ein-

9 Cipher Block Chaining ist ein kryptographisches Verfahren, bei dem – um Angriffe zu erschweren – zu jedem Block zu verschlüsselnden Textes Material aus dem vorangehenden, schon verschlüsselten Block über ein bitweises XOR hinzugerechnet wird. Für den ersten Block stammt das Material aus einer zufälligen, schwer vorherzusagenden Zeichenfolge, dem Initialization Vector (IV).

10 bzw. leicht auseinander ableitbar. OpenVPN verwendet solche Schlüsselpaare für das Senden/Empfangen der Signatur (des HMAC) bzw. für die Verschlüsselung/Entschlüsselung der Nachricht.

11 Das Verfahren wurde von Martin Hellman, Whitfield Diffie und Ralph Merkle an der Stanford Universität entwickelt und 1976 veröffentlicht. Weniger bekannt ist, dass bereits Anfang der 70er Jahre die britischen Wissenschaftler James Ellis, Clifford Cocks und Malcolm J. Williamson ein ähnliches Verfahren beschrieben hatten, für welches aber – aus Gründen der Geheimhaltung – nie ein Patent beantragt worden war.

12 RSA – Das Verfahren wurde 1977 von den Wissenschaftlern Ronald L. Rivest, Adi Shamir und Leonhard Adleman am Massachusetts Institute of Technology (MIT) entwickelt. Auch hier waren James Ellis, Clifford Cocks und Malcolm J. Williamson wohl diejenigen, welche schon vorher ein ähnliches System erfunden hatten.

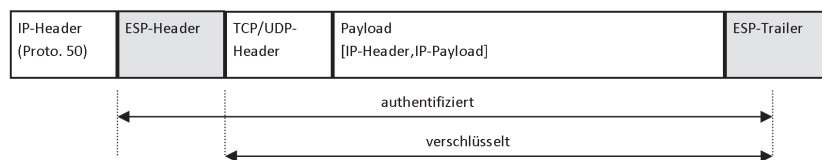


Abb. 1: Authentifizierungs- und Verschlüsselungsbereich in IPsec ESP-Paket (Tunnelmodus)

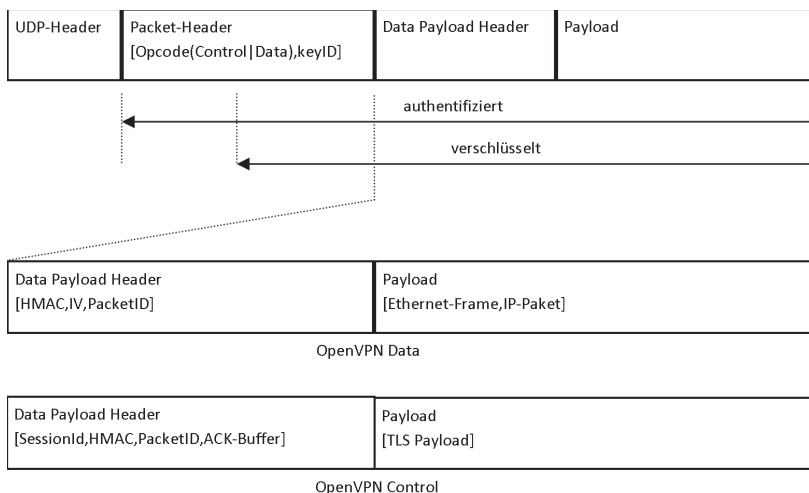


Abb. 2: Authentifizierungs- und Verschlüsselungsbereich in OpenVPNs Data- und Control Channel Datagramm (tun/tap-Modus) (nach [4], Kap. 8.5)

schaltet und mit beiden Seiten je einen Diffie-Hellman-Austausch durchführt. Es muss also sichergestellt werden, dass sich die beiden Parteien am Beginn der Kommunikation einander ausweisen. Deshalb gehört ein RSA-basiertes Public-Key-Verfahren zum initialen Ritual des Aufbaus einer IPsec- oder OpenVPN-Verbindung. Jeder der Kommunikationsteilnehmer verfügt über ein Paar nach einer mathematischen Funktion einander zugeordneter Schlüssel. Einer dieser Schlüssel – der private – bleibt beim Besitzer. Der andere Schlüssel ist öffentlich und für die Verteilung an Kommunikationspartner vorgesehen. Ihre öffentlichen Schlüssel tauschen die Kommunikationspartner während des initialen Handshakes in Form von Zertifikaten aus. Im Zertifikat, einem unterschriebenen digitalen Dokument, bestätigt eine dritte Instanz, der beide Seiten vertrauen (Certificate Authority, CA) die Zugehörigkeit eines Schlüssels zu einer bestimmten Identität. Diese Bestätigung trägt als Echtheitssiegel eine mit dem privaten Schlüssel der CA erstellte und mit deren öffentlichem Schlüssel verifizierbare elektronische Signatur. Jeder dieser asymmetrischen Schlüssel kann

nun verwendet werden, um Daten zu verschlüsseln, in der Weise, dass das Ergebnis des Anwendens des einen Schlüssels (des öffentlichen oder des privaten) jeweils den zugeordneten Schlüssel zur Entzifferung benötigt. Es ist klar, dass für vertrauliche Daten derjenige Schlüssel zum Verbergen benutzt wird, dessen zum Entschlüsseln erforderliches Gegenstück privater Natur ist, also der Public Key des Gegenübers. Umgekehrt zieht man zum Erzeugen einer digitalen Signatur seinen privaten Schlüssel heran. Dabei wird vom zu signierenden Text (z. B. einer E-Mail) mit Hilfe einer Einwegfunktion ein Message Digest erstellt und mit dem Private Key verschlüsselt. Jeder, der den Text erhält, vermag mit Hilfe des Public Keys des Absenders die Signatur zu entschlüsseln und das Digest mit dem Ergebnis zu vergleichen, das er erhält, wenn er seinerseits mittels der (öffentlich bekannten) Hashfunktion aus dem Text das Message Digest erzeugt. OpenVPN benutzt für das anfängliche Begrüßungszeremoniell den Initiationsritus des SSL/TLS-Protokolles, den TLS-Handshake mit gegenseitiger Authentifizierung nach dem RSA-Verfahren.

3. Keyed HMAC<sup>13</sup>s (Authentizität der Sendung, Integrität und Unleugbarkeit) Um sicherzustellen, dass ein Dritter in der verschlüsselten Nachricht nicht willkürlich Zeichen austauscht, wird zusammen mit jeder Nachricht ein Message Digest übertragen. In die Berechnung dieser Prüfsumme geht einer der Schlüssel (HMAC Send/Receive Key) ein, welche beide Seiten während des TLS-Handshakes generierten. Auf diese Weise wird die Datenintegrität gewährleistet. Denn schon der Austausch eines Zeichens macht die Prüfsumme ungültig. Der in die Prüfsumme einbezogene Schlüssel aber verhindert den gleichzeitigen Austausch von Botschaft und Prüfsumme (Authentizität der Daten) und gewährleistet zusätzlich die für den E-Commerce wichtige Unleugbarkeit. Denn er lässt sich eindeutig einer Identität zuordnen.

### ... und ihren verschlungenen Wegen

„95 % of the tech support problems that people have with VPNs are with the networking or firewall layers, not the cryptography layer.“ ([2], Folie 53)

Die Skalierbarkeit von VPN-Lösungen basiert auf zwei Säulen: der Public-Key-Kryptographie auf der einen sowie der Möglichkeit, Netzwerkpakete in andere Netzwerkpakete zu kapseln, auf der anderen Seite. Den Daten selbst wird – wie im letzten Abschnitt beschrieben ist – erstens etwas angetan, das sie vor Unbefugten verbirgt. Zweitens werden die Transportbehälter der Daten entsprechend markiert, dass sie auch auf der richtigen Poststelle ankommen, wo man sie entziffern kann. Davon handelt der folgende Abschnitt.

#### 1. Ein kleines Beispiel

Das folgende Szenario (Abb. 3) soll die Problematik verdeutlichen, vor der jede VPN-Implementierung steht, wenn nicht nur zwei Rechner, sondern mehrere Netze mit vielen Rechnern über große<sup>14</sup> Strecken miteinander zu verbinden sind.

<sup>13</sup> Keyed Hash Message Authentication Code (Keyed HMAC) ist ein Message Digest auf Basis einer kryptographischen Hashfunktion. Dabei geht ein geheimer Schlüssel in die Berechnung des Hashes ein.

<sup>14</sup> Groß ist hier nicht räumlich gemeint, sondern in Rücksicht auf die Anzahl der Vermittlungsstellen, die der Transport passiert.

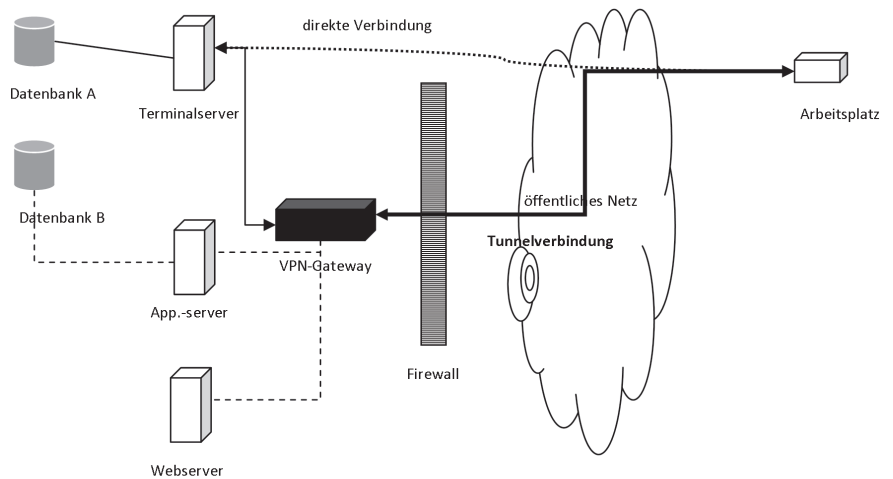


Abb. 3: Ein Beispielszenario

Der Datenverkehr über das Internet wird von Routern vermittelt. Diese kennen die Wege und entscheiden anhand eines Merkmales in den Headerinformationen der IP-Pakete, der IP-Adresse und der Netzwerkmaske über die Richtung des Weitertransportes. Die IP-Adresse wirkt wie ein Adressaufkleber an einem Päckchen. An dem Arbeitsplatz (Abb. 3, ganz rechts) mag nun ein Buchhalter in der Zweigstelle einer großen Organisation sitzen und über ein Buchhaltungsprogramm der Zentrale seine Umsätze mitteilen. Dieses Buchhaltungsprogramm läuft jedoch nicht auf seinem Arbeitsplatz, sondern auf einem sogenannten Terminalserver in der Zentrale. Der Buchhalter verbindet sich über eine Client-Anwendung (TS-Client) mit dem Terminalserver, wo er nach erfolgreichem Login sein Buchhaltungsprogramm öffnet. Die Buchhaltungssoftware bezieht ihre Datenbasis wiederum aus einer Datenbank, welche auf einem weiteren Server – dem Datenbankserver – liegt. Soweit läuft alles ohne Probleme über erprobte Kommunikationsstandards der Internettechnologie (Abb. 3 siehe: ‚direkte Verbindung‘).

Doch Finanzdaten sind vertrauliche Daten, weshalb einige zusätzliche Maßnahmen getroffen werden müssen. Im Netz der Zentrale befindet sich eine Maschine, die als sogenanntes VPN-Gateway arbeitet. Deren Aufgabe ist es, den Datenverkehr mit den externen Arbeitsplätzen über die Internetwolke in das durch eine Firewall geschützte vertrauenswürdige Netz der Zentrale sicher zu

tunneln. Als Tunnel sei hier einfach eine direkte Verbindung zwischen zwei Kommunikationspartnern bezeichnet, über die Daten an Dritte geschleust werden. Das Gateway soll also nicht nur *sichern*, sondern auch *weiterleiten*. Auf der anderen Seite des Tunnels arbeitet auf dem Buchhalter-PC im Hintergrund eine zweite Clientsoftware, der VPN-Client, die Gegenstelle zu dem VPN-Gateway. Diese verbindet sich mit dem Gateway, weist sich aus und verständigt sich mit ihm über die Modalitäten für die Absicherung der weiteren Kommunikation. In der Folge kann sie alle Daten, die an die Zentrale gehen, verschlüsseln und signieren. Umgekehrt prüft sie die Signatur der von der Zentrale kommenden Daten, entschlüsselt die Daten und reicht sie dem Buchhalter bzw. seiner Anwendung weiter. So vollzieht, wenn der TS-Client die Kommunikation in Richtung Terminalserver aufbaut, die neu eingeführte, in der Transparenz unscheinbare VPN-Software ihre kryptographischen Anwendungen und sendet alle Daten in ein zweites Paket verpackt und neu adressiert zum VPN-Gateway. Diese neu adressierten Pakete reisen nun genauso bequem über das Internet wie vorher die unbehandelten. Am VPN-Gateway angekommen, wird das äußere Paket ausgepackt, die Verschlüsselung entfernt und das Ursprungspaket gemäß seiner eigentlichen Zieladresse zum Terminalserver weitergeleitet.

Nun kann der Terminalserver antworten. Diese Antwort geht natürlich unverlüsselt und direkt an die im erhaltenen

Paket angegebene Absenderadresse. Das ist aber nicht die Adresse des VPN-Gateways, sondern die des Buchhalter-PCs. Da gibt es jetzt ein Problem: Wir unterstellen einmal, dass der Terminalserver solche direkten Verbindungen in das Internet überhaupt unterhalten darf und dass nicht eine Firewall diese unterbindet. Bei Einsatz von IPsec weist spätestens der IPsec-Paketfilter beim Empfänger solche Sendungen zurück, bei OpenVPN liegen die Dinge (wie weiter unten ausgeführt) komplizierter. Grundsätzlich muss man aber sicherstellen, dass solche Re-tourpakete nicht direkt, sondern über das VPN-Gateway vermittelt zugestellt werden. Denn erst dieses stellt Authentizität und Vertraulichkeit her.

Dafür gibt es nun verschiedene Handhaben, z. B. Network Address Translation (NAT) oder statische Routen, welche auch normale, ungeschützte Kommunikation vermitteln helfen, also nichts dem VPN Eigentümliches sind. Bei NAT gibt sich das Gateway gegenüber dem Terminalserver als Sender des Paketes aus und erhält demzufolge auch die Antworten vom Terminalserver, die es dann wieder verpacken und an den VPN-Client zum Buchhalter PC weiterleiten kann. Über statische Routen wird dem Terminalserver mitgeteilt, dass Pakete, die an des Buchhalters Adresse gehen, nicht über die vom Terminalserver zu benutzende Standardvermittlungsstelle (Default Gateway) gesendet werden dürfen, sondern über das VPN-Gateway. Wenn sich die Anzahl der Mitspieler auf der Serverseite erhöht (z. B. durch Einbindung von Domänencontrollern oder Updateservern) gestaltet sich das Ganze aber recht komplex.

Fürs Erste haben wir gesehen, dass Netzwerkmechanismen wie die Kapselung von Netzwerkpaketen und die ergänzenden Maßnahmen, welche sich für die Paketvermittlung daraus ergeben, für das VPN konstitutiv sind. Das betrifft IPsec (im Tunnel Mode) ebenso wie OpenVPN.<sup>15</sup>

<sup>15</sup> Gleich aus mehreren Gründen verbietet sich aber eine naheliegende Lösung: die VPN-Software direkt auf dem Terminalserver zu installieren. Schließlich lassen sich über das dedizierte VPN-Gateway auch andere Client-Server-Verbindungen bedienen, wie z. B. die Kommunikation mit einem Applikationsserver oder einem Webserver.

2. Jetzt wird es schwieriger  
Für die weiteren Betrachtungen müssen zum Verständnis einige Voraussetzungen erläutert werden.

Erstens: Netzwerkkommunikation ist ein komplexer, mehrschichtiger Vorgang. Damit Systeme unterschiedlicher Herkunft über diverse Übertragungsmedien miteinander kommunizieren können, bedarf es gewisser Standards, die Kodierungs- und Übertragungsregeln festlegen. Die allgemeine Vorlage dafür liefert ein Modell, das OSI-Referenzmodell.

Dieses Modell spiegelt sich in der Struktur der Datenströme, welche über das Internet gehen, wider. Für die Netzwerkhardware sind nur ganz bestimmte Bereiche dieses Datenstromes interessant (z. B. nicht die Adresszeile des Webbrowsers), die Vermittlungsstellen (Router) interessiert nur ein bestimmter Bereich und in den Höhen der Anwendung kommen die Niederungen der Kommunikation gar nicht mehr an. Wenn im Folgenden von Schichten oder Layern die Rede ist, bezieht sich das auf das OSI-Modell.

#### Das OSI-Referenzmodell (Open Systems Interconnection Reference Model) [8]

beschreibt den Informationsaustausch von Systemen, die einander offen sind durch den gegenseitigen Einsatz geeigneter Standards.

Ganz allgemein lassen sich an jeder Kommunikation gewisse Merkmale absehen, etwa dass sie aus einer Folge elektrischer Signale oder Tintenstrichen besteht, dass sie über ein Kabel oder den Äther erfolgt, dass die Striche oder Signalfolgen eine gewisse Struktur aufweisen etc. Für den Chip auf einer Netzwerkkarte, welche die elektrischen Signale aussenden soll, ist es unerheblich, was diese Signale etwa auf einem Router bewirken, so wie es für die eingesetzte Tinte unerheblich ist, ob in ihr lateinische oder japanische Sätze formuliert sind. Der Sinn eines Satzes bleibt derselbe, ob er in Stein gemeißelt oder auf einem Blatt Papier vorliegt.

Es sind also leicht verschiedene Abstraktionsebenen zu bilden, die sich voneinander unabhängig beschreiben und modellieren lassen. Eine dieser Ebenen könnte die Übertragung physischer Signale umfassen, eine weitere den Transport vermitteln, eine dritte die Signalstruktur beschreiben, in welcher sich für den Empfänger Information darstellt. Und für jede dieser Ebenen sind gewisse Normen, wie z. B. die phonetischen Bildungsregeln der Sprachlaute, verbindlich, an die sich die Beteiligten zum Zwecke der Verständigung halten müssen.

Diese Abstraktion hat ganz praktische Vorteile. Zum einen lassen sich die Normen formulieren, zum anderen können Hilfsmittel oder Dienste standardisiert zur Verfügung gestellt werden. Wie sich ein Briefeschreiber nicht erst einen Stift schnitzen muss, wenn er der Oma einen Brief schreibt (und einen zweiten Stift für die Steuererklärung an das Finanzamt), kann der Programmierer für die Veröffentlichung einer Webanwendung Bibliotheken benutzen, ohne sich mit der Netzwerkhardware des Servers oder den Datenstrukturen für die Paketvermittlung im Internet auseinandersetzen zu müssen.

Das OSI-Referenzmodell beschreibt den Kommunikationsprozess in sieben aufeinander aufbauenden Schichten, von der physischen Signalübertragung bis hin zur Zeichenrepräsentation. Die sieben Schichten des OSI-Modells sind: Physical Layer (Bitübertragung), Data Link Layer (Sicherheitsschicht), Network Layer (Vermittlungsschicht), Transport Layer (Transportschicht), Session Layer (Sitzungsschicht), Presentation Layer (Darstellungsschicht), Application Layer (Anwendungsschicht). Abbildung 5 versucht, die relevanten Schichten des OSI-Modells auf den Kommunikationsprozess von OpenVPN abzubilden.

Das Modell ist selbst kein Kommunikationsstandard, sondern formuliert, welche Dienste von einem Layer dem darüberliegenden Layer angeboten werden müssen bzw. wie die angebotenen Dienste genutzt werden. Es ist also ein Leitfaden für die Entwicklung von Kommunikationsprotokollen. Die konkrete Umsetzung erfolgt in den ausformulierten Protokollen der jeweiligen Schicht bzw. deren Implementation durch ein Betriebssystem.

Kasten 2: das OSI-Referenzmodell

Zweitens: Auf einer dieser Schichten (Layer 3, die Netzwerkschicht, welche für die Paketvermittlung zuständig ist) spielen nun die IP-Adressen eine Rolle. Diese sind nach Adressbereichen gruppiert, wie die Postanschriften bestimmten Postleitzahlbereichen angehören, und zwar aus demselben Grund: um sie effizienter befördern zu können. Der Verkehr läuft nie direkt zwischen zwei Briefeschreibern oder zwei PCs. Immer gibt es Vermittlungsstellen, die dazwischengeschaltet und für bestimmte Postleitzahlbereiche oder IP-Adressbereiche zuständig sind. Jenseits jenes Layers 3 helfen auch die beliebten Rückgriffe auf Beispiele aus dem Postbetrieb nicht mehr. Man stelle sich also ein Dorf vor, wo noch sehr viel persönlich und per Hand erledigt wird, mit Netzwerkbriefkästen, die nur den registrierten Einwohnern offenstehen. Jedes Netzwerkgerät besitzt eine (theoretisch) weltweit eindeutige Hardwareadresse, die MAC-Adresse. Innerhalb eines lokalen Netzes (LAN) kann jeder Sender jeden Empfänger direkt über diese MAC-Adresse adressieren, und jeder Empfänger nimmt nur die Sendungen an, die auch für seine Hardwareadresse bestimmt sind (es sei denn, das Paket ist ausdrücklich für alle gemeint). Die IP-Adresse und die dazugehörige Subnetzmaske werden benötigt, um zu entscheiden, ob ein Kommunikationspartner zum lokalen Netz gehört oder nicht. Ist das nicht der Fall – wie im oben angeführten Beispiel – wird das Paket an die MAC-Adresse eines bestimmten Gerätes (des Default Gateways) im LAN gesendet, welches sich um die Weitervermittlung der Pakete an entfernte Adressaten kümmert.<sup>16</sup> Zu Beginn des Verbindungsaufbaus kann die Netzwerklogik des Betriebssystems zwar feststellen, dass ein Paket nicht an einen lokalen Adressaten geht, sondern an das

<sup>16</sup> Die Anwendungen, welche auf den höheren Schichten des OSI-Modells miteinander kommunizieren (z. B. der Terminal Service Client), verwenden für die Adressierung selbstverständlich nicht diese MAC-Adressen, sondern Merkmale, die leichter zu konfigurieren und vor allem nicht so zufällig wie die Zurechnung einer MAC-Adresse in einen lokalen Netzwerkbereich sind (die Netzwerkkarte kann schließlich kaputtgehen und ausgetauscht werden). Außerdem gehen moderne Netzwerkanwendungen ganz selbstverständlich davon aus, über ein Internet zu laufen. Da wird also für das Ziel der Verbindung so etwas wie ein registrierter Name (DNS-Name) oder wenigstens eine IP-Adresse konfiguriert.

Default Gateway, die 'Hardwareanschrift' des Default Gateways muss sie aber erst ermitteln. Über einen sogenannten Address Resolution Protocol Request (ARP-Request) wird die zur bekannten IP-Adresse zugehörige MAC-Adresse eingeholt und für weitere Verbindungen gespeichert. Diese Requests sind keine IP-Pakete, werden also niemals einen Router passieren, sondern ein Kommunikationsprotokoll, das auf der Ebene 2 des OSI-Modells arbeitet. Diese Requests verwenden zur Zustellung das Merkmal MAC-Adresse und zwar, weil sie an alle 'Einwohner' adressiert sind, in ganz allgemeiner Form (Broadcast). Trotzdem alle diese Sendung erhalten und auswerten: nur der Inhaber der IP-Adresse antwortet.

Drittens: Oberhalb des Layers 3 befindet sich die Transportschicht. Hier gibt es zwei Kommunikationsprotokolle: TCP<sup>17</sup> und UDP<sup>18</sup>. Diese Schicht kümmert sich um die Ende-zu-Ende-Kommunikation der Anwendungen, stellt diesen gleichsam Briefkästen (Portadressen) zur Verfügung. OpenVPN verwendet eine dieser Datenstrukturen, das UDP-Protokoll<sup>19</sup>, um Datenpakete einzupacken.<sup>20</sup> Kapselung bedeutet aber, dass nicht einfach nur die höheren Schichten verschlüsselt, signiert etc. sind (wie etwa beim Aufruf einer Webseite über SSL). Es werden komplette Datenstrukturen aus tieferen Schichten des OSI-Modells

<sup>17</sup> Transmission Control Protocol

<sup>18</sup> User Datagram Protocol

<sup>19</sup> Das von OpenVPN hierfür standardmäßig gewählte Protokoll für die Kapselung ist das Transportprotokoll UDP, die verbindungslose, 'unzuverlässige' Schwester des Transportprotokolls TCP. Bei der Übertragung über das Internet können Pakete verloren gehen oder beschädigt werden. TCP verfügt über einige zusätzliche Mechanismen, die dem Rechnung tragen und Korrekturen ermöglichen. Kapselt man nun (über IP-Pakete oder Ethernet Frames) TCP-Datagramme in andere TCP-Datagramme, gibt es eine Redundanz dieser Sicherheitsmechanismen, die zu Störungen, wie z. B. häufigen Verbindungsabbrüchen, führen kann. Deshalb verwendet man ein Trägerprotokoll, welches diese Sicherheitsmechanismen nicht kennt – eben UDP. Zuverlässigkeit wird über die in den gekapselten Daten vorhandenen Mechanismen der höheren Protokollschichten erreicht. Es besteht jedoch die Möglichkeit, OpenVPN auch über TCP zu betreiben.

<sup>20</sup> IPsec verwendet zwei spezielle Layer 3-Protokolle, deren Informationen nach dem IP-Protokoll-Header eingefügt werden und in ihrer Payload entweder alle höherschichtigen Protokolle kapseln (Transport Modus) oder ein weiteres IP-Paket einschließen. (siehe auch Abb. 1)

(Ethernet Frames aus Layer 2 oder IP-Pakete aus Layer 3) in die Payload des UDP-Datagramms eingefügt. Dadurch funktioniert der VPN-Dienst Protokollunabhängig und für alle höherschichtigen Anwendungen gleich. Erst diese Kapselung macht aus einem SSL-VPN ein echtes VPN.

Viertens: Im Unterschied zu IPsec ist OpenVPN in der Lage, neben IP-Paketen (Layer 3) Ethernet Frames (Layer 2) zu kapseln. Das bedeutet, über einen OpenVPN-Tunnel können nicht nur IP-Pakete, sondern auch Datenstrukturen anderer Layer 3-Protokolle wandern bzw. Broadcasts – wie der oben erwähnte ARP – gesendet werden. Ferner lassen sich über OpenVPN auch die Netzwerkgeräte, welche auf Ethernet Layer arbeiten, (Bridge, Switch) emulieren und virtuelle Ethernet LANs erzeugen, die über Internetrouten zustande kommen.<sup>21</sup> OpenVPN verwendet dazu virtuelle Netzwerkgeräte, tun- oder tap-Devices.

Was geschieht, wenn für eine in einem bestimmten Postleitzahl- oder IP-Adressbereich aufgegebenen Sendung als Absenderanschrift der Bereich der Zieladresse angegeben wird? Wahrscheinlich dürfte das für die Post kein Problem sein, zumindest für den Fall, dass das Päckchen zustellbar ist. Ein Router allerdings müsste dieses Paket verwerfen, wenn es überhaupt die heimische Netzwerkkarte verlässt. Warum aber sollte jemand auf so eine abwegige Idee kommen? Das hängt mit der oben besprochenen Kapselung und deren Auswirkungen auf die Transportdirektiven zusammen. In komplexen Netzwerkkombinationen gibt es nicht nur – wie im Beispiel – zwei oder drei Server oder Dienste, welche zusammenwirken müssen, damit ein dezentraler Arbeitsplatz voll funktionstüchtig ist. Darum kann es sinnvoll sein, den entfernten Arbeitsplatzrechner direkt in den Adressbereich des Zielnetzes in der Zentrale einzubin-

<sup>21</sup> OpenVPN kapselt IP-Pakete oder Ethernet Rahmen in der Payload von UDP-Datagrammen. IPsec kapselt IP-Payload bzw. – im tunnel mode – komplette IP-Pakete. Es gibt aber Konfigurationen für verschiedene BSD-Derivate, wo das Etherip-Protokoll im Betriebssystem implementiert ist ([6][7]). Das sind IP-Pakete, die Ethernet Rahmen transportieren. Diese können von IPsec dann gekapselt werden.

## tun/tap

Tun und tap sind Treiber des Betriebssystemkerns für virtuelle Netzwerkgeräte. Diese bestehen nur in Software, d. h. hinter diesen verbirgt sich keine physische Netzwerkkarte und es ist auch kein Netzwerkkabel mit ihnen verbunden. Der Treiber erzeugt auf Anforderung (Öffnen von `/dev/net/tun` für Lese- und Schreibzugriff) das entsprechende Netzwerkgerät und stellt für dieses zwei Interfaces zur Verfügung: ein character device `/dev/tunX` oder `/dev/tapX` sowie ein virtuelles Netzwerkinterface `tunX` oder `tapX`. Daten, die vom Betriebssystemkern an `tunX` oder `tapX` gesendet werden, können von einer Anwendung, die sich mit dem Gerät verbindet, über `/dev/tunX` oder `/dev/tapX` ausgelesen werden. Umgekehrt schreibt die Anwendung speziell formatierte Daten wie in eine Datei nach `/dev/tunX` oder `/dev/tapX`. Der Betriebssystemkern empfängt diese, als kämen sie aus dem realen Netzwerk.

Ist das virtuelle Netzwerkgerät einmal vorhanden, kann man es wie ein echtes Netzwerkgerät benutzen, d. h. eine IP-Adresse vergeben, das Routing konfigurieren, Firewallregeln auf den Datenverkehr durch das Interface legen usw.

Ein tun-Device arbeitet dabei mit reinen IP-Paketen (OSI-Layer 3) ohne Ethernet-Header, das tap-Device erwartet und übergibt Ethernet Frames (OSI-Layer 2) und wird verwendet, um physisch getrennte Subnetze oder Netzwerkgeräte in einem Ethernet LAN zusammenzufassen.

Das tap-Device erscheint z. B. in der Übersicht mit dem Kommando `ip addr` als normale Netzwerkkarte neben anderen Netzwerkkarten. Das tun-Device repräsentiert – wie unten dargestellt – eine IP-Verbindung:

```
>ip addr
...
3: vpn-init: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500
   qdisc pfifo_fast state UNKNOWN qlen 100
    link/none
    inet 172.16.0.1 peer 172.16.0.2/32 scope global vpn-init
...
```

Kasten 3: tun/tap

den. Man entschlägt sich damit der Kalamitäten, welche die Paketvermittlung im obigen Exkurs bereitete, denn diese wird dann von der schon bestehenden Infrastruktur des Zielnetzes bereitgestellt. Dieses Problem, vor welches die Layer 3-Vermittlung gestellt ist, lässt sich nur durch einen Trick umgehen.

Sehen wir uns dazu den Einsatz von OpenVPN im Verwaltungsnetz an: Ein Thin Client ersetzt den PC, und der VPN-Client läuft in einem zwischen diesen und das Netz der Fakultät geschalteten Netzwerkgerät (in der VPN-Box, ein Linksys-Router mit OpenWrt-Betriebssystem). Abbildung 4 zeigt das Schaltbild eines solchen Linksys-Gerätes, das um die für unsere OpenVPN-Konfiguration notwendigen Ergänzungen erweitert wurde.

Beim Starten der VPN-Box werden zunächst durch das Betriebssystem die Netzwerkinterfaces konfiguriert, die

VLANs sowie die Bridge eingerichtet. Diese erhält eine über einen Algorithmus aus der MAC-Adresse abgeleitete private IP-Adresse. Die 5 physischen Schnittstellen haben dieselbe MAC-Adresse, sind

aber verschiedenen VLANs (virtuellen LANs) zugewiesen: standardmäßig der Uplinkport (Internetport) dem einen, die 4 LAN-Ports einem zweiten. Der Netzwerkverkehr in dem einen VLAN ist in dem anderen nicht sichtbar und ein Austausch zwischen beiden zunächst nicht vorgesehen. Der Uplink-Port erhält eine IP-Adresse im Netz der Fakultät/ des Providers. Wireless LAN ist deaktiviert (und erscheint deshalb nicht in der Abbildung). OpenVPN legt beim automatischen Start ein tap-Device an und fügt es der bestehenden Bridge hinzu. Wir haben also jetzt auf dem Gerät zwei getrennte Netzwerkbereiche (VLANs) und in dem einen dieser VLANs eine Bridge, bestehend aus den physischen LAN-Ports der VPN-Box und dem virtuellen tap-Device. Dann wird – gemäß der Konfiguration – über einen TLS-Handshake eine verschlüsselte IP-Verbindung (Tunnel) mit dem VPN-Gateway in der Ferne ausgehandelt und aufgebaut.

Diese besteht aus zwei Kanälen: einem für die Nutzlast, die Datenpakete, welche durch den Tunnel befördert werden, und einem Kontrollkanal, über welchen u. a. die Kommunikation für das Schlüsselmanagement verläuft. Die Verbindung verläuft zwischen den Anwendungssockets der beiden OpenVPN-Instanzen. Beide Sockets verfügen über eine routbare IP-Adresse und eine Portnummer (Default 1194).

Startet jetzt auf dem Thin Client der TS-Client, empfängt die VPN-Box einen ARP-Request auf einem der LAN-

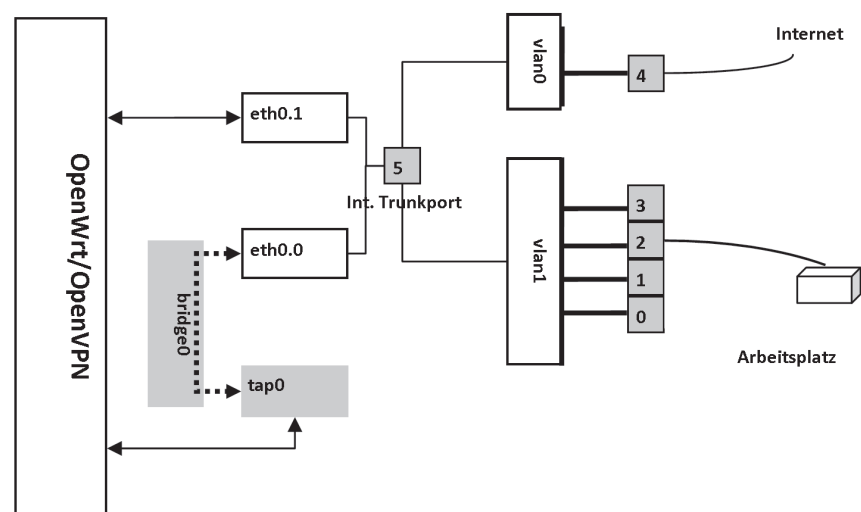


Abb. 4: die VPN-Box

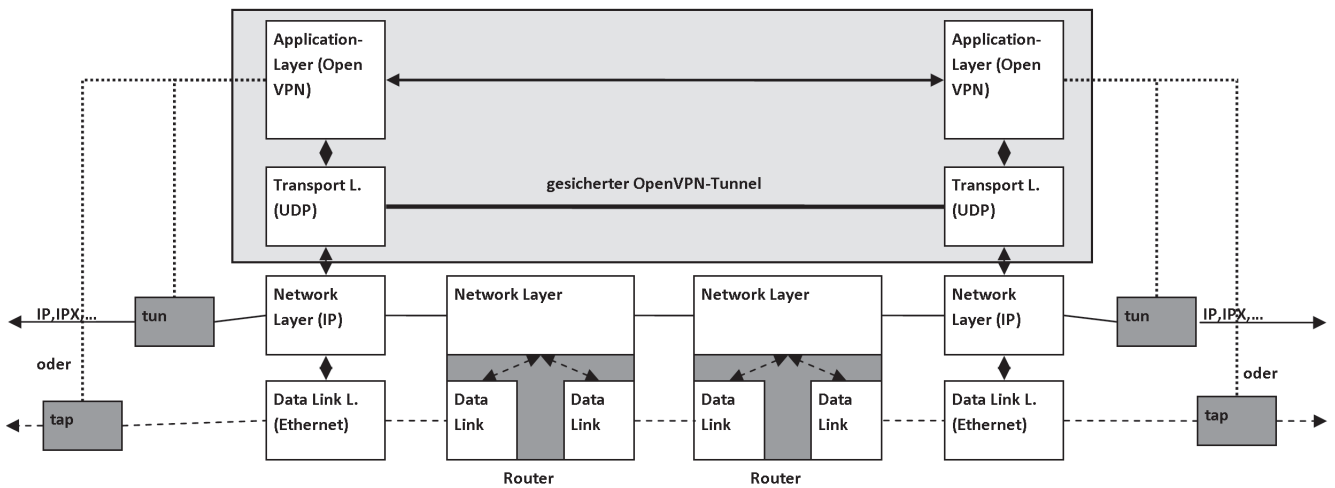


Abb. 5: tun und tap-Device im OSI-Modell

Ports<sup>22</sup>. Da dieser an alle Teilnehmer des LAN gerichtet ist und das LAN über die Bridge zwischen LAN-Ports und tap-Device vermittelt wird, landet der Request auch beim tap-Device, wird von OpenVPN gelesen, in ein UDP-Datagramm verpackt und über den Tunnel verschlüsselt zu der VPN-Instanz auf dem VPN-Gateway geroutet. Auf der Gatewayseite verarbeitet das OpenVPN die an es adressierten Tunnelpakete (äußeres Paket mit socket-Adresse als Anschrift), entschlüsselt und entpackt sie und schreibt diese in sein eigenes tap-Device. Die entpackte Payload im tap-Device besteht nun aus dem Ethernet Frame mit dem ARP-Request. Das tap-Device funktioniert wie eine Netzwerkkarte, d. h. es hat eine IP-Adresse und eine MAC-Adresse. Die IP-Adresse des tap-Devices auf dem Gateway entstammt dem Subnetz der Bridgeadresse vom VPN-Client. Die Bridge auf dem Client 'sieht' also die MAC-Adresse des Gateway-tap-Devices (und hat diese auch bereits gelernt<sup>23</sup>). Beide befinden sich in einem LAN oder: das tap-Device des VPN-Gateways steckt virtuell genauso in einem Port der Bridge wie der Thin Client mit seinem realen Netzwerkkabel. Der Frame unseres Thin Clients ist nun also auf der virtuellen tap-Netzwerkkarte

des VPN-Gateways gelandet und fragt immer noch seinen ARP-Request nach dem Default Gateway. Weiter muss er aber nicht. Neben der privaten IP-Adresse in dem tap/Bridge-LAN hat das tap-Device des Gateways eine zweite IP-Adresse, die des Default Gateways für das Subnetz des Thin Clients. Das tap-Device, welches ja als Ethernet-Gerät ARP beherrscht, kann also mit seiner MAC-Adresse antworten.<sup>24</sup> Nun ist der Thin Client nach der Antwort auf seinen ARP-Request wirklich angekommen in seinem (fernen) Subnetz. Und er lernt auch, solange er läuft, nichts anderes kennen. Die Kommunikation im LAN (mit den anderen 'Dorfbewohnern', den anderen Thin Clients) geht über den VPN-Tunnel ebenso, wie alles andere, was über das Default Gateway zu vermitteln ist, wie z. B. eine Browsersitzung. Wenn etwa auf dem Thin Client der Internet Explorer gestartet wird, der – weil unkonfiguriert – wahrscheinlich irgendeine Seite von Microsoft ansteuert, geht das Ganze die selben verschlungenen Wege. Der Thin Client weiß nichts von den VLANs und der Bridge auf der VPN-Box. Diese Strukturen sind ebenso transparent wie der VPN-Tunnel. Für ihn

sieht die Welt so aus wie für gewöhnliche PCs, die ihr LAN quasi direkt aus der Dose in der Wand bekommen.

Nachdem der Thin Client sich auf diese Weise in seiner neuen Netzwelt orientiert hat, wird alles wieder ganz einfach. Das Netz der Zentrale, welches sich auch in Subnetzen strukturiert, ist dem entfernten Thin Client genauso zugänglich wie Rechnern in einem Subnetz, das physisch direkt im Verwaltungsnetz hängt. Zwar sind jetzt im Verwaltungsnetz immer noch Zurüstungen zu treffen (namentlich muss irgendwo stehen, dass das virtuelle LAN der Thin Clients unter dem Anschluss des VPN-Gateways erreichbar ist), aber diese Maßnahmen sind – für komplexe Anwendungsfälle – weniger aufwändig als im ersten Beispiel.

OpenVPN wird von uns auch im Modus mit tun-Device verwendet, um die VPN-Boxen beim Start zu konfigurieren.<sup>25</sup> Dies sei nur am Rande erwähnt.

Die virtuellen Netzwerkkarten tun und tap von OpenVPN verhalten sich wie Netzwerkbriefkästen, über die OpenVPN Netzwerkpakete (etwa aus dem Innern des Verwaltungsnetzes oder von einer Anwendung) im Klartext lesen kann, um diese dann über seinen Netzwerksocket verschlüsselt und signiert ins Internet weiterzuleiten – und umgekehrt natürlich.

22 Das ist allerdings nicht der erste Netzwerkruf des erwachenden Systems, weil dieses vorher noch über DHCP seine IP-Konfiguration ermittelt.

23 im Ergebnis der DHCP-Unterhandlungen, über die sich der Thin Client seine IP-Konfiguration und damit seine IP-Adresse im Verwaltungsnetz geholt hat

24 Übrigens wird auch gatewayseitig im ARP-Cache die MAC-Adresse des Thin Clients abgelegt. Und nicht etwa die MAC-Adresse der VPN-Box, welche erst dann dort erscheint, wenn man etwa per ssh vom VPN-Gateway auf die private IP-Adresse der VPN-Box-Bridge geht. Dafür gibt es in der Routing-Tabelle des Kernels einen Eintrag, der besagt, dass gesuchtes Subnetz direkt über das tap-Device zu erreichen ist.

25 Auch das IPsec-VPN kennt neben der normalen Tunnelmode-Verbindung für die Sacharbeit eine weitere Verbindung im Transport Mode für die Konfiguration und Steuerung des Tunnels.

## Fazit

Man erwartet jetzt wahrscheinlich noch eine Aufstellung darüber, was ‚besser‘ ist: OpenVPN oder IPsec. Im Unterschied zu Layer 3-VPN-Technologien wie IPsec kann OpenVPN von Hause aus auch auf dem Layer 2 des OSI-Schichtenmodells arbeiten, es ist also in der Lage, Ethernet-Frames zu kapseln und eine solche Konstruktion, wie oben geschildert wurde, zu ermöglichen. Für OpenVPN spricht die Portierbarkeit, weil es als User Space-Programm läuft und mit seinen virtuellen Interfaces weniger Annahmen über die Netzwerkimplementation des Betriebssystemkerns machen muss; anders als IPsec, welches sich direkt in den Netzwerk-Stack einklinkt. Allerdings sind sowohl die virtuellen Interfaces als auch das Bridging Features des Betriebssystems bzw. Module, welche mit Administratorrechten geladen werden. OpenVPN nutzt die OpenSSL-Bibliotheken. Die kryptographischen Funktionen – wie die Cipher Suite – stehen unabhängig von der OpenVPN-Implementation zur Verfügung, während IPsec diese aus dem eigenen Code bezieht. Das wird wohl der Hauptgrund für die Interoperabilität der OpenVPN-Implementationen in verschiedenen Betriebssystemumgebungen sein – ganz im Gegensatz zu den vom jeweiligen Hersteller nach Geschmack implementierten IPsec-Transforms. Ob OpenVPN – wie oft gelobt – einfacher zu konfigurieren ist und weniger Lernaufwand als IPsec erfordert, mag jeder selbst beurteilen. Aber so etwas, wie die oben beschriebene Konfiguration verlangt schon genaue Kenntnis des Betriebssystems. Will man sie dann auch noch verstehen, steht man auf recht verlorenem Posten. Denn im Unterschied zu dem über RFCs ausgezeichnet dokumentierten IPsec, zu dem es in den Software-Handbüchern immer auch einen groben Überblick darüber gibt, was etwa ESP bedeutet oder worum es sich bei einer Diffie-Hellman-Gruppe handelt, findet man im Userland des OpenVPN eher Beschreibungen zu den diversen Konfigurationsmöglichkeiten für Bastler. Da freut es, wenn man im Internet einmal zwei Anwender findet, die sich darüber austauschen, wo denn

nun in den von OpenVPN benutzten SSL-Handshake-Modi die Authentifizierung konkret einhakt.<sup>26</sup>

OpenVPN kann im Unterschied zu IPsec als wenig privilegierter Dienst laufen. Eine Hintertür öffnet dem Angreifer also nicht gleich das ganze Betriebssystem. Und – es ist Open Source.

In Hinsicht auf die Tunnelsicherheit gibt es keine Unterschiede zwischen beiden VPN-Techniken. Was aber die restliche Infrastruktur unserer VPN-Lösungen anbetrifft, folgen ein paar Anmerkungen:

- Thin Client und VPN-Box können im Fehlerfall leichter ersetzt werden als ein PC.
- Für beide besteht die Möglichkeit zentraler Administration (auch wenn diese für die Wyse Thin Clients noch nicht realisiert wurde).
- CUPS erleichtert die Druckerverwaltung enorm und ist für mich persönlich das Feature schlechthin (auch wenn es gar nichts mit dem VPN im engeren Sinn zu tun hat). Ich möchte aber an dieser Stelle davor warnen, solche Canon-Drucker anzuschaffen, welche für die Kommunikation und den Druck nur auf proprietäre Protokolle setzen.
- Im Vergleich zu unserem IPsec-VPN benötigen wir keine Network Address Translation. Dafür ist aber die ordentlich und ohne ‚Taschenspielertricks‘ geroutete Verbindungslogik von IPsec leichter zu durchschauen als ein tap-getriebenes OpenVPN.
- Was das Problem Bridging versus Routing angeht, so haben wir von den Vorteilen des Bridgings außer dem oben genannten wenig, da wir weder auf Windows-Broadcasts angewiesen sind noch andere Netzwerkprotokolle – wie etwa IPX – einsetzen. Dagegen können sich die ARP- und DHCP-Broadcasts<sup>27</sup> noch als lästig erweisen, wenn das Clientnetz wächst. Die Skalierbarkeit und damit auch die Transparenz für die Nutzer stehen also auf dem Prüfstand.

<sup>26</sup> Das hatte ich oben unterschlagen. Der Zertifikatsaustausch macht ja nur Sinn, wenn dann auch etwas signiert oder verschlüsselt wird, um die Identität des Gegenübers zu prüfen.

<sup>27</sup> Schon jetzt gibt es zuweilen regelrechte Anfragefluten auf dem DHCP-Server.

## Literatur

- [1] BELL, MICHAEL; HERBST, ROLAND; HOKE, TILL; NATUSCH, DORIS; SCHWAN, MATTHIAS: *Bausteine für eine sichere Hochschulverwaltung*. DFN-Bericht. Berlin, 2003. DFN e.V. (Hrsg.)
- [2] YONAN, JAMES: *The User-Space VPN and OpenVPN*. <http://openvpn.net/papers/BLUG-talk/index.html>, 2003.
- [3] CHARLIE HOSNER: *OpenVPN and the SSL VPN Revolution*. SANS Institute InfoSec Reading Room. [http://www.sans.org/reading\\_room/whitepapers/vpns/openvpn-ssl-vpn-revolution\\_1459](http://www.sans.org/reading_room/whitepapers/vpns/openvpn-ssl-vpn-revolution_1459), 2004.
- [4] SNADER, JOHN C.: *VPNs Illustrated: Tunnels, VPNs, and IPsec*. Amsterdam: Addison-Wesley, 2005.
- [5] HEISE SECURITY: *FBI-Backdoor in IP-Sec-Implementierung von OpenBSD?* <http://www.heise.de/security/meldung/FBI-Backdoor-in-IPSec-Implementierung-von-OpenBSD-1153180.html>, Dezember 2010.
- [6] *Virtual Ethernet Tunneling*. <http://www.indelible.org/ink/tunneling/>, 2009
- [7] NIELS PROVOS: *The IPsec Architecture in OpenBSD*. <http://www.openbsd.org/papers/ipsec-slides.ps>
- [8] ISO/IEC 7498-1: *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. Second Edition. Corrected and reprinted 1996. [http://standards.iso.org/ittf/PubliclyAvailableStandards/so20269\\_ISO\\_IEC\\_7498-1\\_1994\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/so20269_ISO_IEC_7498-1_1994(E).zip)