



# 1. VORLESUNG

14.10.24 - ITS

## Beispiel aus der Praxis (sudo - Folien fehlen)

### 19.88 Internet Worm

#### > "How it works"

- Wie befällt er neue Maschinen?

- sendmail Bug (seit langem bekannt)
- finger Bug; Buffer Overflow (nur VAX befallen)
- Remote execution (rsh, rexec)

- Welche Accounts werden angegriffen?

- Offensichtliche Passwörter
- Build-In-Wörterbuch
- /usr/dict/words
- Trusted Host Beziehung (.rhosts)

- Was der Wurm nicht tut:

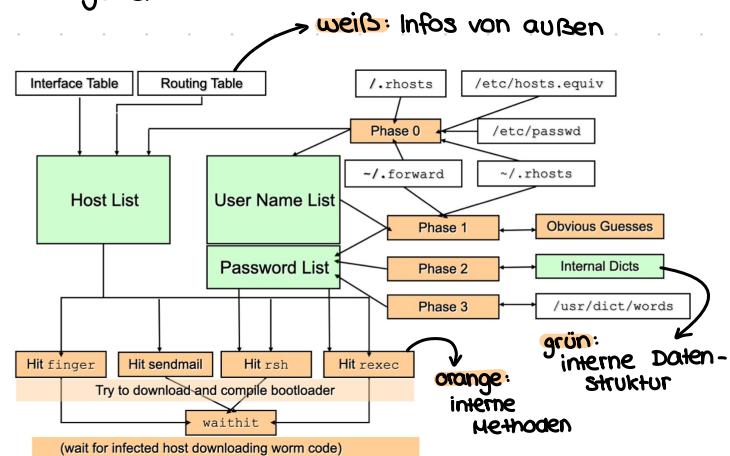
- Versuchen root access zu erhalten (wollte nicht die Maschine übernehmen)
- well-known Accounts angreifen
- Daten zerstören → z.B. BrutalTrojaner
- „Zeitbomben“ zurücklassen

#### Programm Struktur:

```
main Routine
argv[0] := "sh"; /* rename process */
Is there already a worm? /* faults here causes mass infection */
Initialize clock;
while (true) {
    /* um PW zu brechen
    cracksome(); /* attack accounts, try to find hosts */
    sleep(30); /* hide the worm */
    Listen for other worms /* faults here causes mass infection */
    create a new process, kill the old /* Camouflage */ → Prozess-ID wechselt
    try to attack some machines;
    sleep(120); /* hide the worm */
    if (running > 12 hours) → aufgeräumt
        cleaning host List; /* reduce memory consumption */
    if (pleasequit && wordcheck > 10)
        exit
}
```

- Welche Hosts werden angegriffen?

- /rhosts und /etc/host.equiv
- .forward Datei gebrauchter Accounts
- rhosts Datei gebrauchter Accounts
- Gateways aus der Routing-Tabelle
- Endpunkte von Point to Point Verbindungen
- Zufällig geratene Adressen
- Nur Sun und VAX



#### Lessons Learned

- (lange) bekannte Bugs fixen
- Starke PW nutzen
- Least privilege Prinzip
- Logging & Auditing
- Keine reflexartigen Reaktionen, z.B. Systeme herunterfahren
- CERT (Comp. Emergency Response Team)

#### Slammer

##### Verbreitung und Folgen

- Schnellster Wurm in der Geschichte (Verdopplung der Pop. alle 8,5 s)

- Folgen:

- Große Teile des Internets nicht mehr erreichbar
- Steuerungssysteme für die Stromversorgung gestört
- Fkt. Störungen bei Geldautomaten
- Steuerrechner von zwei Atomkraftwerken in den USA betroffen

## "How it works"

- Slammer passt in ein UDP Packet
- Slammer nutzt Buffer-Overflow an UDP Port 1434 (SQL-Monitor, der auffällig war)
- Nach Infektion:
  - "Raten" zufälliger IP-Adressen
  - Angriff über UDP
- Keine Schadfkt. im eigentlichen Sinn
- Charakteristika:
  - nur durch Bandbreite beschränkt
  - Konkurrenz mit anderen Würmern wg. aggr. Verbreitungsstrategie

## Lessons Learned

- Grundproblematik: Nicht behobene Bugs in Anwendungen
- Bundling von Software, z.B. Zeichenprogr.
- Angriffe über UDP können zu extrem schneller Verbreitung führen
- Gegenmaßnahmen:
  - Filtern des entspr. Verkehrs (UDP Port 1434) ü. Firewall
  - Fehler und Schwächen beheben
  - Nicht notwendige Dienste abschalten

# 2. VORLESUNG

21.10.24 - ITS

## KAPITEL 2: GRUNDLAGEN

### Ziele der Informationssicherheit

#### Hauptproblem:

- Informationssicherheit kann nicht gemessen werden:
- Es gibt keine Maßeinheit für IS
  - Sicherheitskennzahlen (security metrics) quantifizieren nur Teilespekte

#### Lösungsansatz: Indirekte Def. von IS durch (Teil-) Ziele

Vertraulichkeit	- Confidentiality	} CIA	jeweils bezogen auf
Integrität	- Integrity		Daten und sie verarbeitende
Verfügbarkeit	- Availability		IT-Systeme

#### ① Vertraulichkeit

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

- In vernetzten Systemen zu betrachten bzgl.: Transport, Speicherung, Verarbeitung
- Sicherheitsmaßnahme: Verschlüsselung
- Teilziel gilt als verletzt, wenn geschützte Daten v. unautorisierten Sbj.en eingesehen werden können.

Kontext Dienste: Vertrauliche IT-Dienste können nur von autorisierten Anwendern genutzt werden.

z.B. E-Mails

#### ② Integrität

Def.: 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 → digitaler Fingerabdruck
- Teilziel verletzt, wenn Daten von unautorisierten Sbj.en unbemerkt verändert werden.
- Kontext Dienste: Integre IT-Dienste haben keine (versteckte) Schadfkt. → typ. Verletzung: Trojaner

Teillösung können sonst auch verändert werden.

z.B. Online-Banking

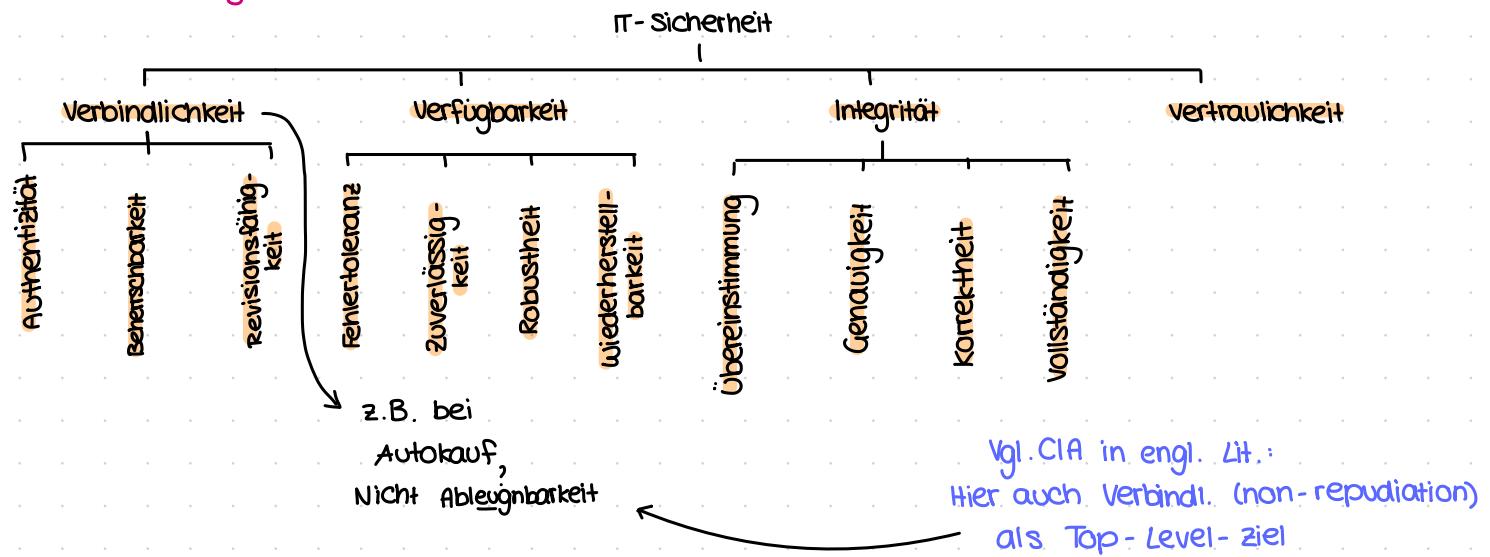
#### ③ Verfügbarkeit

Def.: Verfügbarkeit (engl. availability) ist gewährleistet, wenn autorisierte Sbj.e 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
  - Teilziel verletzt, wenn ein Angreifer die Dienst- und Datennutzung durch legitime Anwender einschränkt.
- z.B. Webserver

### Ziele und abgeleitete Ziele in dt. IS-Literatur

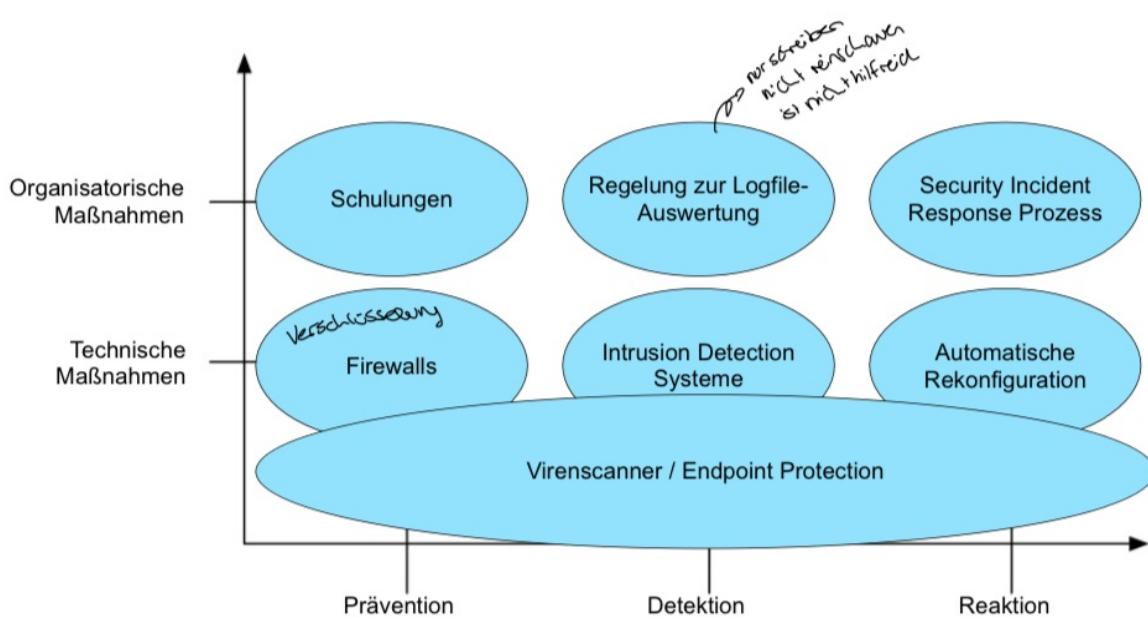


### Systematik zur Einordnung von Sicherheitsmaßnahmen

#### Warum Sicherheitsmaßnahmen einordnen?

- zum Erreichen der IS-Teilziele müssen Sicherheitsmaßnahmen umgesetzt werden
- Sicherheitsmaßnahmen gibt es zuhauf; sie entwickeln sich wie Dienste und Angriffe ständig weiter.
- Wir orientieren uns an zwei bewährten Dimensionen:
  - Lebendzyklus potentiell erfolgreicher Angriffe auf Dienste / Daten
  - Unterscheidung zw. techn. u. organ. Maßnahmen (=> Faktor Mensch)

#### 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 Sicherheitsereignissen, bei denen die präventiven Maßnahmen unzureichend waren.
- Reagierende Maßnahmen dienen der Wiederherstellung des Soll-Zustands nach dem Erkennen von unerwünschten Sicherheitsereignissen

## Welche Maßnahmen werden benötigt?

### Grundidee:

- Maßnahmenauswahl ist immer szenarienspezifisch
- Risikogetriebenes Vorgehensmodell
  - ↳ Vorteil risikobasierter Ansatz: Wkeit Angriff schaden

### kernfragesstellungen: → geringes Risiko ⇒ Geld nicht wert

- Welche Sicherheitsmaßnahmen sollen wann und in welcher Reihenfolge ergriffen werden?
- Lohnt sich der damit verbundene Aufwand?

### Voraussetzung Risikomanagement:

- Analyse des Schutzbedarfs
- Überlegungen zu mögl. Angriffen und deren Auswirkungen
- Ermittlung / Evaluation passender Lösungswege
- Entscheidung möglichst auf Basis quantitativer Bewertung

## Technik & Organisation - ISO/IEC 27000

(wichtigste: 27001 E- 27002)

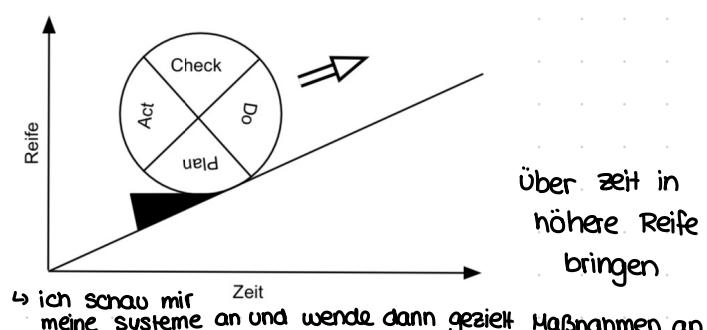
ISO/IEC 27000 → Idee: Anwendung der Grundprinzipien des Qualitätsmanagement auf das Management der Info.sich

ISO/IEC 27000 wird mehrere Dutzend einzelne Standards umfassen → wichtig: muss die Umsetzen falls zertifiziert

Norm ISO/IEC 27001 legt Mindestanf. an sog. Information Security Management Systems (ISMS) fest

- Zertifizierungen mögl. für:

- Organisationen (seit 2005)
- Personen (seit 2010)
- Inhaltliche Basis: durch Anwendung des Deming-Zyklus (PDCA)
  - Kontinuierl. Verbesserung
  - Risikogetr. Vorgehen
- Seit 2008 auch DIN ISO/IEC 27001



## Kerninhalte / Struktur von DIN ISO/IEC 27001

- Begriffsdef.
- PDCA-basierter Prozess zum Konzipieren, Implementieren, Überwachen und Verbessern eines ISMS
- Mindestanf. u.a. an Risikomanagement, Dokumentation und Aufgabenverteilung
- Normativer Anhang A enthält: → so besonders: nicht nur Anhang sondern integrativer Bestandteil, muss umgesetzt werden
  - Def. v. Maßnahmen (controls)
  - Gruppierung in 4 Kategorien

## 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: 2024 Anhang A - Maßnahmen

■ Anhang A wurde ziemlich stark umgebaut

- Maßnahmenziele sind nicht mehr angegeben: „nur“ nach Controls
- Umgruppierung und Zusammenfassung alter Controls (93 Maßnahmen → 114 Controls)
- Gruppierung auf vier Gruppen anstatt 14 vorher
- 10 neue Controls

**A5**  
Organisatorische Maßnahmen  
(37 Maßnahmen)

**A6** Personenbezogene Maßnahmen (8)

**A7** Physische Maßnahmen (14)

**A8**  
Technologische Maßnahmen  
(34)

### A.5 Organisatorische Maßnahmen

5.12	Klassifizierung von Informationen	<b>Maßnahme</b> Informationen müssen entsprechend den Informationssicherheitserfordernissen der Organisation auf der Grundlage von Vertraulichkeit, Integrität, Verfügbarkeit und relevanten Anforderungen der interessierten Parteien klassifiziert werden.
------	-----------------------------------	---

→ Im LRZ drei Stufen: Öffentlich, Intern, Vertraulich (hier ist anzugeben, wer die Berechtigten sind)

→ Welche Daten brauche ich damit der Dienst verfügbar ist, Risikobetrachtung der Daten

→ Stakeholder: auch Partnerorganisationen (ggf. Eigentümer) berücksichtigen

### A.6 Personenbezogene Maßnahmen

6.7	Remote-Arbeit	<b>Maßnahme</b> Es müssen Sicherheitsmaßnahmen ergriffen werden, wenn Mitarbeiter aus der Ferne arbeiten, um Informationen zu schützen, die außerhalb der Räumlichkeiten des Unternehmens abgerufen, verarbeitet oder gespeichert werden.
-----	---------------	--

Richtlinie zum Umgang mit mobilen Geräten und Arbeit außerhalb des LRZ

- Datensparsamkeit und Verschlüsselung
- Schutz vor unberechtigtem Zugriff und Diebstahl
- Netzzugänge und sichere Verbindungen ins LRZ
- Nutzung von Programmen auf dienstl. Geräten
- Nutzung freigegebener Programme auf privaten Geräte

## 3. VORLESUNG

28.10.24 - ITS

### A.7 Physische Maßnahmen

7.14	Sichere Entsorgung oder Wiederverwendung von Geräten und Betriebsmitteln	<b>Maßnahme</b> Arten von Geräten und Betriebsmitteln, die Speichermedien enthalten, müssen überprüft werden, um sicherzustellen, dass jegliche sensiblen Daten und lizenzierte Software vor ihrer Entsorgung oder Wiederverwendung entfernt oder sicher überschrieben worden sind.
------	--	--

- > Richtlinie zur Weitergabe und Entsorgung von Datenträgern
  - Papier ⇒ Shredder (Sicherheitsstufe 4 nach DIN 66399)
  - Datenträger, Wechseldatenträger ⇒ verschrotten
  - Weiterverwendung ⇒ irreversibel überschreiben
- > Entsorgung Datenschutz- und Informationssicherheitskonform, z.B. zertifizierte Entsorgungsdienstleister
- > Entsorgung von Geräten: Zurücksetzen (Config) und alle lokalen PW löschen

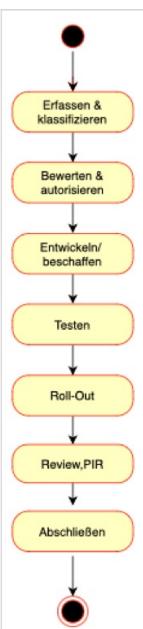
### A.8 Technische Maßnahmen

8.32	Änderungssteuerung	<b>Maßnahme</b> Änderungen an Informationsverarbeitungseinrichtungen und Informationssystemen müssen Gegenstand von Änderungsmanagementverfahren sein.
------	--------------------	---

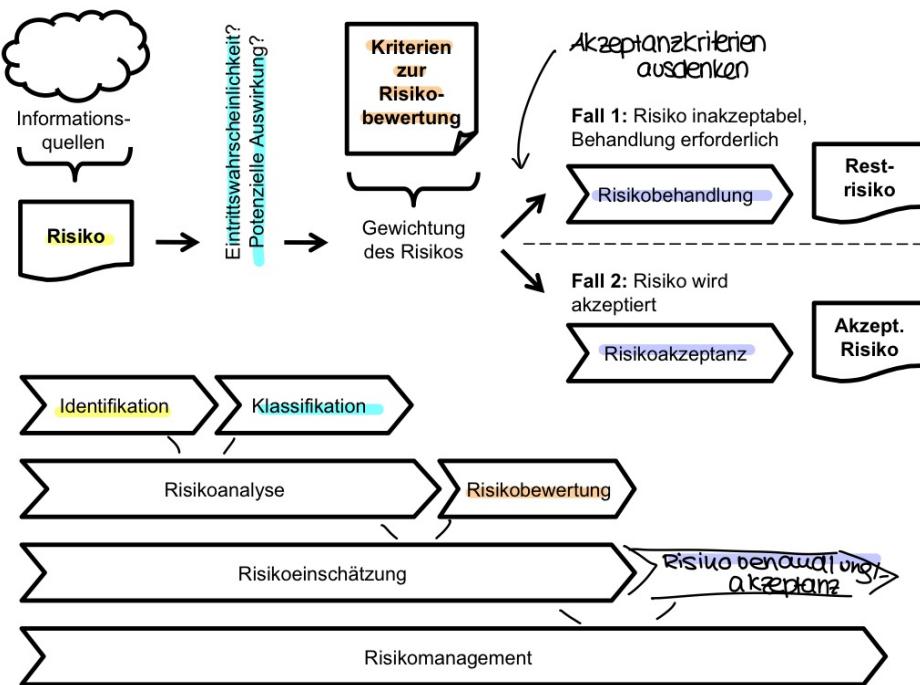
- > Rückgriff auf ISO/IEC 20000 - Service managementsystem
  - Change-Management Prozess und - Richtlinie

→ Risiken abmildern, wenn eine Änderung durchgeführt wird.

→ Wartungszeiten einhalten

S Suppliers / eingehende Schnittstellen	I Input	P Prozess	O Output	C Customers / ausgehende Schnittstellen
<b>ISRM</b> Incident- Service Request Management	Request for Change (RfC)		Informationen bzgl. Erfolg von Changes	<b>ISRM</b> Incident- Service Request Management
<b>SPM</b> Service Portfolio Management	Informationen zu geänderten Service Bedarfen (RfC)		Informationen bzgl. Erfolg von Changes	<b>SPM</b> Service Portfolio Management
<b>RDM</b> Release and Deployment Management	Informationen aus dem umgesetzten Release für das PIR		freigegebenes RfC zur Umsetzung des Release	<b>RDM</b> Release und Deployment
<b>PM</b> Problem Management	RfC		Informationen bzgl. Erfolg von Changes	<b>PM</b> Problem Management
<b>CAPM</b> , <b>SACM</b> Capacity/Availability/Continuity Management	RfC		Informationen bzgl. Erfolg von Changes	<b>CAPM</b> , <b>SACM</b> Capacity/Availability/Continuity Management
<b>CONF</b> Configuration Management	Status von Komponenten vor Umsetzung des Changes		Änderungen am Status von Komponenten bzw. neuer Status nach Ausrollen des Change.	<b>CONF</b> Configuration Management
<b>SUPPM</b> Supplier Management	RfC		Änderungen an Verträgen	<b>SUPPM</b> Supplier Management
				

## Grundlagen des Risikomanagements

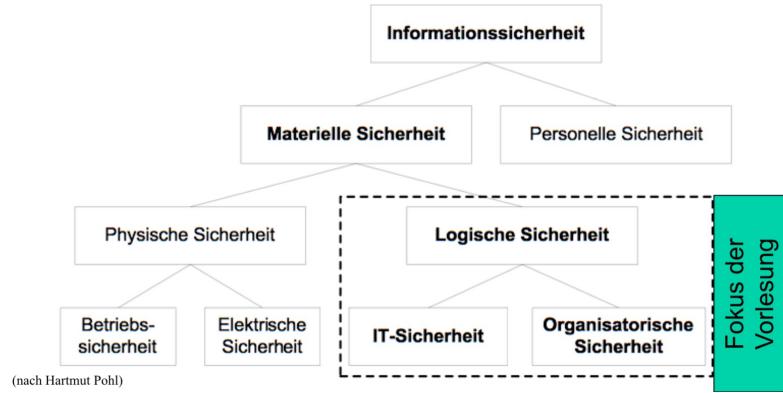
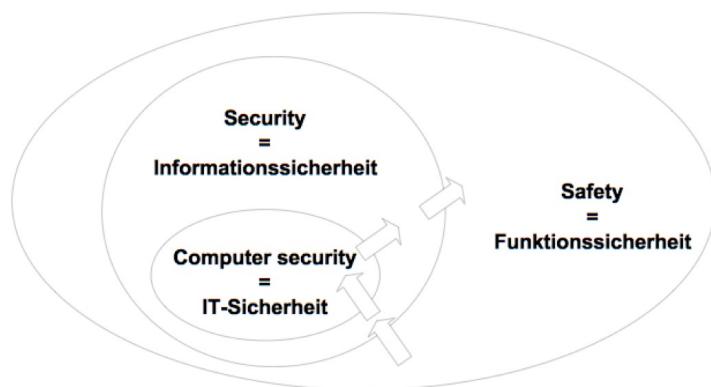


## Security & Safety

- > **Safety („Fktssicherheit“)**
  - Betriebssicherheit für sicherheitskrit. Programme
  - Ausfallsicherheit (Reliability)
  - Gesundheitsrel. Sicherheitseigensch. / Ergonomie
- > **Security („Sicherheit“)**
  - Hardware-/Software-/Netz-basierte Angriffe und Gegenmaßnahmen
  - Security Engineering: Design und Implementierung sicherer IT-Systeme
    - Security Policies: Sicherheitsanforderungen und deren Umsetzung
    - Anwendung von Kryptographie

# Safety vs. Security

Legende:



## KAPITEL 3: Technische Schwachstellen und Angriffe

### Handelnde Personen

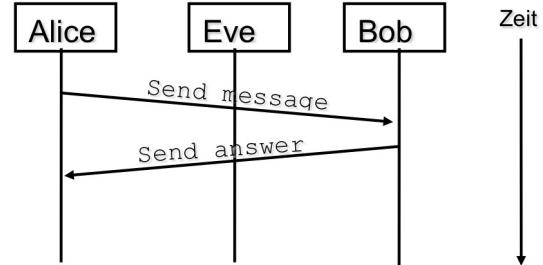
> Um Sicherheitsprobleme und -protokolle zu erläutern, werden häufig die folgenden Personen verwendet:

→ 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) Abhörer / passiver Angreifer
- Mallory, Mallet (M): (Malicious attacker) Aktiver Angreifer



### Angreifertypen

> Antwort auf: Was können / machen Eve, Mallory und Mallet?

> Angreifertypen umfasst insbesondere Angaben zu

→ Position des Angreifers: Innenräte; Besucher, Einbrecher; Internet / extern

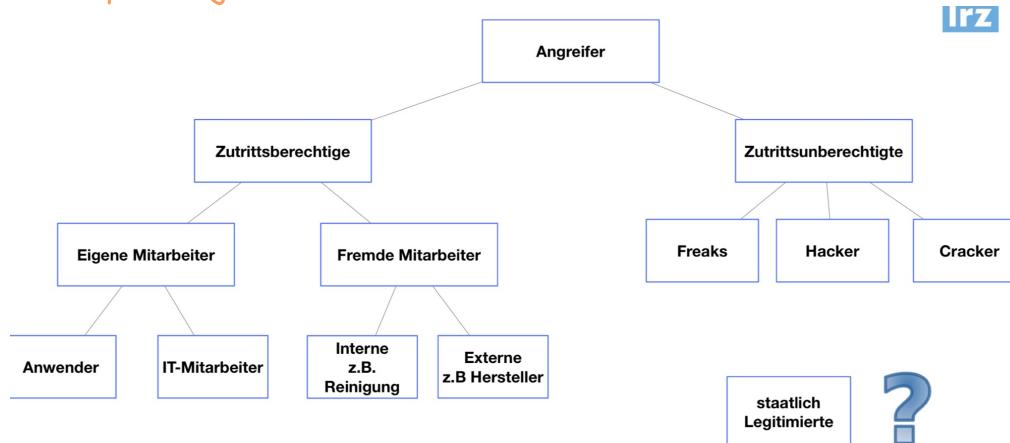
→ Fähigkeiten des Angreifers (= Wissen + finanzielle Mögl.), z.B. bei

- experimentierfreudigen Schülern und Studierenden
- Fachleuten mit prakt. Erfahrung
- erfahrenen Industriespionen / Geheimdiensten

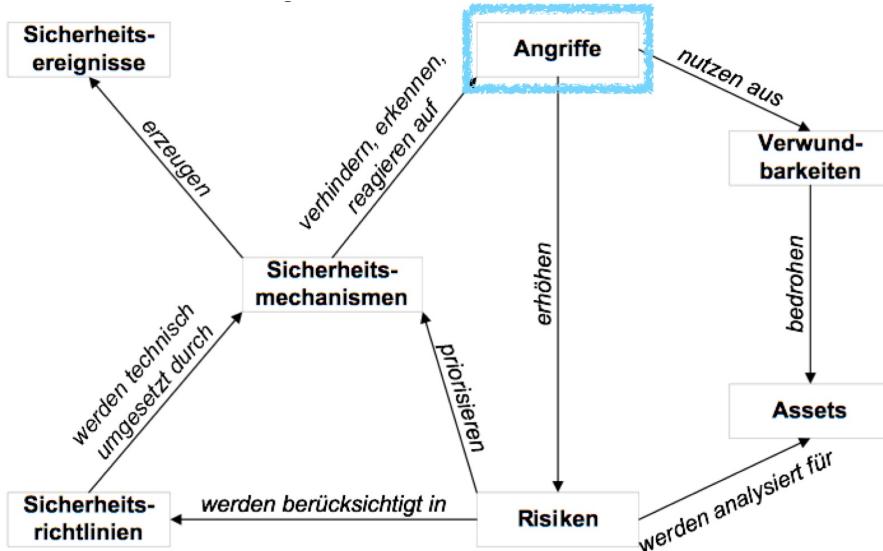
→ Motivation bzw. Zielsetzung des Angreifers, z.B. Geld, pol. o. rel. Fanatismus, Spielbetrieb

→ Spez. Charakteristika durchgef. Angriffe, z.B. passives Abhören vs. aktive Eingriffe

### Tätertypisierung



## Begriffe und Zusammenhänge



## Angriffsarten

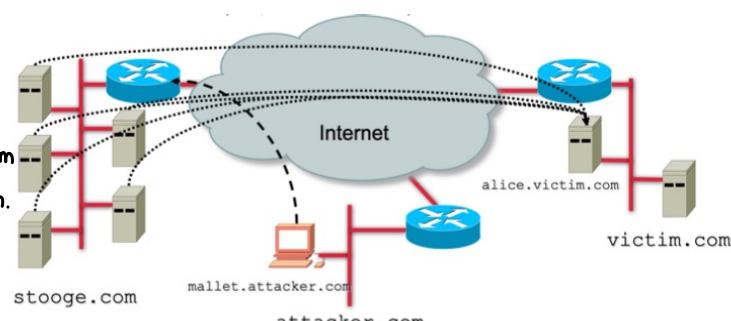
- > Erfolgreiche Angriffe haben negative Auswirkungen auf die
  - **Vertraulichkeit** (unberecht. Zugriff auf Daten) und /oder
  - **Integrität** (Modifikation von Daten) und /oder
  - **Verfügbarkeit** (Löschen von Daten, Stören von Diensten)
- > Eigenschaften zur Differenzierung von Angriffen sind, 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

## Denial of Service (DoS) und DDoS

- > Angriff versucht, das Zielsystem oder Netz für berechtigte Anwender unbenutzbare 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, ...)
  - Zerstörung oder Veränderung der Konfiguration
  - Physische Zerstörung oder Beschädigung
- > Bsp.: Überlasten eines Web-Servers durch massive Anfragen

## SMURF

- > 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 und IP-Broadcast am Router komplett deaktivieren
- > Besser:
  - Server so konfig., dass sie nicht auf Broadcast-Pings antworten
  - Router so konfig., 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 am Opfersystem

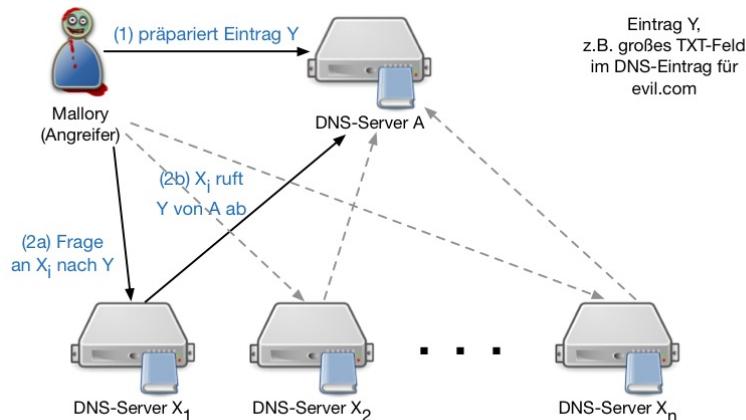
### > Grundprinzip:

- 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:

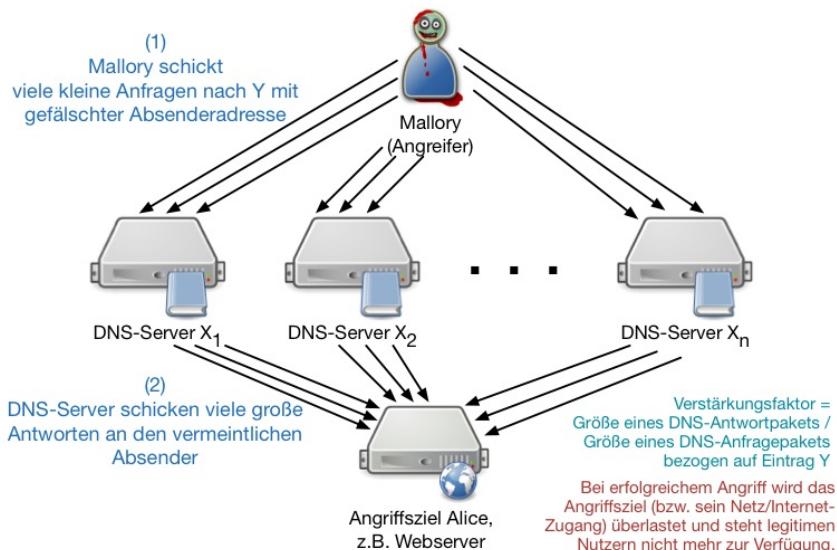
- Angriffe auf DNS-Root-Nameserver 2006
- Seit Frühjahr 2012 häufige Scans nach DNS-Servern, wachsende Anzahl an Vorfällen

## Vorbereitung



Ergebnis: DNS-Server X<sub>i</sub> haben Eintrag Y in ihrem Cache und liefern ihn auf Anfrage aus

## Ausführung



## Diskussion und Gegenmaßnahmen

> DNS server X<sub>n</sub> beantworten rekursive Anfragen aus dem Internet

> Ablauf:

- 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 X<sub>i</sub>
- X<sub>i</sub> fragt A und schreibt Ergebnis Y in seinen Cache
- Danach viele Anfragen nach Y an die Server X<sub>n</sub> mit gefälschter Absenderadresse von Alice

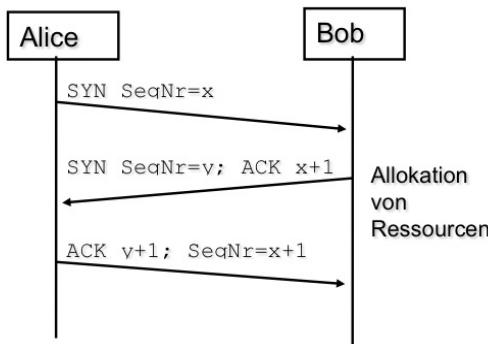
- Folge: Alice wird mit DNS-Antworten überflutet

### > Gegenmaßnahme:

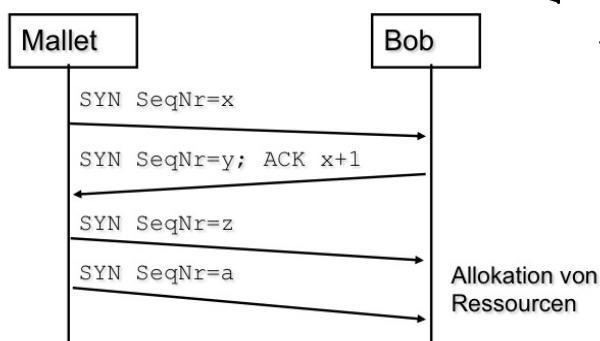
- Keine rekursiven Anfragen von extern beantworten
- [Schwellenwerte für ident. Anfragen desselben vermeintl. Clients]

## SYN Flooding

- ② TCP 3-Way-Handshake zum Verbindungsauftbau



- ② SYN Flooding

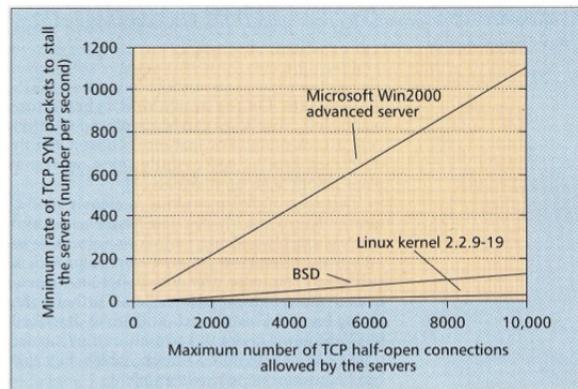


→ „Halboffene“ TCP-Verbind. so lange aufbauen, bis Ressourcen von Bob erschöpft

→ Bob kann dann keine weiteren Netzverbindungen mehr aufbauen

## Reaktion der Betriebssysteme

- ② Minimale Anzahl von SYN-Paketen für erfolgreichen DoS  
Quelle: [Chang 02]



- ② Wiederholung von „verlorenen“ SYN-Paketen:

- ② Exponential Backoff zur Berechnung der Wartezeit

- ② Linux und W2K  
(3s, 6s, 12s, 24s,...)

- ② BSD  
(6s, 24s, 48s,...)

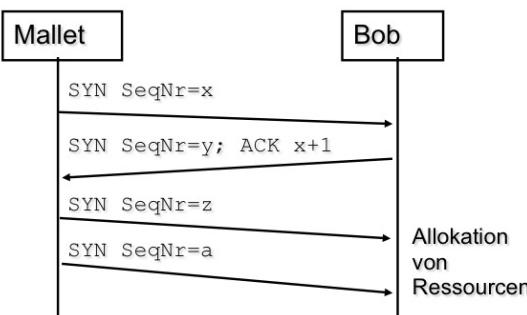
- ② Abbruch des Retransmit

- ② W2K  
nach 2 Versuchen (d.h. nach 9 Sekunden)

- ② Linux  
nach 7 Versuchen (d.h. nach 381 Sekunden)

- ② 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

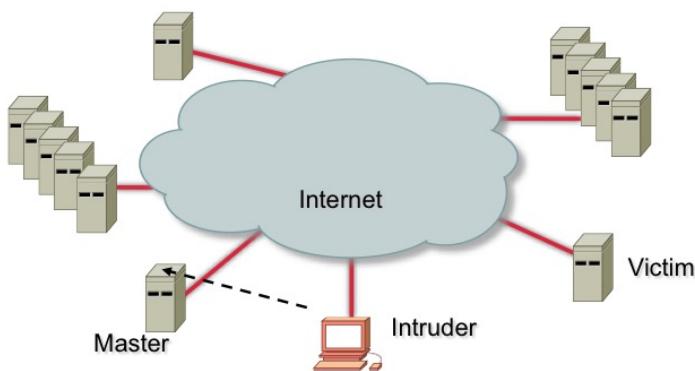
> SYN Cookies (Bernstein 1996):

Seq-Nr. y von Bob „kodiert“ Adressinfo von Mallet. Ressourcen werden erst reserviert, wenn tatsächliches ACK y+1 von Mallet eingent.

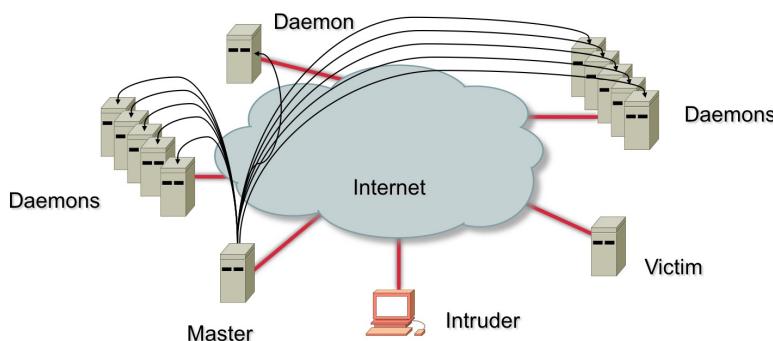
## Grundsätzlicher Ablauf - Botnet

> Dreistufig. Verfahren:

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

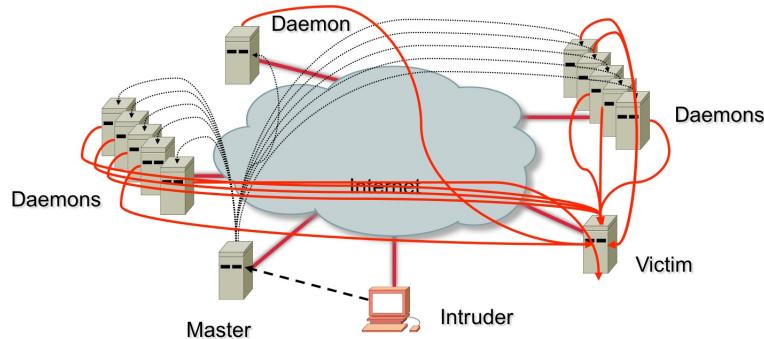


2. Master versucht automatisiert, weitere Maschinen zu komromittieren, um DDOS-Software (Daemon) zu installieren, bzw. schiebt anderen Nutzern Malware unter.



3. Intruder startet Programm auf Master, das allen Daemonen mitteilt, wann und gegen wen der Angriff zu starten ist.

Zum vereinbarten Zeitpunkt startet jeder Daemon Das-Angriff



### Mirai Botnet

- ❑ 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](https://shodan.io) Suchmaschine für IoT
- ❑ Gegenmaßnahmen:
  - ❑ Filtern des Mirai Infektionscode mit IDS
  - ❑ Patchen der Schwachstellen
  - ❑ Abschotten der Geräte, bzw. des Zugangs zum Internet

# 4. VORLESUNG

04.11.24 - ITS

## 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 Konfig.-Anpassungen
- > 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
- > Allg. Ansätze:
  - Anzahl Verbindungen und Datenvolumen überwachen (Anomalieerkennung)
  - Bug- und Sicherheitswarnungen (z.B. CERT) verfolgen

### Beispiel

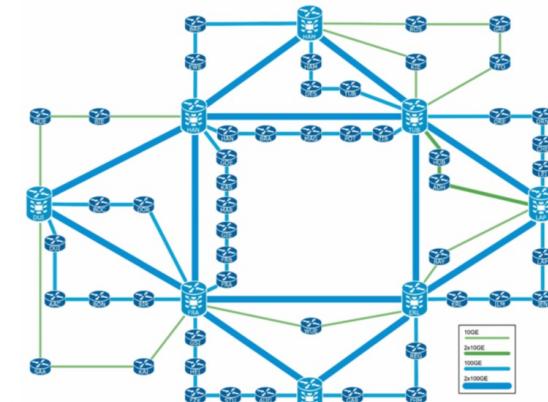
#### Erpressungsversuch mit DDoS-Drohung

Betreff: DDOS www.zhs-muenchen.de  
 Datum: Mon, 5 Sep 2011 02:50:02 -0600  
 Von: samiliaivspopek@yahoo.com  
 An: <hostmaster@rz.de>  
 Your site [www.zhs-muenchen.de](http://www.zhs-muenchen.de) will be subjected to DDoS attacks 100 Gbit/s.  
 Pay 100 btc(bitcoin) on the account 17Ra8qjGLisGz1RaUqdA2YHgspdK01r  
 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

Irz

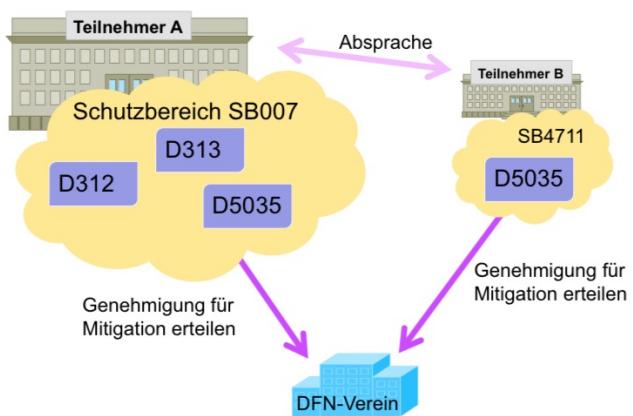
## DFN: Dt. Forschungsnetz Verein e.V.



## DFN DDoS Mitigation Dienst

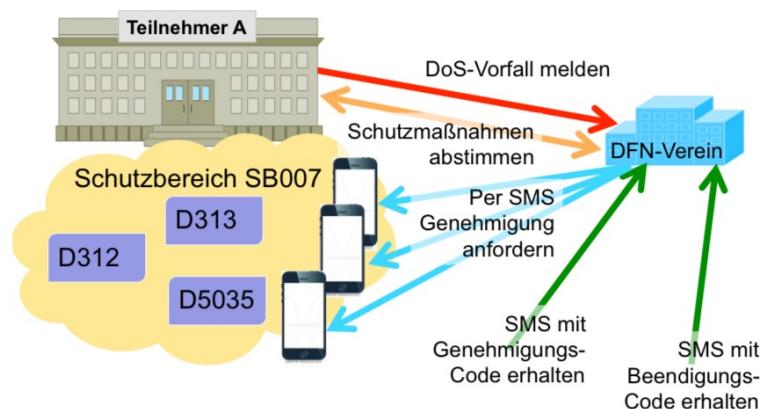
### Registrierungsprozess

DFN  
Deutsches  
Forschungsnetz



### Genehmigungsprozess

DFN  
Deutsches  
Forschungsnetz

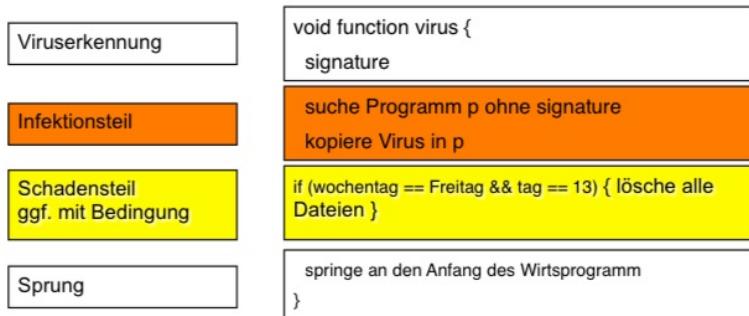


## Schadsoftware

### Malicious Code - Virus

- > Def.:
  - Befehlsfolge; benötigt Wirtsprogramm zur Ausführung
  - Kein selbstständig ablauffähiges Programm
  - Selbstreplikation (Infection weiterer Wirte (Programme))

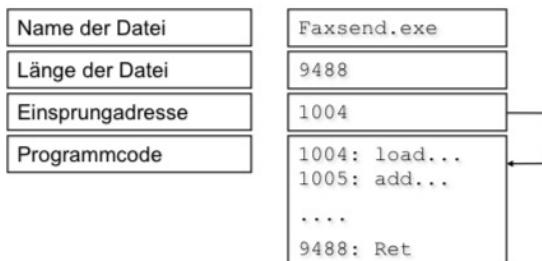
## > Allg. Aufbau.



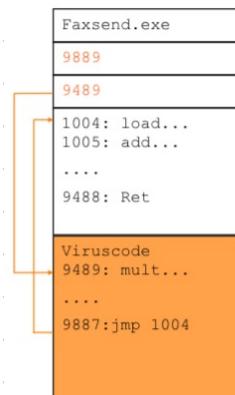
→ Daneben ggf. Tammingsteil (selbstentschlüss. Code, Padding,...)

## Programm - Viren - Infektion

- > Dateiformat vor der Infektion  
(vereinf. Bsp.)



- > Datei nach der Infektion



## Manipulierte Virensignaturen

- > Zwei Haupt - Angriffsvektoren:
  - Angreifer bringen bekannte Viren-Signaturen in harmlosen Dateien unter und lassen diese über Online-Virenscanner 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

## Malicious Code - Wurm

- > Def.:
  - Eigenständig lauffähiges Programm - benötigt keinen Wirt!
  - Selbstreplikation (z.B. über Netz oder USB-Sticks (mit "Autorun"))
  - Einzelne infizierte Maschinen werden als Wurm - Segmente bezeichnet
- > Bsp.e.:
  - Internet-Wurm (1988)
  - ILOVEYOU (2000)
  - Code Red (2001; Defacement von Microsoft IIS Webserven)

## Malicious Code - Trojanisches Pferd

- > Def.:
 

Ein Programm, dessen Ist - Fkt. nicht mit der angegebenen Soll - Fkt. übereinstimmt:

  - Sinnvolle o. alt. „Nutzfkt.“
  - Versteckte (schad.) Fkt.
  - Keine selbstständige Vermehrung
- > Bsp.: Unix Shell Script Trojan [Stoll 89]

## „Staatstrojaner“

- > Chaos Computer Club (CCC) analysiert zugespielte DLL: mfc42ul.dll
  - Wird per Registry - Eintrag geladen

- Klinkt sich bei der Initialisierung in explorer.exe ein

> Fkt.:

- Screenshots
- Abhören von Skype- und VoIP-Gesprächen
- Nachladen weiterer Module
- Kommunikation mit Command and Control (C&C) Server

## „Staatstjaner“ - Analyse

Kommunikation:

- > Einseitig verschlüsselt zwischen Malware und C&C-Server
- > Mit AES-ECB (Electronic Code Book Mode)
  - Jeder Block wird mit dem ident. Schlüssel verschlüsselt, d.h. gleiche Klartextblöcke ergeben ident. Chiffre-Blöcke
  - Schlüssel in allen Varianten identisch
- > „Authentisierung“ über konst. 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 Nachladefkt. 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

## „Staatstjaner“ - Befehlssatz C&C

- ? 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

## „Staatstjaner“ - Staatstjaner 2021

- > 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-späh-software, Fa. NSO, Israel) überwacht
  - Sept. 21: Bundeskriminalität soll Pegasus gekauft haben
  - Kleinerei: Auskunft wegen staatswohlbegründeten Geheimhaltungsinteressen

## Malicious Code - 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:
  - Fkt. 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)

## Malicious Code / Troj. Pferde - weitere Formen

> 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

## Ransomware

- > Krypto-Erpressungstojaner
- > Malware verschlüsselt Dateisystem und verlangt „Lösegeld“
- > WannaCry (Mai 2017)

→ Uni

## Ransomware - Schutz

- > Updates und Patches installieren
- > Backups anlegen
  - andere Medien (Bänder)
  - Dateisysteme, Netzaufwerke nicht dauernd angebunden lassen
- > Schutzsoftware (Virenscanner) installieren
- > „Nur E-mails und Anhänge von bekannten Absendern öffnen“
  - Absender können sehr einfach gefälscht werden
  - Rechner des Absenders kann komromittiert sein
  - Ggf. über anderen Kanal beim Absender nachfragen

## E-mail - Security (Spam)

### Spam E-mail

## Phishing

### Arten von Phishing

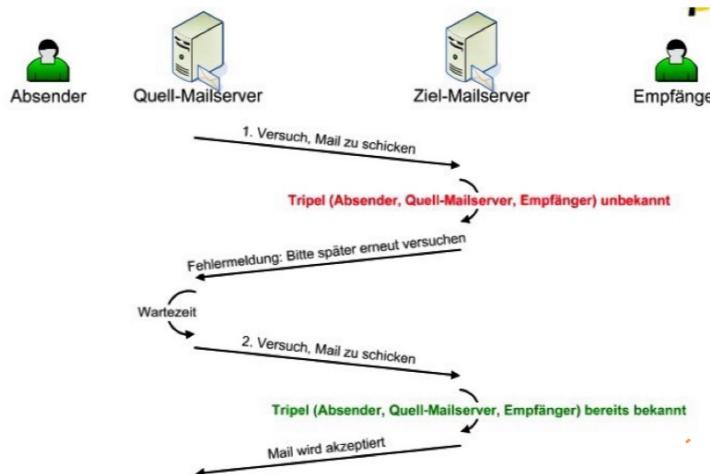
- > Clone-phishing ("Update" echter E-Mails)
- > Spear-phishing (personalisiertes, zielgerichtetes Phishing)
- > Whaling (Phishing, z.B. gegen hochrangigen Mitarbeiter)
- > CEO Fraud (Manipulation zur Überweisung von Geld)
- > IBAN Fraud (manipulierte PDF-Rechnung mit geänderter Kontoverbindung)
- > Vishing (Voice Phishing; Ziel: Opfer ruft Angreifer an)

## Spam - 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
    3. Filtersoftware lernt aus Beispielen
  - > 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 Erkennungsraten abhängig von Training (NN) bzw. Modellierung (Bayes).

Spamfilter

## Greylisting gegen Spam (1/2)



## Systemname Angriffe (Buffer Overflows, Backdoors, Rootkits,...)

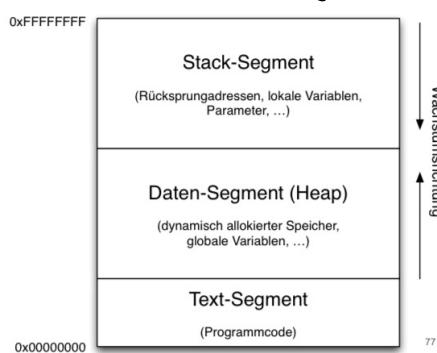
## Exploits : Buffer Overflow - Hier: stack smashing

- > Ziel: Ausführen von Code auf fremden Rechnern unter fremden Rechten (z.B. root)

## > Vorgehen:

- Auswahl des Ziels: Lokal, Remote
  - Überschreiten interner Programmpeffer, z.B. durch überlange Eingabe
  - Dabei Manipulationen

## > Speicherabbild eines Programms:



## Ausnutzen von Buffer Overflows in Stack-Segmenten

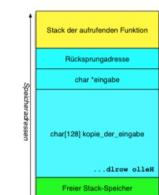
Ziel: Stack gezielt überschreiben, so dass

- > Rücksprungadresse auf Angreifer-Code umgebogen wird
  - > Angreifer-Code das System komromittiert (z.B. Starten einer interaktiven Shell oder Nachladen beliebiger Schadprogramme)

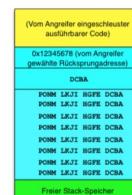


Anmerkung: Darstellung des Stack-Aufbaus vereinfacht!

IT-Sicherheit I WS 24/25 | © Helmut Reiser



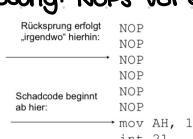
Stack bei



Stack bei

## Kleinere Hürden beim Stack-Smashing

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



- > Der 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.  $\Rightarrow$  Lösung: Fuzzing

## Stack-Smashing - Schutz

- > Am Besten: Sicheres Programmieren, z.B. strncpy statt strcpy
  - Unterstützung durch Code-Analyse-Tools, z.B. Splint
- > Stack-Guarding:
  - Beim Aufruf einer Funktion wird hinter der Rücksprungadresse ein Kontrollzeichen („Canary“) angelegt
  - Vor dem Rückprung wird geprüft, ob das Kontrollzeichen noch intakt ist.
  - Variante: Mehrere Kopien der Rücksprungadresse.
- > Nicht-ausführbare Stacks (non-executable stack)
  - Code auf dem Stack wird bis 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.

## Buffer-Overflows - Weitere Aspekte

- > Heap Corruption
- > Problematisch sind nicht nur String-Operationen
- > Format String Attacks

## Systemnahe Angriffe - Account/Password Cracking

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

### Systemnahe Angriffe

#### 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 aktivierte Netzdienste
  - Suche nach ungewöhnlichen SUID/SIGID-Programmen
- Reaktion bei erkannten Hintertüren:
  - Vollständiges Entfernen der Schadsoftware möglich?
  - Ggf. Maschine neu bzw. aus „sauberem“ Backup aufsetzen.
  - Verwundbarkeit, die zur Kompromittierung geführt hat, muss behoben werden!

### Systemnahe Angriffe

#### Rootkits

- 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

#### 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
  - Hebt 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)

### Systemnahe Angriffe

#### Rootkits (Forts.)

- Rootkits der 2. Generation
  - Motivation: Alle Systemprogramme einzeln auszutauschen ist aus Angreifersicht aufwendig und fehleranfällig.
  - Neue 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

### RSA Security Hack

#### Einfallstor Adobe Flash

- 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.
- Eingebeutes SWF-File nutzt Adobe-Flash-Player-Lücke aus.
- Schadcode (Abwandlung von „poison ivy“) späht Mitarbeiterrechner aus und überträgt u.a. Passwörter an den Angreifer.
- Folgen:
  - SecurID-Quellen und -Seeds werden ausgespäht
  - US-Rüstungsunternehmen Lockheed Martin wird mit „nachgebauten“ SecurID-Tokens gehackt; zahlreiche weitere Unternehmen betroffen
  - Rund 40 Millionen SecurID-Token werden ausgetauscht

## Web-basierte Angriffe (XSS,...)

### Cross Site Scripting

- > Einbetten von Schadcode in (vertrauenswürdigen) anderen Code
- > Typisches Ziel bei XSS:
  - sensible Daten, z.B. Cookies, an den Angreifer übertragen
  - Identitätsdiebstahl, impersonation attack

### XSS - Grundproblem

- > Anwendung prüft Benutzereingaben nicht ausreichend

### DOM-basiertes (lokales) XSS

- > Lokal bedeutet hier: Ohne Beteiligung eines webservers
- > Auslöser: JavaScript-Fkt. prüft übergebene Parameter nicht

- Beispiel:

```
<HTML>
<TITLE>HTML-Beispieldokument DOM-XSS</TITLE>
Hello
<SCRIPT>
var pos=document.URL.indexOf("username")+9;
document.write(document.URL.substring(pos,document.URL.length));
</SCRIPT>
<BR/>
Dies ist ein Beispiel-HTML-Dokument.
</HTML>
```

> Als Parameter übergebener Code wird von anfälligen Browsern ausgeführt.

- Aufruf mit:

[http://www.example.com/index.html?username=<script>alert\('XSS-Problem!'\)</script>](http://www.example.com/index.html?username=<script>alert('XSS-Problem!')</script>)

### Reflexives (nicht-persistentes) XSS

- > Ablauf:

- Webserver liefert Webseite mit Inhalt aus, der vom Benutzer übergebene (und somit nicht-persistente) Parameter (inkl. JavaScript-Code) enthält.

### Persistentes (stored) XSS

- > Schadcode wird vom Webserver gespeichert und bei jeder Anfrage ausgeliefert.
- > Dadurch sehr breit gestreuter Angriff
- > Besonders problematisch, wenn der „verseuchte“ Webserver als besonders vertrauenswürdig konfiguriert ist.
- > Gegenmaßnahmen:
  - Webapplikation muss Script-Code aus Benutzereingaben entfernen oder „ungefährlich“ machen.
  - Script-Code kann anhand der Meta-Zeichen, z.B. <, erkannt werden.
  - Client-seitig: JavaScript deaktivieren oder Plugins wie Noscript verwenden.
  - Content-Security-Policy-Spezifikation vertrauenswürdiger Script-Quellen, alles andere wird nicht ausgeführt

## Web server security

### SQL Injection

#### Erwarteter Aufruf

<http://webserver/cgi-bin/find.cgi?ID=42>

#### Erzeugte SQL-Abfrage

SELECT autor, text FROM artikel WHERE ID=42

#### Aufruf mit SQL-Injektion

<http://webserver/cgi-bin/find.cgi?ID=42;UPDATE+USER+SET+TYPE='admin'+WHERE+ID=2>

#### Erzeugte SQL-Abfrage: 2 Befehle!

SELECT autor, text FROM artikel WHERE ID=42; UPDATE  
USER SET TYPE='admin' WHERE ID=2

## Netzbasierte Angriffe (Sniffing, Portscans,...)

### Sniffer: Abhören des Netzwerks

- > Local Area Network (LAN):

- Oft gemeinsam genutztes Medium (shared medium), z.B. WLAN
- Netzwerk-Karten können im Prinzip gesamten Verkehr mit hören, aber
- geben nur die an den Rechner adressierten Pakete weiter
- Gefahr: "Promiscuous Mode"

- > Wide Area Network (WAN):

- Jeder Vermittlungsrechner kann Nachrichten „mitlesen“, z.B. Mirror-Ports an Routern
- „Anzapfen“ von Leitungen (z.B. durch Geheimdienste)

## > Tools:

- Übergang zw. Werkzeug des System- sowie Netzadministrators und Cracker-Tool sind fließend
- tcpdump, ngrep: Standard-Werkzeuge in vielen UNIX-/Linux- Distributionen
- wireshark: Packet-Analyser (Linux, Windows)

## Port-Scanner

> Suchen auf entferntem Rechner nach „offenen“ Ports

- Versuch eines Verbindungsaufbaus / pro Port
- Falls erfolgreich: Port ist „offen“

> Damit Identifikation von Diensten

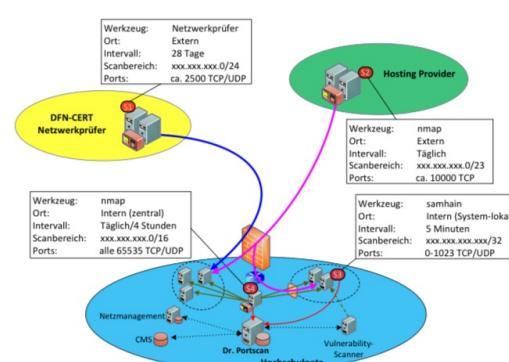
> Gezielte Suche nach Rechnern, die Dienste mit bekannten Schwächen anbieten

> Auch hier ist der Übergang zwischen nützlichem Werkzeug und Cracker Tool fließend

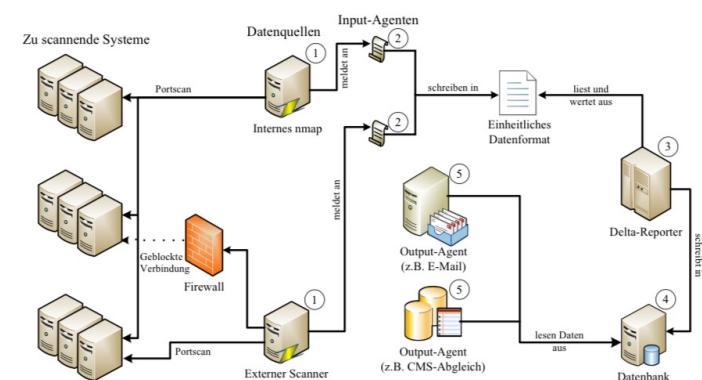
> Port-Scans werden oft als Angriff gewertet und deshalb getarnt durchgeführt

> Bsp.: nmap

## Proaktive Netzüberwachung mit Portscans



## Auswertung von Portscan-Ergebnissen



## 3. Bewertung von Schwachstellen

### Common Vulnerability Scoring System (CVSS)

#### Motivation für schwachstellenbewertung

- > Hauptziel: Priorisierung
- > Betrifft sowohl die Entwickler als auch die Betreiber von Software / Systemen
- > Idee: Quantitative Bewertung von Schwachstellen anhand einer def. Menge verschiedener Charakteristika  
⇒ Jeder Schwachstelle wird ein Zahlenwert zugeordnet.
- Problem: Objektivität / Einheitlichkeit
- > CVSS-Ansatz:
- Dreiteilung in unveränderliche bzw. zeitlich und räumlich variable Charakteristika

### Common Vulnerability Scoring System v4

- > CVSS ist an der CNU entstanden und wird inzwischen von FIRST (Forum of Incident Response and Security Teams) gepflegt.
- > Vier Gruppen von Bewertungsmerkmale:
  - Base Metrics: Grundlegende Eigenschaften der Verwundbarkeit
  - Threat Metric: Umsetzung - Proof-of-Concept Code oder aktive Exploits
  - Environmental Metrics: Anwenderspez. Eigenschaften d. Verwundbarkeit
  - Supplemental Metrics: Extrinsische Attribute um den Kontext zu berücksichtigen

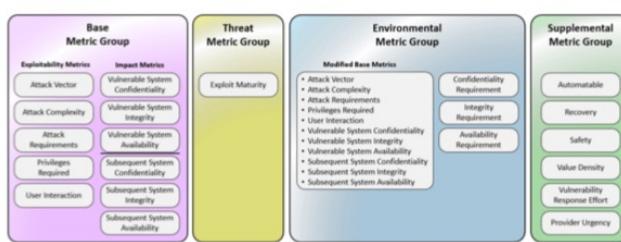


Figure 1: CVSS Metric Groups

- > Base Metrics werden oft von Herstellern / Sicherheitsunternehmen veröffentlicht

## CVSS Metrics - Grundidee

- > Input: Einfache Bewertung von Schwachstellen durch vorgeg. Fragen und Antwortmögls.
- > Outputs:
  - CVSS-Score (= Zahl) zwischen 0,0 (harmlos) und 10,0 (katastrophe)
  - CVSS-Vektor = kompakter String, Kurzfassung des gesamten Inputs
- > Scoring-Formel

## CVSS v4 - Base Metrics

- > Die Base Metrics bewerten die intrinsischen, konstanten Eigenschaften von Schwachstellen; drei Unterguppen: Exploitability metrics, Impact metrics und Scope.
- > Exploitability metrics (techn. Aspekte der Schwachstellenausnutzung):
  - Attack vector: physisch, auf dem Rechner, vom LAN aus, via Internet?
  - Attack complexity: Trivial, anspruchsvoll?
  - Attack Requirements: Spezielle Umgebungs- oder Ausführungsbedingungen nötig?
  - Privileges required: Jeder, reg. User, Admin?
  - User interaction: Angreifer braucht User?
- > Impact metrics (Auswirkung der Schwachstelle):
  - Auswirkungen auf C,I,A für verwundbares und Nachfolgesysteme
  - Jeweils gar nicht, gering oder stark?

## CVSS v4 - Threat Metric Group

- > Die Threat Metrics bewerten den jeweils (zeitlich) aktuellen und damit variablen Stand der Schwachstelle
- > Exploit code maturity?
  - Not defined: nimmt den Worst Case an
  - Attacked: Erfolgreiche Angriffe bekannt
  - Proof of concept: PoC Code ist öffentlich
  - Unreported: Keine PoC bekannt

## CVSS v4 - Environmental Metrics

- > Die Env. Metrics bewerten die Schwachstelle im Hinblick auf das Einsatzgebiet des betroffenen Systems; sie unterscheiden sich also z.B. je nach Organisation
- > C,I,A Requirements?
  - Undefined, oder:
  - Abschaltung: Gering; mittel; hoch
- > Modified Base Metrics?
  - Falls Sicherheitsmaßnahmen im Einsatz sind, die sich auf die Base-Metric-Eigensch. der Schwachstelle auswirken, können diese individuell modifiziert werden.
  - Default: not defined

## CVSS v4

### Supplemental Metric Group

- Die Supplement Metrics sind optional und bewerten zusätzliche externe Faktoren.
- Automatable? Kann der Angreifer automatisiert mehrere Ziele angreifen?
- Recovery? Resilienz des Systems. Wie schnell kommt man zu stabilem Zustand?
- Safety? Impact des Angriffs auf Safety?
- Value Density? „Dichte“ des Angriffs - Angriffsziel hat wenig Ressourcen (z.B. einzelner Email-Client) oder sehr viele (z.B. zentraler Email Server).
- Vulnerability Response Effort? Wie hoch ist der Aufwand um auf den Angriff zu reagieren?
- Provider Urgency? Provider liefern zusätzliche Einschätzungen des Schweregrades?



## CVSS v4

### Zusammenspiel der Gruppen

- CVSS v3 Score:
  - Man muss Base Score, Temporal Score und Environmental Score unterscheiden.
  - "CVSS Score" ist oft Synonym für CVSS Base Score - nur der ist verpflichtend
- Bei CVSS v2 galt:
  - Base Score >= Temporal Score >= Environmental Score
  - Beispiele:
    - Praxisrelevanter Score ist niedriger, wenn offizieller Fix verfügbar ist.
    - Praxisrelevanter Score ist niedriger, wenn CIA Requirements niedrig sind.
- Ab CVSS v3 gilt o.g. Ungleichung nicht mehr zwingend wg. optionaler Modified Base Metrics...
- CVSS v4 Threat Metric Group realisiert Zeitbezug (Temporal Score)
- CVSS v4 Supplemental Metric Group bilden Kontext und äußere Attribute ab

- CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:X/RL:X/RC:X/CR:X/IR:X/X/AR:X/MAV:X/MAC:X/MPR:X/MUI:X/MS:X/MC:X/MI:X/MA:X
- Cisco: CVSS v3 Base Score: 8.8

#### Base Parameters

Once discovered, analyzed, and catalogued, there are certain aspects of a vulnerability that do not change, assuming the initial information is complete and correct. These immutable characteristics will not change over time, nor in different environments. The base metric group captures the access to and impact on the target.

AV: Adjacent	AC: Low	Base Score:
PR: None	UI: None	
S: Unchanged	C: High	
I: High	A: High	

8.8



Beispiel:

<https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20181101-sap>

## zero Day Exploits

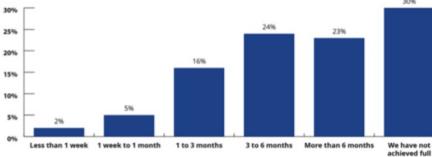
#### Analyse von „zero-day“ Exploits

- "Before we knew it - An empirical study of zero-day attacks in the real world", Bilge/Dumitras, Oktober 2012  
[http://users.ece.cmu.edu/~tdumitra/public\\_documents/bilge12\\_zero\\_day.pdf](http://users.ece.cmu.edu/~tdumitra/public_documents/bilge12_zero_day.pdf)
- Wie lange werden Sicherheitslücken ausgenutzt, bevor sie allgemein bekannt (und beseitigt) werden?
  - Untersuchung für 11 Millionen Windows-PCs mit Symantec-Software
  - Dauer schwankt zwischen 19 Tagen und 30 Monaten
  - Durchschnitt liegt bei 312 Tagen (!)
- Google Project Zero
  - 0Day „In the Wild“, Angriffe auf Basis von Zero Day Exploits seit 15.07.14  
<https://googleprojectzero.blogspot.com/p/0day.html>
  - Spreadsheet mit „Date discovered“, „Date Patched“ (Patch Available) zwischen 1 und 165 Tagen

#### Analyse von „zero-day“ Exploits

- Ponemon Institut: State of Endpoint Security Risk -10.2018  
<https://cdn2.hubspot.net/hubfs/468115/whitepapers/state-of-endpoint-security-2018.pdf>

Figure 17. How long did it take your organization to achieve full adoption of your EDR?



- Wie wirkt sich die Veröffentlichung einer Sicherheitslücke aus?
  - Anzahl an Malware-Varianten steigt um das bis zu 85.000-fache
  - Anzahl beobachteter Angriffe steigt um das bis zu 100.000-fache

## Zusammenfassung

- Angreifermodelle beschreiben Fähigkeiten, Motivation usw.
- Angriffe zielen darauf ab, den individuellen Schutzbedarf (Vertraulichkeit, Integrität, Verfügbarkeit) zu verletzen:
  - Malware-/Rootkit-infizierte Systeme bieten keine Vertraulichkeit mehr
  - Buffer Overflow Exploits zerstören die Integrität von Software
  - DoS-Angriffe stören die Verfügbarkeit
  - ...
- Zu jedem Angriff gibt es mehr oder weniger effektive / kostspielige Gegenmaßnahmen:
  - Ziel ist aber kein Flickenwerk aus einzelnen Maßnahmen, sondern ein von Grund auf sicheres Design (=> Security Engineering).
  - Kenntnis von Angriffsvarianten und -wegen ist Voraussetzung für die Konzeption adäquater Sicherheitsmaßnahmen.

# KAPITEL 4

## Social Engineering - der Faktor Mensch in der IT-Sicherheit

### 1. Social Engineering - Begriffsbildung und -abgrenzung

#### Begriffsbildung und Abgrenzung

- > 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 Nutzfkt.tät
  - Schockanrufe - moderner Enkelttrick

#### Social Engineering - Fkt. weise

- > Ausnutzung menschl. Eigenschaften oder Gefühle, u.a.: Hilfsbereitschaft, Vertrauen, Angst, Respekt vor Autorität, Schutzbedürfnis, Neugierde, Faulheit, ...
- > Jede menschl. Schwäche kann ausgenutzt werden
- > Social Engineering gibt es immer und überall: Eltern, Lehrer, Freunde, Werbung, gesell. Normen
- > Bei IT-Sec wird oft primär an Technik gedacht, aber zu wenig an den "Faktor Mensch"

### 2. Angreifer-Perspektive:

#### Ausgewählte Bsp.

### Kategorisierung und Arten von Social-Engineering-Angriffen

#### Social Engineering - Kategorisierung

- > Grundlegend zu unterscheiden:
  - Passive Angriffe (keine Interaktion mit dem Opfer), u.a. Belauschen von Gesprächen, shoulder surfing, dumpster diving, baiting
  - Aktive Angriffe, u.a.: pretexting, phishing
- > etablierte Kategorien:
  - Human-based Social Engineering (ohne techn. Hilfsmitteln)
  - Computer-based Social Engineering (mit techn. Hilfsmitteln)
  - [Reverse Social Engineering] (Opfer wendet sich freiwillig an Angreifer)

#### Kategorie Human-based Social Engineering

- > Dumpster Diving: Klausurenwürfe in der Papiertonne?
- > Shoulder Surfing: Notebook-Nutzung im Hörsaal?
- > Tailgating
- > Badge Surveillance: Selbstgedruckte Mitarbeiterausweise?
- > Pretexting
- > Quid pro quo: Schokolade für Hausaufgabenblätter?
- > People Watching
- > Diversion Theft

#### Kategorie Computer-based Social Engineering

- > Phishing:
  - Clone phishing, Spear phishing (personal), Whaling, CEO Fraud, Vishing, Evil Twins
- > Baiting: im Hörsaal verlorener USB-Stick
- > Forensic analysis: "Dumpster diving" für Elektronik
- > Electronic badges: Duplizieren elektr. Schlüssel

## Typische Eigenschaften von erfolgreichen Social Engineers

- > Können gut mit Menschen kommunizieren
- > Sind geduldige Schauspieler
- > Sind sich nicht zu gut

### 3. Anwender-Perspektive

#### Gegenmaßnahmen für Social-Engineering-Angriffe

##### Gegenmaßnahmen beim Pretexting (sich für jemand anderen ausgeben)

- > Gesundes Haß an Misstrauen und Vorsicht
- > Gespräch beenden und zurückrufen oder Kanal wechseln
- > Bei Zweifeln Kollegen zu Rate ziehen

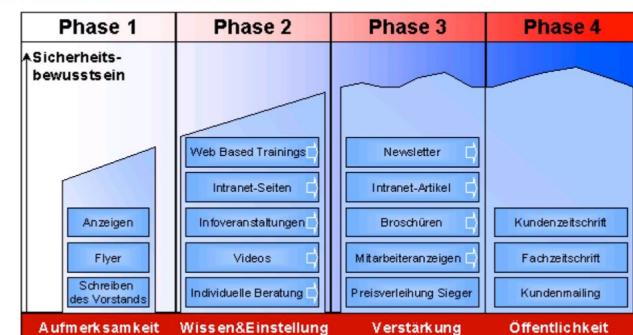
##### Gegenmaßnahmen

- > Gutes Social Engineering fkt. immer
- > Bsp. Maßnahmen:
  - Technisch:
    - Dumpster Diving: Aktenvernichtung / Papierkörben abschließen
    - Shoulder Surfing: Sichtschutzfolien für Notebook-Displays
    - Tailgating: Wachdienst, 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

##### Awareness-Maßnahmen - Planung

- > Wie alles rund um IT-Sicherheit auch eine Budgetfrage
- > Organis. Randbedingungen
- > Komb. versch. Ansätze

Vier-Phasen-Modell nach Fox/Kaun



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

#### Durchführung von Social Engineering Penetration Tests

##### Penetrationstests - Grundidee

- > PENTESTS (allg.) 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: "Innenräte")

##### Ablauf

- > SE-Pentest = Projekt mit 5 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 Misserfolgen oder im Anschluss demotivieren
- Keine Beschädigungen, z.B.
  - keine Gewaltanwendung (Fenster einschlagen,...)
  - kein Entwenden von Gegenständen

## Planung / Zielsetzung

- > Festlegung des Testumfangs:
  - Beratung
  - Budget- und Ethikrandbedingungen, Ziele und Deliverables
  - Testzeitraum und -orte
  - Werkzeugwahl, z.B. Telefon; Vorabinformationen
- > Vertragliche Regelungen:
  - Dienstleistungsvertrag
  - (Mind. 2) Ansprechpartner und "Get out of jail free"-Karten für Notfälle
  - Schriftl. Erlaubnis zur Dokumentenfälschung, ...
  - Art der Erfolgsnachweise: Videos / Fotos zulässig?
  - Berichtsmodalitäten, z.B. wöchentlich

## Informationsakquise

- > Per Internet (OSINT):
  - Organigramme
  - Jahresberichte, Stellenanzeigen, ...
  - Mitarbeiternamen mit E-Mail-Adressen und Tel.nummern
  - Aktuelle Projekte, Produkte, Presseerklärungen, Kunden, ...
  - Jargon
  - Beiträge in Diskussions- / Support- Webforen mit Firmen-E-Mailadresse
  - Ggf. Social-Network-Profile des Personals
- > Vor Ort:
  - Personal: Typ. Kleidung, Ausweise, ...
  - Gebäude: Raumpläne, überwachte Bereiche, ...

## Angriffskonzeption

- > Welche Angriffe sind erfolgsversprechend?
  - Rollen / Zuständigkeiten
  - "Drehbuch" / Personenbeschreibungen erstellen
- > Reihenfolge und Zeitplan festlegen
- > Im Zusammenspiel mit dem Auftraggeber:
  - Gewählte Szenarien genehmigen lassen
  - Abbruchkriterien definieren
  - Vertragl. und gesetzl. Erlaubnis prüfen
  - Ggf. Dritte einbeziehen
- > Requisiten beschaffen / Material vorbereiten: Uniformen, Ausweise, Dokumente
- > Üben, üben, üben...

## Durchführung

- > Per E-Mail: Abschicken und abwarten
- > Per Telefon: Notizen machen, lokale Störungen vermeiden
- > Vor Ort:
  - Üblicherweise Teamarbeit
  - Wartezeiten sinnvoll nutzen
- > Wichtig: Nichts tun, was man nicht darf!

## Berichtswesen

- > weniger spannend, aber für den Auftraggeber das Wichtigste
- > Schriftlich und / oder als Präs. / Disk.
- > Struktur ähnlich zu techn. Pentest-Reports:

- Methode und Szenario (Angriffsplan) beschreiben
  - Durchführung und Ergebnis dokum., ggf. Beweise beifügen
  - Handlungsopt aufzeigen, ggf. Empfehlungen aussprechen
- > Möglichst keine Schuldzuweisungen an Einzelpersonen
- > Auf Überbleibsel hinweisen, z.B. geöffnete, nicht mehr verschlossene Schlösser, z.B. an Schränken

## Digitale Sorglosigkeit

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.

# KAPiTEL 5

## Rechtliche Regelungen

### 1. Strafgesetzbuch (StGB)

#### Überblick

- > 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: Ausspähnen von Daten
- > § 202b: Abfangen von Daten
- > § 202c: vorb. des Ausspähens und Abfangens von Daten
- > § 202d: Datenhehlerei
- > § 205b: Strafantrag
- > § 303a: Datenveränderung
- > § 303b: Computersabotage
- > § 303c: Strafantrag

§ 202a StGB

Ausspähnen von Daten

Irz

(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.

§ 202c StGB

Diskussion

- 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](https://pentest24.de/wp-content/uploads/2022/02/HAWELLEK_LEITFADEN_.pdf)

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.

§ 202c StGB

Vorbereitung des Abfangen oder Ausspähens von Daten („Hackerparagraph“)

Irz

(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

§ 202d StGB

Datenhehlerei

Irz

- 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.

§ 303a StGB

Datenveränderung

§ 205 StGB

Strafantrag

Irz

• 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.

§ 303b StGB

Computersabotage

- (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.

(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) 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.

## 2. Datenschutz (EU-DSGVO, BayDSG)

### Informationelle Selbstbestimmung

- > (Implizites) Grundrecht, selbst über Preisgabe und Verwendung seiner personenbezogenen Daten zu bestimmen.
- > Personenbezirkbarkeit liegt vor, wenn aus den Daten auf eine Einzelperson rückgeschlossen werden kann.
  - Name, Matr.nr., E-Mail-Adresse, ...
  - IP-Adresse?

### Datenschutz - Gesetzgebung

- > Eur. Datenschutzgrundverordnung (EU-DSGV)
- > Bundesdatenschutzgesetz (BDSG)
- > Bay. Datenschutzgesetz (BayDSG)
- > Regelungen auch in anderen Gesetzen, im Umfeld von IT-Diensten besonders relevant z.B.
  - Telekommunikationsgesetz (TKG)
  - Telemediengesetz (TMG)

### Grundprinzipien:

- Verbot mit Erlaubnisvorbehalt
- Datenvermeidung und Datensparsamkeit
- Zweckbindung
- Transparenz

### Datenschutz-Gesetzgebung

### 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.

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

- EU-DSGV seit 25.05.18 in Kraft

### EU-DSGV

- 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)

### EU-DSGV Rechte

- 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 anderes)

### EU-DSGV Pflichten für Verantwortliche

- Datenschutzfreundliche Vereinstellungen (data protection by default) Art. 25
- Führen eines **Verzeichnisses der Verarbeitungstätigkeiten** (Art 30):
  - Kontaktdata 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

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

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

32

## EU-DGSVO

### Auskunftsrechte und -pflichten

- 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



## EU-DSGV

### Datenschutz-Folgenabschätzung

- 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, Gewerkschaftsgehörigkeit hervorgeht sowie genetische, biometrische Daten zur eindeutigen Identifizierung, Gesundheitsdaten oder Daten zum Sexualleben oder zur sexuellen Orientierung hervorgehen
  - systematische und umfangreiche Überwachung öffentlicher Bereiche



### Typische Aufgabenbereiche eines Universitäts-DSB

- Videoüberwachung von Bereichen / Räumen
- Anwesenheitslisten und Notenaushänge
- Personal-, Studierenden-, Alumniverwaltungswerzeuge
- 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
- ...



### 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 Administratoraktivitäten**, Auswertung nur anlassbezogen mit Vier-Augen-Prinzip
- Jährliche Schulung, schriftliche Verpflichtung von Administratoren auf das **Datengeheimnis** (§ 5 BDSG)



## 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
- Karentzeit 2 Jahre, Evaluation des Gesetzes nach 4 Jahren

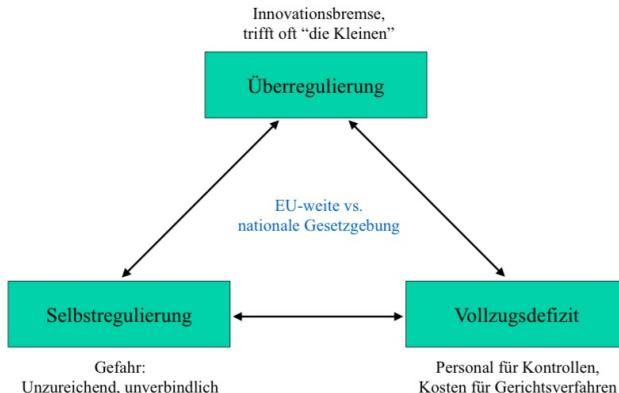


### IT-Sicherheitsgesetz 2.0 - IT-SIG 2.0 (2021)

- 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 Vertrauenswürdigkeitserklaerung abgeben
    - BMI kann Einsatz untersagen
    - Einführung eines neuen IT-Sicherheitskennzeichens; Verantwortlich BSI

- EU-Richtlinie zur Netz- und Informationssicherheit (NIS-2) am 27.12.22 veröffentlicht
- gilt nicht unmittelbar wie EU-Verordnung - Mitgliedsstaaten müssen Richtlinie bis **Oktobe 2024** in nationales Recht umsetzen
- NIS definiert Maßnahmen zur Gewährleistung eines hohen gemeinsamen Sicherheitsniveaus von Netz- und Informationssystemen
- Aufbau nationaler Kapazitäten für Cybersicherheit
- Mindestanforderungen und Meldepflichten für KRITIS
- Gesetzentwurf Bund NIS-2-Umsetzungs- und Cybersicherheitsstärkungsgesetz v. 2.10.24
  - <https://dserver.bundestag.de/btd/20/131/2013184.pdf>
  - Anlage 1 definiert „besonders wichtige und wichtige Einrichtungen“
  - Öffnungsklausel für Länder §28 Abs. 9
- Bayerisches Digitalgesetz (BayDIG) tritt am 28.10. in Kraft
- § 49 Abs. 3 LSI legt „Einrichtungen mit Bedeutung für den Binnenmarkt“ fest

## Spannungsfeld Gesetzgebung



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

## 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, Karentzeiten
  - Einflussnahme durch Lobbyisten
  - Umsetzungs- und Kontrolldefizite
  - Rechtssicherheit vs. unerwartete Gerichtsurteile

# KAPiTEL 6

## Kryptographische Grundlagen

1. Kryptologie: Begriffe, Klassifikation
2. Steganographie

### Kryptologie - Begriffe, Klassifikation

- > 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, Komm.-partner zu authentisieren, ..
- > Steganographie (verdecktes Schreiben): Methoden, die bereits die Existenz der geheimen Nachricht verborgen (geheime Nachricht in anderer, nicht geheimer „Nachrichten“ verborgen)
- Unterscheidung: linguist. u. techn. Steganographie

### Linguistische Steganographie

- > Semagramme: Nachrichten, die in Details von Schriften oder Bildern verborgen sind.
- > Maskierung (Open Code): Nachricht verborgen in offen übertragener, unverfälschter Nachricht
- > Jargon, Millieu-Code: Sondersprachen oder Sonderzeichen bezgl. o. gesell. Art

### Technische Steganographie

- > Alle Arten von „Geheimtinten“
- > Steganographie in digitalen Bildern; Bsp.e mit outguess

### Steganographie in Bildern

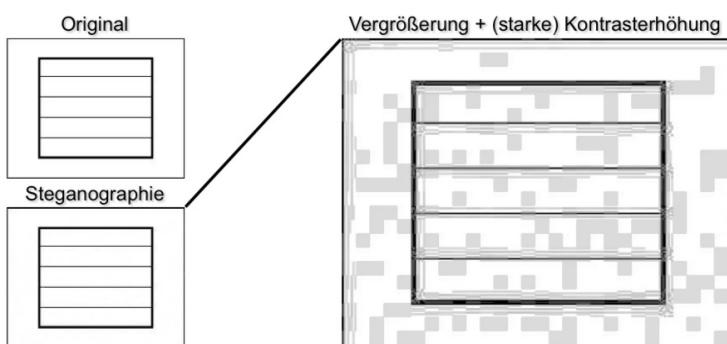
- > Cover = Bild in das die Nachricht eingebettet wird
- > Finde redundante Bits im Cover: Least Significant Bits, „Rauschen“, Nähe zusammenliegende Farben
- > Kodieren der Nachricht in diesen redundanten Bits
- > Steganographie führt zu „sehr geringen Veränderungen“ im Bild

	Pixel 1	rot	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	1	0	0	1	1	1	1	0
1	0	0	1	1	1	1	0				
	grün	0	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table>	0	0	1	0	0	1	1	1
0	0	1	0	0	1	1	1				
	blau	1	<table border="1"> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	1	0	1	1	0	0	0
1	1	0	1	1	0	0	0				
	Pixel 2	rot	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table>	1	0	0	1	1	1	0	1
1	0	0	1	1	1	0	1				
	grün	0	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0				
	blau	1	<table border="1"> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table>	1	1	0	1	1	0	1	1
1	1	0	1	1	0	1	1				



### Merkmale

Unterschiede bei "sehr strukturierten Bildern" mit hohem versteckt. 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 unverfälschte 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 verdeckte Daten erkennen können.
  - Ununterscheidbarkeit (indistinguishability): Monitor/Zensor soll bei einem ihm bekannten verdeckten Kanal nicht erkennen können, ob aktuell verdeckte Daten übertragen werden oder nicht.
  - Bandbreite (bandwidth): Länge der pro Zeiteinheit verdeckt übertragbaren Daten.

## Ron Rivest Spreu-und-Weizen-Algorithmus

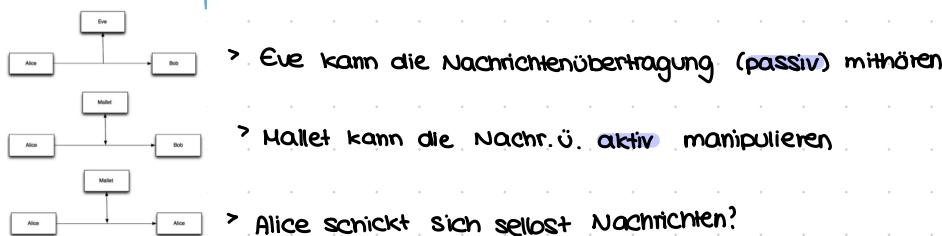
- Geheime Nachrichten sind „Nadeln im Heuhaufen“
- Alice schickt kontinuierlich Datenpakete an Bob
- Bob wertet aber nur einen Bruchteil aller Datenpakete aus
  - Alice und Bob müssen vorab / out-of-band ein Auswahlverfahren festlegen, um Spreu und Weizen trennen zu können.
- Beispiel:
  - Prüfsummen-Verfahren, 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.

## 3. Kryptographie, Begriffe und Def.

### Kryptographie - Begriffe

- > **KlarTEXT (Plaintext)**: zu verschlüsselnde Nachricht
- > **GeheimTEXT (Ciphertext)**: verschlüsselte Nachricht
- > **Verschlüsselung, Chiffrierung (Encryption)**: Vorgang, der Klartext in Geheimtext (Chiffertext) überführt
- > **Entschlüsselung, Dechiffrierung (Decryption)**: Überführung von Geheimtext in Klartext
- > **Chiffrierverfahren (Cryptographic Algorithm, Cipher)**: Algorithm. Verfahren zur Ver- bzw. Entschlüsselung
- > Algorithmen werden parametrisiert über **Schlüssel (key)**

### Angriffszenarien



### Def. - Kryptographische 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 Verschlüsselungsfkt.
- >  $d: C \times K \rightarrow M$  ist Entschlüsselungsfkt.

$$\forall k_e \in K : f(k_e) = k_d$$

$$d(e(m, k_e), k_d) = m$$

	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

### Kryptosystem: Bsp. - Substitution

- > **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:  $\epsilon: A_1^n \rightarrow A_2^m$
- > Schlüssel  $K_E = K_D$

### Kryptosystem: Bsp. - Permutation

- > 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)
- > 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

Zyklischschreibweise:

(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:

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

### Kryptosystem - Symm. Verfahren

- > Komm. partner 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  $c$ :  $m = d(c, k_{A,B}) = d(e(m, k_{A,B}), k_{A,B})$

## Kryptosys - Asymmetrische Verfahren

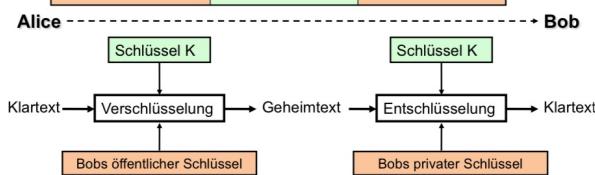
> Jeder Partner besitzt Schlüsselpaar aus private key und public key

> 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 öffentl. Schlüssel:  $c = e(m, k_e^B)$
4. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel:  $m = d(c, k_d^B) = d(e(m, k_e^B), k_d^B)$

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



## 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)

## Kryptoanalyse



- 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

### Abschätzung

#### 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

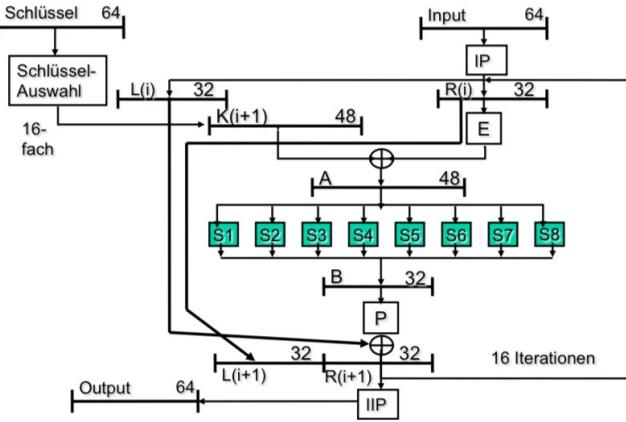
# KAPiTET 7

## Symmetrische Kryptosysteme

### Symmetrische Verschlüsselungsverfahren

#### Data Encryption Standard (DES)

##### DES



### 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  $2^{56}$
- ✗ 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

- > 29 beidseitig bestückte Platinen mit je 64 Deep Crack Chips
- > knackte DES-Schlüssel innerhalb weniger Tage

### DES Varianten - Double und Triple DES

- > **Double-DES**:  $\text{DES}(\text{DES}(m,K_1),K_2)$
- > Erwartete Komplexität bei Schlüssellänge  $n: 2^{2n}$
- > Known-Plaintext Angriff möglich ist mit Kompl.  $2^{n+1}$
- > D.h. doppelte Ausführung von DES bringt keine relevante Steigerung der Sicherheit

### Block- und Stromchiffren

#### Blockchiffren (Bsp.: 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 (Bsp.: 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

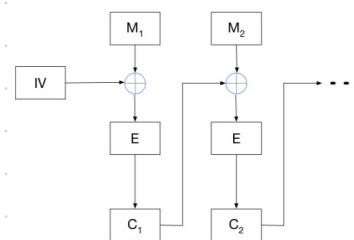
### Betriebsmodi von Blockchiffren

- **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-Modus; aber: Kein wahlfreier Zugriff.

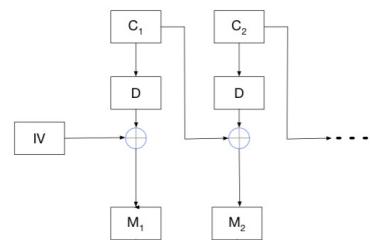
#### Cipher Block Chaining (CBC-Modus)

##### Verschlüsselung



■ Fortpflanzung von Übertragungsfehlern?

##### Entschlüsselung



## Advanced Encryption Standard (AES)

### AES

#### Kandidaten

- 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, L. Knudsen
HPC	USA	R. Schroeppel	TWOFISH	USA	B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson
LOKI97	Australien	L. Brown, J. Pieprzyk u.a.			

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

### AES

#### Round 2 Finalisten und Ergebnis

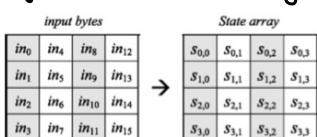
- Finalisten der Runde 2:

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](http://www.nist.gov/aes) mit Link auf AES-Homepage

### AES

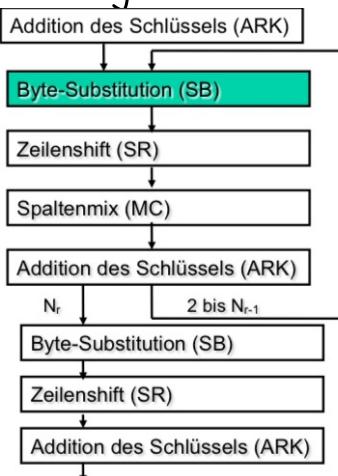
- Variable Blocklänge:  $32 \cdot N_b$  Bits
- Variable Schlüssellänge:  $32 \cdot N_k$  Bits
- $N_b$  und  $N_k$  aus  $[4, 8]$ ; im Standard eingeschränkt auf 4, 6 oder 8
- Abgeleitete Runden-Anzahl  $N_r = \max(N_b, N_k) + 6$
- Folgende Bsp.e für  $N_b = N_k = 4$  (Block- und Schlüssellänge 128 Bits; 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 Nr-1:

1. Byte-Substitution (SubBytes, SB)
2. Zeilenshift (ShiftRows, SR)
3. Spaltenmix (MixColumns, MC)
4. Addition des Rundenschlüssels (AddRoundkey, ARK)

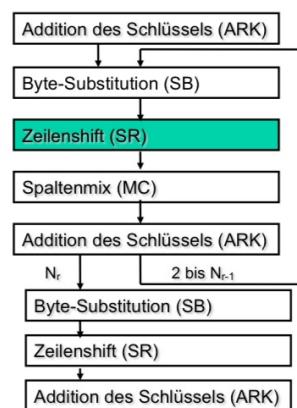
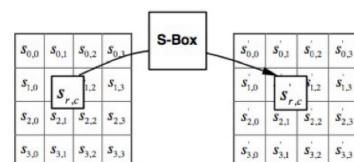
#### Entschlüsselung: Runde 1 bis Nr-1:

1. Inverser Zeilenshift
2. Inverse Byte-Subst.
3. Add. des Rundenschlüssels
4. Inverser Spaltenmix

> Letzte Runden Nr

analog, aber ohne  
(inversen) Spaltenmix

### AES: Ver- und Entschlüsselung



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

### AES Bytesubstitution

#### Implementierung

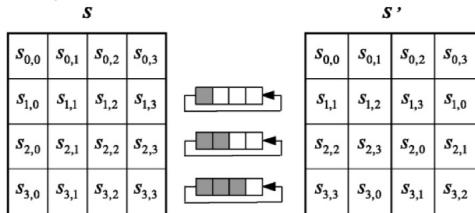
#### Rijndael S-Box (aus FIPS 197)

- Eingabe 53 wird zu Ausgabe ed

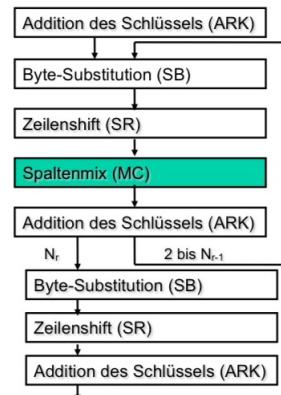
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	2f	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	99	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	f0	93	26	3f	f7	cc	34	a5	e5	f1	71	d8	31	15	
3	04	0e	03	4b	5b	92	3a	9f	9a	07	0e	02	eb	27	b2	75
4	9b	83	2c	1b	6e	09	00	5b	2b	d6	01	64	57	6f	46	4d
5	53	d1	00	60	20	f0	b1	5b	6a	cb	39	4a	4c	58	c6	
6	d0	ef	aa	fb	43	4d	33	85	5b	45	f9	02	7f	50	3c	9f
7	51	a3	40	8f	92	9d	38	7f	bc	b6	da	21	10	ff	f3	d2
8	c0	13	ec	5f	97	4d	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	08	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	44	a9	6c	56	f4	ea	65	7a	ae	08
c	be	78	2e	1c	0b	c6	9b	7d	12	4b	bd	82	1	8a	9e	
d	70	3e	56	68	03	f5	0e	61	39	57	b9	86	c1	1d	9e	
e	e1	f8	98	11	99	d9	80	94	9b	1e	87	a9	ce	55	28	ff
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

## 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



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

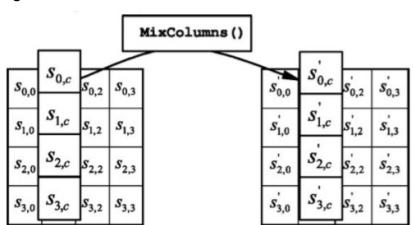


## Addition und Multiplikation in Galois-Fields (GF)

- > Add. (= Subtr.) modulo 2 = stellenweise XOR-Verknüpfung ⊕
- > Multiplikation • in GF(2<sup>8</sup>) entspricht Polynommultiplikation modulo irreduziblem (nur durch 1 oder sich selbst teilen) Polynom vom Grad 8. Für AES:  $m(x) = x^8 + x^4 + x^3 + x + 1$

## AES Spaltenmix (MixColumns())

- > Angewendet auf jede Spalte des State



- > Jede Spalte wird als Polynom vom Grad 3 mit Koeffizienten aus GF(2<sup>8</sup>) aufgefasst: Multiplikation mit dem festen Polynom  $a(x) \bmod x^4 + 1$

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

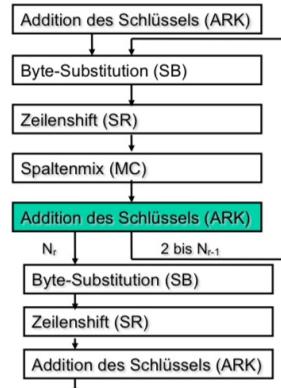
## AES: Ver- und Entschlüsselung

- > Darstellbar als Matrizenmultiplikation

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

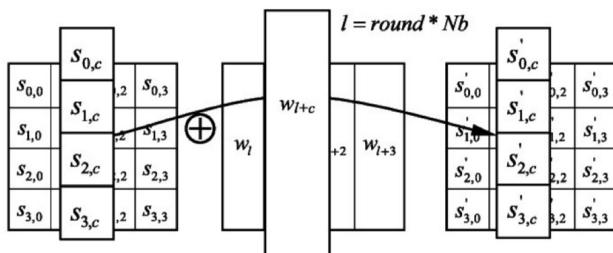
Ausmultipliziert:

$$\begin{aligned} s'_{0,c} &= (\{02\} \cdot s_{0,c}) \oplus (\{03\} \cdot s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (\{02\} \cdot s_{1,c}) \oplus (\{03\} \cdot s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \cdot s_{2,c}) \oplus (\{03\} \cdot s_{3,c}) \\ s'_{3,c} &= (\{03\} \cdot s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \cdot s_{3,c}). \end{aligned}$$



## AES: Add. des Rundenschlüssels

- > Fkt. 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
  
```

## Bestimmung des Rundenschlüssels - Erläuterung

- > 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  $N_b * (Nr+1)$  Wörter berechnet werden
- > Die ersten  $N_k$  Wörter entsprechen dem vom Anwender gewählten Schlüssel
- > Wort  $w[i]$  entspricht  $w[i-1] \oplus w[i-N_k]$

## 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])
    // 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
```

## AES - Design - Kriterien

- > Design - Kriterien müssen offen gelegt werden
- > Abschätzungen und Stellungnahme zur Widerstandsfähigkeit gegen bekannte Angriffe
- > Schlüsselauswahl mit nichtlinearer Durchmischung wg. 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...

## Kryptoregulierung

## 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)

## Kryptoregulierung

## Internationale Regelungen

- **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

## Kryptopolitik in Deutschland

- 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)

# KAPITEL 8

## Assymmetrische und hybride Kryptosysteme

### Assymmetrische Kryptosysteme

#### RSA und Sicherheit von RSA

##### ASSYMM. Kryptosysteme - 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. Fkt  $f$  mit  $f(k_d) = k_e$  soll nicht umkehrbar sein („Einwegfkt.“)

###### > Einsatz zur Verschlüsselung:

- Alice schickt Nachricht  $m$  mit Bobs Public Key verschlüsselt an Bob
- Bob entschlüsselt den empfangenen Chiffertext mit seinem priv. Schlüssel

###### > Einsatz zur elektr. Signatur:

- Alice verschlüsselt ein Dokument mit ihrem privaten Schlüssel
- Bob entschlüsselt das Dokument mit Alices öffentl. Schlüssel

### RSA

###### > Sicherheit basiert auf dem Faktorisierungsproblem:

- Geg. 2 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

### RSA - Erzeugung eines Schlüsselpaares

###### > Randomisierte Wahl von zwei ähnlich großen, unterschiedl. Primzahlen $, p$ und $q$

###### > $n = pq$ ist sog. RSA-Modul

###### > Euler'sche Phi-Fkt. 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 öff. Schlüssel besteht aus dem RSA-Modul  $n$  und dem Verschlüsselungsexponenten  $e$ .

###### > Bestimme Zahl $d$ als multiplikativ Inverse von $e$ bzgl. $\Phi(n)$

$$d = e^{-1} \bmod \Phi(n)$$

- Berechnung z.B. über den erweiterten Eukl. Algorithmus
- $n$  und  $d$  bilden den privaten Schlüssel;  $d$  muss geheim gehalten werden

### RSA - Ver- und Entschlüsselung

■ 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

## RSA - Sicherheit/mögl. Angriffe

### 1. Brute force:

- > Ausprobieren aller mögl. Schlüssel
- > Entspricht Zerlegung von  $n$  in die Faktoren  $p$  und  $q$
- > Dauert bei großen  $p$  und  $q$  mit heutiger Technik hoffnungslos lange

### 2. Chosen-Ciphertext-Angriff:

- > Angreifer möchte Ciphertext  $c$  entschlüsseln, also  $m = c^d \pmod{n}$  berechnen
- > Angreifer kann einen Ciphertext  $c'$  vorgeben und bekommt  $m'$  geliefert
- > Angreifer wählt  $c' = s^e c \pmod{n}$ , mit Zufallszahl  $s$
- > Aus der Antwort  $m' = c'^d \pmod{n}$  kann  $m = m' s^{-1}$  rekonstruiert werden.

### 1. Angriffe auf Signaturen (vgl. spätere Folien zur dig. Signatur)

- Multiplikativität von RSA  $m^{r \cdot e} = (mr)^e$  erlaubt die Konstruktion gültiger Signaturen für ein Dokument, das aus korrekt signierten Teildokumenten zusammengesetzt ist.

### 2. Timing-Angriff: [Kocher 1995]

- Überwachung der Laufzeit von Entschlüsselungsoperationen
- Über Laufzeitunterschiede kann privater Schlüssel ermittelt werden
- Gegenmaßnahme: Blinding; Alice berechnet statt  $c^d \pmod{n}$  mit einmaliger Zufallszahl  $r$

$$(r^e c)^d \pmod{n} = r c^d \pmod{n}$$

und multipliziert das Ergebnis mit der Inversen von  $r$ .

- Folge: Dauer der Entschlüsselungsoperationen hängt nicht mehr direkt nur von  $c$  ab, Timing-Angriff scheitert.

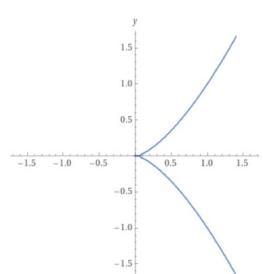
## RSA - Mathematische Angriffe

- > Mathemat. Angriffe lassen sich auf Faktorisierung zurückführen
- > Schnellster bekannter Algorithmus: General Number Field Sieve (GNFS)
  - Laufzeitkomplexität  $L(N) = e^{(c + o(1)) \sqrt[3]{\log(N)} \cdot \log(\log(N))^2}$
  - Speicherplatzkomplexität:  $\sqrt{L(N)}$
- > Angriffe werden ggf. einfacher:
  - Wenn die Anzahlen der Ziffern von  $p$  und  $q$  große Unterschiede aufweisen (z.B.  $|p| = 10$  und  $|q| = 120$ )
  - Falls  $d < 1/3 \cdot \sqrt[n]{N}$ , kann  $d$  leicht berechnet werden
  - Die ersten  $m/4$  Ziffern oder die letzten  $m/4$  Ziffern von  $p$  und  $q$  sind bekannt.

## Elliptische Kurven

- Funktionen der folgenden Form

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



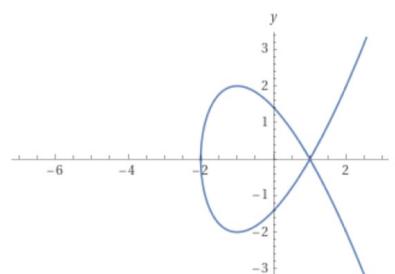
## Elliptische Kurven

- Funktionen der folgenden Form

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

- Beispiel:

$$y^2 = x^3 - 3x + 2$$



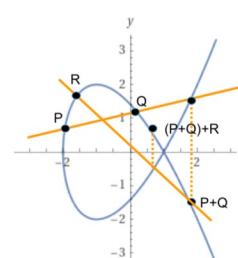
## Elliptische Kurven Rechnen

- Wähle zwei Punkte  $P$  und  $Q$  auf der Kurve

• Gerade durch  $P$  und  $Q$  schneidet Kurve in drittem Punkt

• Spiegelung des Punktes an der X-Achse liefert  $P+Q$

• Wähle Punkt  $R$  Gerade durch  $(P+Q)$  und  $R$  liefert  $(P+Q)+R$



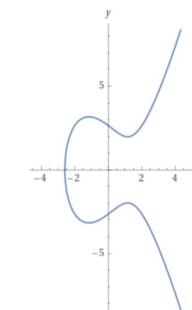
## Elliptische Kurven

- Funktionen der folgenden Form

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

- Beispiel:

$$y^2 = x^3 - 4x + 7$$

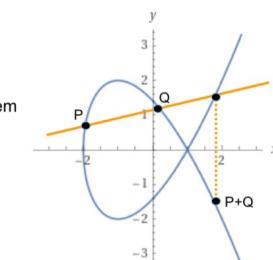


## Elliptische Kurven Rechnen

- Wähle zwei Punkte  $P$  und  $Q$  auf der Kurve

• Gerade durch  $P$  und  $Q$  schneidet Kurve in drittem Punkt

• Spiegelung des Punktes an der X-Achse liefert  $P+Q$



## Elliptische Kurven Rechnen

- Wähle Tangente an Punkt  $P$

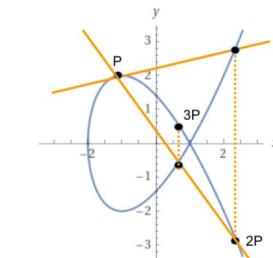
• Spiegelung am Schnittpunkt liefert  $(P+P)$

• Weitere Gerade von  $2P$  nach  $P$  liefert  $3P$

• Verfahren lässt sich beliebig fortsetzen:  $4P, 5P, 6P, 7P, \dots, NP$

• Gegeben sei  $NP$ . Wie aufwändig ist es  $N$  zu berechnen?

• Linearer Aufwand  $O(N) = N$



## Elliptische Kurven - Komplexität erhöhen

- > Ellipt. Kurven sind Gruppen, d.h. Assoziativgesetz gilt:  $4P = 3P + P = 2P + 2P$
- > NP Zerlegung in Zweierpotenzen, bzw. Darstellung von N als Binärzahl
  - $117P = 64P + 32P + 16 + 4P + 1$
  - Komplexität  $O(N) = \ln(N)$
  - Als Einwegfkt nutzbar

### Elliptische Kurven

#### Verschlüsselung

- Alice und Bob einigen sich auf Punkt P
- Alice wählt geheimes A, Bob geheimes B
- Alice berechne AP, Bob BP
- Öffentliche Schlüssel AP und BP können verteilt werden
- Alice berechnet mit geheimem Schlüssel A:  $BP^A$
- Bob berechnet:  $AP^B$
- Damit haben beide  $(BP)^A = BA^P = (AP)^B$
- Dieses gemeinsame Geheimnis kann als symmetrischer Schlüssel verwendet werden (Elliptic Curve Diffie-Hellman)
- In der Kryptographie keine reellen Werte sondern ganzzahlige verwendet, d.h.  $y^2 = x^3 + a * x + b \text{ mod } p$

### Elliptische Kurven

#### ECDSA Signatur (r,s) Berechnung

1.  $k = \text{Zufallszahl}(\{1, 2, \dots, n-2\})$
2.  $Q = kG$
3.  $r = \text{OS2I}(\text{FE2OS}(x_Q)) \text{ mod } n$   
IF  $r == 0$  goto 1.  
(mit  $x_Q$  X-Koordinate von Q)  
OS2I = Octet string to Integer Conversion  
FE2OS = Finite Field Element to Octet String
4.  $k_{inv} = k^{-1} \text{ mod } n$
5.  $s = k_{inv} * (r * d_A + \text{OS2I}(H_\tau(M))) \text{ mod } n$   
IF  $s == 0$  goto 1.  
 $H_\tau$  Hashfunktion liefert die  $\tau$  ersten Bits ( $\tau = \lceil \ln(n) \rceil$ ),  $d_A$  private key von A
6. Output (r,s)

## Schlüssellängen und Schlüsselsicherheit

Wie lang muss ein sicherer Schlüssel sein?

### Einflussfaktoren

- Symmetrisches oder asymmetrisches Verfahren ?
- Algorithmus
- PC-/softwarebasierter Angriff, oder
- Angriff mit dedizierter Hardware
  - Angriff mit integrierter Schaltung (ASIC, application specific integrated circuit)
  - Angriff mit programmierbarer integrierten Schaltung (FPGA, field programmable gate array)
  - GPGPU (General-purpose computing on graphics processing units)
- Kosten und Ressourcenbedarf

### RSA Schlüssellängen

- RSA Challenge: Belohnung für das Brechen von RSA Schlüsseln, z.B. durch Faktorisierung

Dezimalstellen	Bits	Datum	Aufwand	Algorithmus
100	332	April 1991	7 Mips Jahre	Quadratisches Sieb
110	365	April 1992	75 Mips J.	
120	398	Juni 1993	830 Mips J.	
129	428	April 1994	5000 Mips J.	
130	431	April 1996	1000 Mips J.	General Number Field Sieve (GNFS)
140	465	Februar 1999	2000 Mips J.	
155	512	August 1999	8000 Mips J.	
160	530	April 2003	k.A.	GNFS(Lattice Sieve)
174	576	Dez. 2003	k.A.	GNFS(Lattice/Line Sieve)
193	640	Nov. 2005	30 2,2-GHz-Opteron-Jahre	GNFS

### Elliptische Kurven

#### Elliptic Curve Based Signature Algorithms

- ECDSA, ECGDSA und EC-Schnorr (s. [BSI-TR-03111](#))
- ECDSA Signatur (r,s)
- Input:
  - Starke Kryptographische Hash-Funktion
  - Privater Schlüssel von A
  - Parameter der Elliptischen Kurve (p,a,b,G,n,h)
    - p Primzahl
    - a,b Parameter der Kurvenfunktion
    - G Basispunkt als Generator des Körpers  $E(\mathbb{F}_p)$  (analog zu P aus vorigen Beispielen)
    - n Ordnung von G; Typischerweise Primzahl mit  $|n| \geq 224$  bit (Länge ab 224 bit)
    - h Cofaktor von G in  $E(\mathbb{F}_p)$ ;  $h = \frac{\#E(\mathbb{F}_p)}{n}$



## Angriffe auf symmetrische Kryptosysteme

- Brute-Force Angriff
  - Durchsuchen des gesamten Schlüsselraums
  - Im Mittel ist halber Schlüsselraum zu durchsuchen
- Referenzzahlen; Größenordnungen (gerundet)

	Größenordnung
Sekunden in einem Jahr	$3 * 10^7$
Alter des Universums in Sekunden	$4 * 10^{17}$
Schlüsselraum bei 64 Bit Schlüssellänge	$2 * 10^{19}$
Masse des Mondes [kg]	$7 * 10^{22}$
Masse der Erde [kg]	$6 * 10^{24}$
Masse der Sonne [kg]	$2 * 10^{30}$
Schlüsselraum bei 128 Bit Schlüssellänge	$3 * 10^{38}$
Anzahl Elektronen im Universum	$10^{77} - 10^{79}$

### RSA Schlüssellängen (Forts.)

- RSA Challenge wurde 2007 eingestellt
  - rund \$30.000 Preisgeld ausbezahlt
- RSA-768 wurde 2009 von Kleinjung et al. „geknackt“
  - hätte \$50.000 Preisgeld eingebracht

Dezimalstellen	Bits	Datum	Aufwand	Algorithmus
232	768	Dez. 2009	1/2 Jahr 80 CPUs (Vorauswahl)	GNFS

- Bereits 2007 wurde von Kleinjung et al. die 1039. Mersenne-Zahl (1039-Bit-Zahl) faktoriert
  - war allerdings nicht Bestandteil der RSA Challenge



- Verschiedene Institutionen geben Vergleiche heraus  
Bits of Security (äquiv. Schlüssellänge symmetrischer Verfahren)
- [NIST](#) (National Institute of Standards and Technology) 2007:

Bits of Security	80	112	128	192	256
Modullänge (pq)	1024	2048	3072	7680	15360

- [NESSIE](#) (New European Schemes for Signatures, Integrity and Encryption) (2003)

Bits of Security	56	64	80	112	128	160
Modullänge (pq)	512	768	1536	4096	6000	10000

## Hybride Kryptosysteme

### Hybride Kryptosysteme

- Vereinen Vorteile von symmetrischen und asymmetrischen Verfahren
- [Asymmetrisches Verfahren zum Schlüsselaustausch](#)
- [Symmetrisches Verfahren zur Kommunikationsverschlüsselung](#)

Alice

Bob

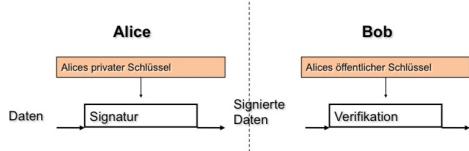


- Beispiele für hybride Verfahren: SSL/TLS, PGP, SSH...
- Oftmals neuer Schlüssel pro Nachricht oder Zeiteinheit, aus K abgeleitet

## Elektronische Signatur

### Elektronische Signatur

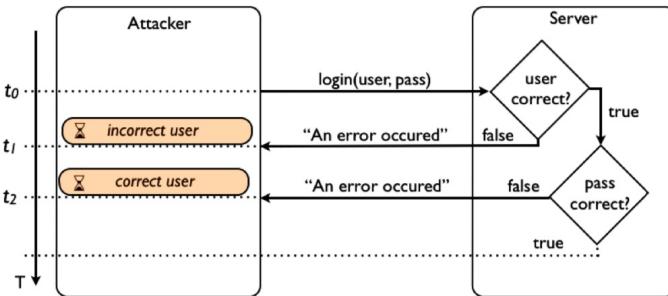
- Alice „signiert“ Daten mit ihrem privaten Schlüssel
- Jeder kann die Signatur mit Alices öffentlichem Schlüssel verifizieren



- Asymmetrische Verfahren sind im Vergleich sehr langsam
- Daher i.d.R. nicht Signatur der gesamten Daten
- [Lediglich kryptographischer Hash-Wert der Daten wird signiert](#) (digitaler Fingerabdruck der Daten)

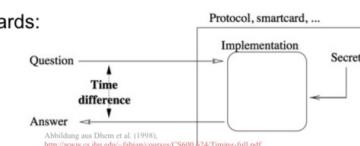
### Timing-Angriffe

- Zunächst am Beispiel Webanwendungen:

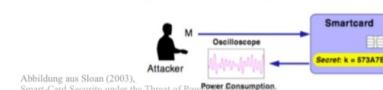


### Timing-Angriffe

- Auf RSA-Smartcards:



- Different power consumption when operating on logical ones vs. logical zeroes.



### Elektronische Signatur

#### Analogie zur Unterschrift

- Zentrale Anforderungen an die (analoge) Unterschrift:
  - Perpetuierungsfunktion: Dokument und Unterschrift sind dauerhaft.
  - Echtheitsfunktion: Die Unterschrift ist authentisch.
  - Die Unterschrift kann nicht wieder verwendet werden.
  - Abschlussfunktion: Unterschrift ist räumlicher Abschluss des Dokuments; dieses kann später nicht verändert werden.
  - Beweisfunktion: Unterzeichner kann seine Unterschrift später nicht leugnen.
- Weitere Anforderungen?
- Bei der Unterschrift auf Papier ist keine dieser Anforderungen vollständig erfüllt! Trotzdem wird die Unterschrift im Rechtsverkehr akzeptiert. Ihre Funktion wird durch Rahmenbedingungen gesichert.

### Elektronische Signatur

#### Erfüllung oder Anforderungen?

- Perpetuierungsfunktion: Fälschungssicher und dauerhaft
- Echtheitsfunktion: Authentizität sichergestellt
- Wiederverwendbarkeit: Wie gewünscht nicht gegeben**
- Abschlussfunktion: Nicht veränderbar
- Beweisfunktion: Unterschrift ist nicht zu leugnen
- Solange privater Schlüssel geheim gehalten wird.
- Abhängig von zweifelsfreier Zuordnung des Schlüsselpaares zu einer Identität (Zertifizierung, CA)
- Digitale Signatur „beinhaltet“ den Dateninhalt
- vgl. 3.
- Jeder kann Signatur bzw. Echtheit mit öffentlichem Schlüssel des Unterzeichners verifizieren.

# KAPITEL 9

## Kryptographische Hash-Fkt.en

### Hash-Fkt.en zur Integritätssicherung

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

### Herkömmliche vs. kryptographische Hash-Fkt.en

- > Prüfsummen dienen der Erkennung von (unbeabsichtigten) Übertragungsfehlern, z.B. beim IPv4-Header
- > Kryptographische Prüfsummen sollen auch absichtliche Manipulationen erschweren

### Grundlagen

#### > Hash-Fkt.en

- bilden "Universum" auf endlichen Bildbereich ab
- sind nicht inj.
- Bildbereich i.d.R. sehr viel kleiner als Universum
- Kollisionen möglich:  $\exists x, y \in U: x \neq y \wedge h(x) = h(y)$

#### > Kryptographische Hash-Fkt. H:

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

### Def. Kryptographische Hashfkt.

#### > Schwache Hash-Fkt. H:

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

#### > Starke Hash-Fkt. H:

- H hat alle Eigenschaften einer schwachen Hash-Fkt.
- Es ist zusätzlich praktisch unmöglich, eine Kollision zu finden, d.h. ein Paar verschiedene Eingabewerte m und m' mit:  $m' \neq m, m, m' \in A^* \wedge H(m) = H(m')$

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

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

Antwort: 23       $P = 1 - \frac{365 \cdot 364 \cdots (365 - (n-1))}{365^n}$       (ab  $n=23$  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!

### Birthday Attack

#### Vorgehensweise

1. Alice sichert mit einem k Bits langen Hash eine Nachricht M
  2. Mallot 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
  - ....

### Konstruktion kryptographischer Hash-Fkt.en

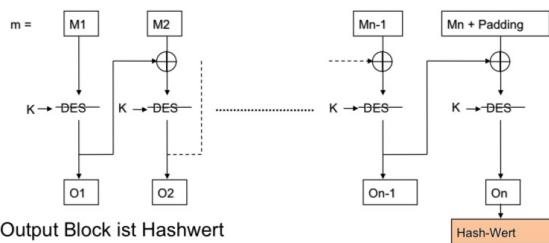
- > Folge von Kompressionsfkt.en G
- > Nachricht m wird in Blöcke  $M_1, M_2, \dots, M_n + \text{Padding}$  mit fester Länge y zerlegt
- > Hash-Verfahren wird mit Initialisierungswert IV vorbelegt.



- > Letzter Block  $M_n$  muss ggf. auf vorgegebene Länge y „aufgefüllt“ werden (Padding)
- > Als Kompressionsfkt. G können verwendet werden:
  - Hash-Fkt.en auf der Basis Symm. Blockchiffren
  - Dediizierte Hash-Fkt.en

## DES als Kompressionsfkt.

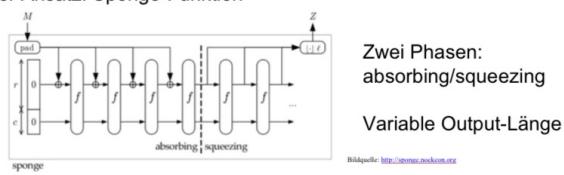
### > DES im Cipher Block Chaining (CBC) Mode



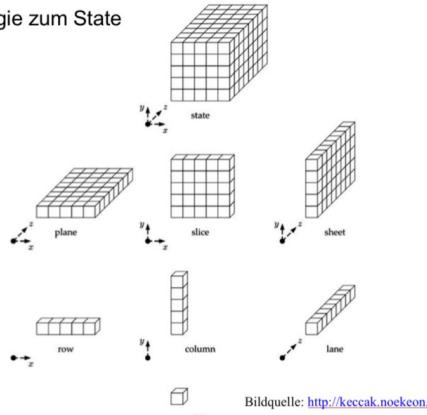
- Letzter Output Block ist Hashwert
- Länge des Hashwerts? 64 Bits

## SHA-3

- 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



## Keccak: Terminologie zum State



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

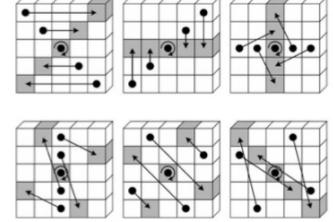
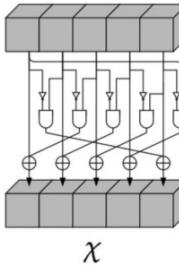
## Keccak: Parametrisierung und Keccak-f

- 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 Permutation  $b = c + r = 25 \cdot 2^l$
- 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:  $b = 1152 + 448 = 1600$ , d.h.  $l = 6$ ,  $w = 64$
- Jede Anwendung von  $f$  besteht aus  $nr$  Runden:  $nr = 12 + 2^l$ , d.h. für SHA3-256:  $nr = 24$

## Keccak-f: Runden

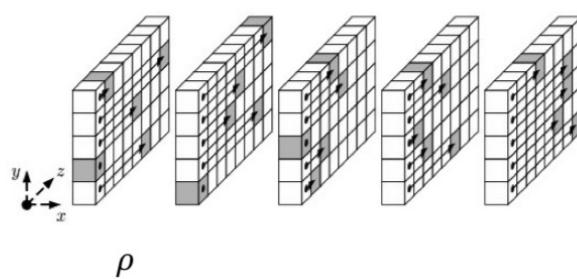
- Jede Runde besteht aus fünf Schritten:

- $R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$ .
- 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

# KAPiTEL 10

## 10. Sicherheitsmechanismen

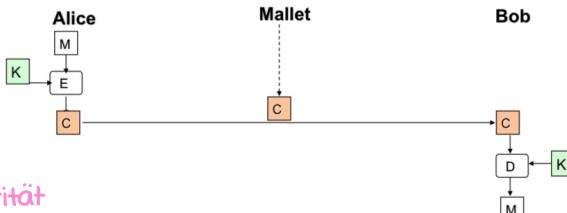
### 1. Vertraulichkeit

### 2. Integritätssicherung

#### Vertraulichkeit (Confidentiality)

- > Schutz der Daten von unberechtigter Offenlegung
- > Wie kann Vertraulichkeit realisiert werden?

- Durch Verschlüsselung (Encryption)
- Mallet kann Chiffrentext mangels Kenntnis des Schlüssels nicht nutzen



#### Integrität

- > 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?

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

#### Angriff auf Mechanismen zur Integritätsicherung

- > Angreifer verändert unbemerkt Daten und Hash-Wert
- > Deshalb: Hash-Wert und ggf. Sequenznummern müssen vor Veränderungen geschützt werden
  - Sequenznummern o. 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 Integritäts sicherung
    - 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  $H(m)$ ?

#### 3. Authentisierung

##### Arten

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

## Benutzerauthentisierung

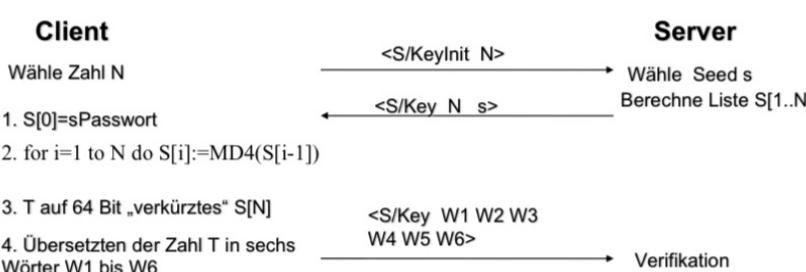
- > Wissen: PW, Passphrase (Unix PW Verfahren, vgl. Kap. 3); Einmal-PW; PIN; ...
- > Besitz: Smartcard, Token, („physischer“) Schlüssel, Token-App auf Smartphone; Kryptograph. Schlüssel als Datei
- > Eigenschaft: Biometrie:
  - Fingerabdruck, Stimmerkennung, Gesichtserkennung, Iris-Scan, Hand-Geometrie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

## Einmal PW

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

## Einmal-PW Verfahren: S/Key (1995)

- > Authentisierungsserver kennt PW des Benutzers



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

## S/Key - Details

- > Verkürzungsfkt.:  $T := S[N]$  (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
- > Übersetzungsfkt. für T
  - Ergebnis 6 kurze (1 bis 4 Zeichen lange) engl. Wörter
  - Wörterbuch mit 2048 Wörtern (in RFC 1760 enthalten)
  - Je 11 Bit von T liefern - als Zahl interpretiert - die Nr. des Wortes
- > Bsp. für einen solchen „Satz“: HIT HARD LIKE A DOOM GOAT

## S/Key - Bewertung

- > Gute Hashfkt.en bieten ausreichend Schutz vor dem Ableiten des n. PWs aus den vorher. n-1 PW.
- > Ohne weitere Schutzmaßnahmen anfällig für Man-in-the-Middle-Angriffe
- > Benutzer muss Reihenfolge der PW 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“

## S1Key und OTP - Angriffe

### > Dictionary Attack:

- Alle Nachrichten werden im Klartext übertragen
- Angreifer kann mit diesen Infos versuchen, das PW d. 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 PWs 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 Authent. d. Servers

## Time-Based One Time PW (TOTP)

> Weiterentwicklung von HMAC based OTP (HOTP) [RFC 4226]:  $\text{HOTP}(K,C) = \text{HMAC-SHA1}(K,C)$  mit Schlüssel/PW K und Counter C

> TOTP spezifiziert in [RFC 6238]:

- $\text{TOTP}(K) = \text{HOTP}(K, C_t)$  mit
- $C_t = \lfloor \frac{T-T_0}{T_x} \rfloor$  wobei
  - To 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 fkt. nicht mehr
- ABER: Gefahr des Diebstahls von K

### Authentisierung Smartcards

#### □ 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



### RSA SecurID Details

- 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)



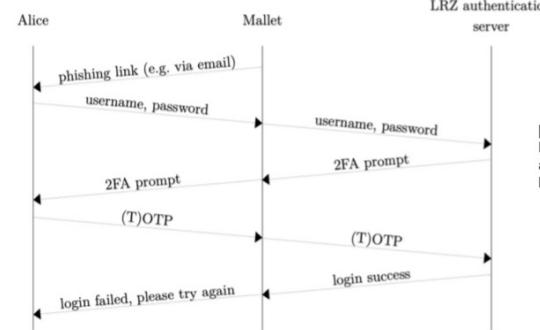
### Authentisierung

#### RSA SecurID

- 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.



## 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.

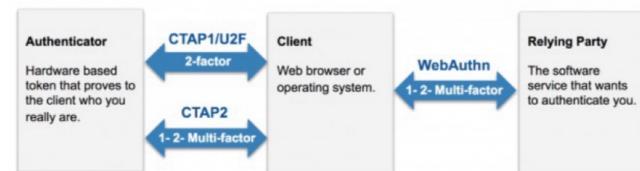
## FIDO2 (Fast Identity Online)

- > 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 ü. WebAuthn Protocol



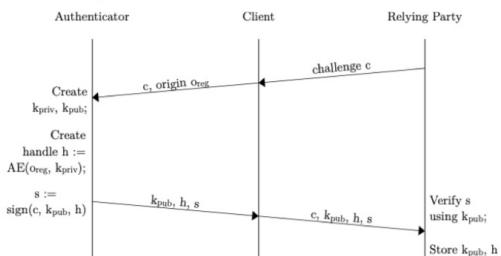
## > Authentisierungsprotokolle für Web-Anwendungen der FIDO Alliance

- CTAP (Client to Authenticator Protocol)
- U2F (FIDO Universal 2nd Factor Protocol)
- WebAuthn (stand. v. W3C)



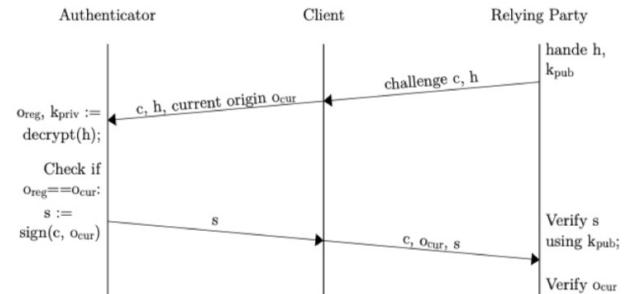
## FIDO2 Registrierung

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



## FIDO2 Phishing Protection

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



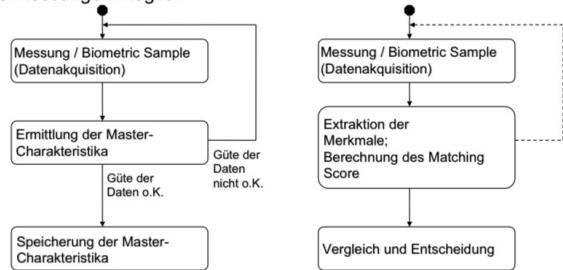
## FIDO2 Authentisierungsarten

- > Passwordless Authentication
- > 2FA, z.B. durch Yubikey, Biometrie oder PIN/PW
- > MFA
- > WebAuthn für Authentisierung im Web spezifiziert
- > Anpassung f. andere Services notwendig, z.B. OpenSSH

## 3.1 Peer Entity / Benutzer

### Biometrie: allg. Vorgehen

- Initialisierung des Systems pro Nutzer
  - Viele Messungen möglich



### Biometrie - Anwendungen

- > Anmeldung an PCs / Notebooks
- > Zutrittskontrolle:
  - zu Räumen in Bürogebäuden, Rechenzentren, ...
  - Zoo Hannover hat Gesichtserkennungssystem
  - Fingerabdruckleser in Fitness-Studios etc.
- > Biometr. Reisepass
- > Kriminalistik, z.B. Fingerabdruck

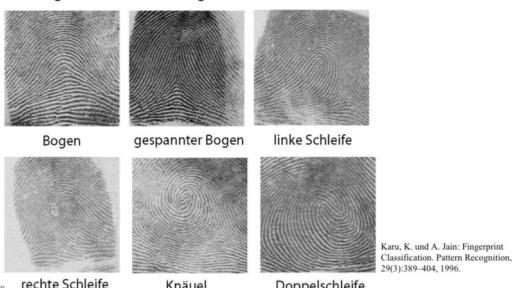
### Fingerabdruck: Merkmalsextraktion

- 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



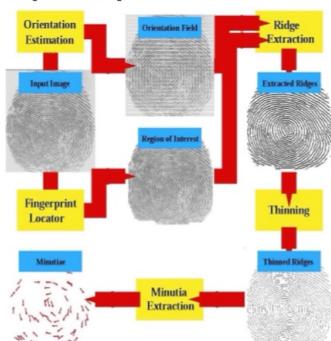
- Solche äquivalente Rillenstrukturen werden zu einer Minuzie zusammengefasst
- Merkmale: Lage der Minuzien
  - Absolut bezüglich des Abdrucks und relativ zueinander
  - Orientierung bzw. Richtung

IT-Sicherheit | WS 24/25 | © Helm



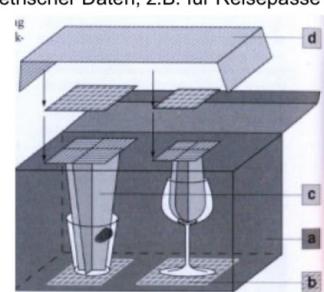
### Fingerabdruck: Minutiae Extraktion

- Algorithmus: Beispiel aus [JHPB 97]



### 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 Authentisierung

### Fehlerarten

- Abschätzung der Fehlerraten:  
N: Anzahl der Identitäten  
FP: Falsch Positiv (Falschakzept.)  
FN: Falsch Negativ (Falschrückw.)

Es gilt (PPK03):  
 $FN(N) \approx FN$   
 $FP(N) \approx 1 - (1 - FP)^N \approx N \times FP$

falls  
 $N \times FP < 0,1$

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

## Benutzerauthentisierung:

### Multimodale Systeme

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

	Wissen	Besitz	Biometrie
Wissen			
Besitz			
Biometrie			

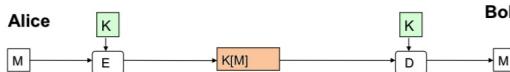
- Auch verschiedene biometrische Verfahren lassen sich kombinieren:
  - Erhöhung der Sicherheit
  - Verringerung der Fehlerraten
  - Z.B. Iris-Scan mit Spracherkennung kombiniert

## 3.2 Datenursprung

### Authentisierung des Datenursprungs

- > Mögl. zur Authent. des Datenursprungs bzw. zur Peer-Entity-Authentication:
  1. Verschlüsselung der Nachricht (Authentisierung erfolgt mittelbar durch Wissen, d.h. Kenntnis d. Schlüssels)
  2. Digitale Signatur
  3. Message Authentication Code (MAC):  $MAC = \text{Hashverfahren} + \text{gemeins. Schlüssel}$
  4. Hashed Message Authentication Code (HMAC)
- > Kombi. d. angegebenen Verfahren

### Authentisierung durch symm. Verschlüsselung

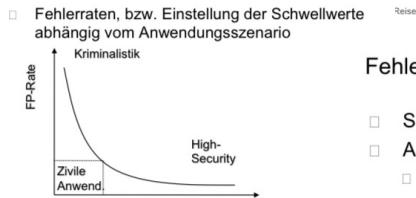
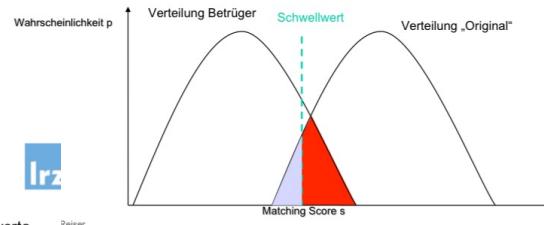


- Merkmale:
  - Authentisierung des Datenursprungs (Nachricht kann nur von Alice stammen, wenn der Schlüssel nur Alice und Bob bekannt ist)
  - Bob wird nicht explizit authentisiert, aber nur Bob kann Nachricht nutzen
  - Vertraulichkeit der Daten (nur Alice und Bob kennen K)
- „Nachteile“:
  - Sender kann die Sendung leugnen (Bob könnte sich die Nachricht auch selbst geschickt haben)
  - Alice / Bob können Zugang / Empfang nicht beweisen

## Biometrische Authentisierung

### Fehlerarten

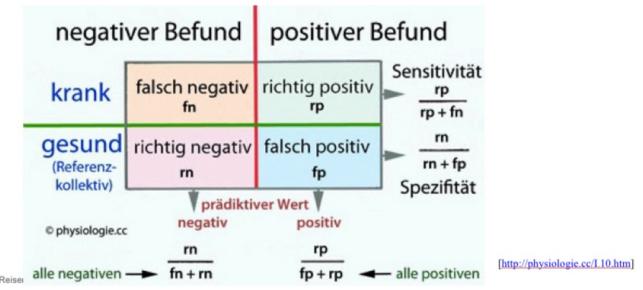
- Biometrische Systeme sind fehlerbehaftet
- Fehlerarten:
  - Falsch Positiv / Falschakzeptanzrate (Mallet wird als Alice authentisiert)
  - Falsch Negativ / Falschrückweisungsrate (Alice wird nicht als Alice identifiziert)
- Fehler sind abhängig von Schwellwerteinstellungen



- 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



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

### Authentisierung durch asym. Verschlüsselung

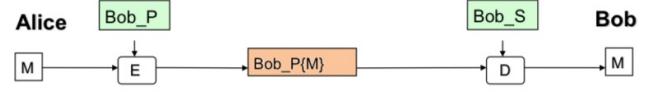


Alice

Bob\_P

Bob\_S

Bob



### 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

## Digitale Signatur



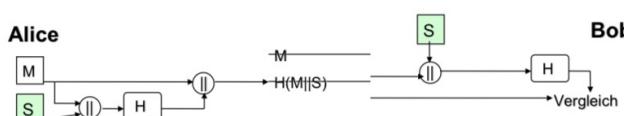
- Merkmale:
  - Authentisierung des Datenursprungs (Nachricht kann nur von Alice stammen; nur Alice kennt ihren geheimen Schlüssel)
  - Jeder kann die Signatur verifizieren (auch ohne Mithilfe von Alice)
  - Alice kann die Sendung nicht leugnen
  - Bob wird nicht authentisiert
  - Keine Vertraulichkeit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

Authentisierung  
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 Integritäts sicherung („blinde“ Modifikation des Chiffertextes wird nicht erkannt)

## Verwendung von Hash-Fkt. zur Authentisierung



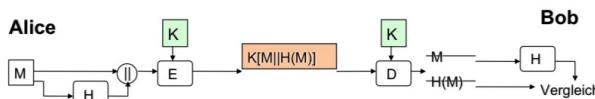
- Authentisierung des Datenursprungs (durch „Geheimnis“ S)
  - Nachricht wird mit S konkateniert und dann der Hash berechnet
- (Daten-) Integrität (durch Hash)
- Keine Vertraulichkeit, jeder kann M lesen
- Alice kann Sendung leugnen
- Alice/Bob können Zugang / Empfang nicht beweisen

## Verwendung von Hash-Fkt. zur Authentisierung

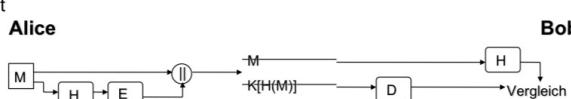


- Zusätzlich Vertraulichkeit durch Verschlüsselung
- Alice kann Sendung leugnen
- Alice/Bob können Zugang / Empfang nicht beweisen

## Verwendung von Hash-Fkt. zur Authentisierung

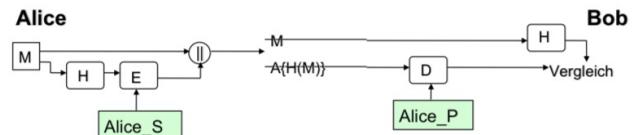


- Authentisierung des Datenursprungs (durch Schlüssel K)
- Vertraulichkeit
- Integrität



- Authentisierung und Integrität, keine Vertraulichkeit

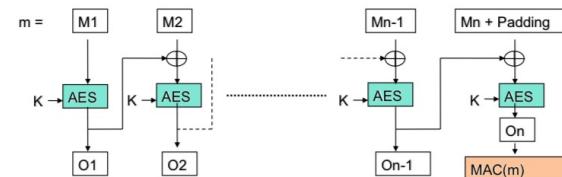
## Verwendung von Hash-Fkt. zur Authentisierung



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

## Authentisierung: MAC

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



▫ AES im CBC Mode

## Sicherheit von MACs

- > 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 MAC (m, K)
  - Für alle  $K_i$  berechnet der Angreifer:  $MAC(m, K_i) == MAC(m, K)$ ?
  - D.h. der Angreifer muss  $2^k$  MACs erzeugen
  - Es existieren aber nur  $2^n$  versch. MACs ( $2^{n-k}$  Schlüssele)
  - D.h. mehrere  $K_i$  generieren den passenden MAC ( $2^{(k-n)}$  Schlüssele)
  - Angreifer muss den Angriff iterieren:
    1. Runde liefert für  $2^k$  Schlüssele 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

### MAC-length extension attack

- Möglich, wenn Hash-Fkt. mit Merkle-Damgard-Konstruktion verwendet wird (z.B. MD5, SHA, SHA-1)
- $\text{MAC}(k, m)$ , z.B.  $\text{SHA-1}(k || m)$
- Dienst liefert für  $m$  MAC als Ausweis für Dienstnutzung
- Angreifer kennt Blocklänge und Länge der Nachricht
- Angreifer kann Nachricht verlängern ohne  $k$  zu kennen
- $\text{SHA-1}(k || mm')$  liefert "gültigen" Hash auch ohne Kenntnis von  $k$
- Bsp.  $m = \text{Überweise - 100-€}$
- Bsp.  $m' = \text{\textbackslash}x00\text{\textbackslash}x00\text{\textbackslash}x00 \& \text{Überweise - 10000000000 - \$}$

### Hashed MAC (HMAC)

- Ges.: MAC, der nicht symm. Verschlüsselung, sondern kryptographische Hash-Fkt. zur Kompression verwendet: Hashes wie SHA-3 sind deutlich schneller als z.B. DES
- Problem: Hash-Fkt.en verwenden keinen Schlüssel
- Lösung HMAC:
  - Belieb. Hash-Fkt.  $H$  verwendbar, die auf (Input) Blöcken arbeitet
  - Sei  $b$  die Blocklänge (meist 512 Bits)
  - Belieb. 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 konst. Input- $(ipad)$  bzw. Output-Pattern  $(opad)$  XOR verknüpft:  $ipad = 0x36$  ( $b$  mal wdh.),  $opad = 0x5c$  ( $b$  mal wdh.)

### HMAC Algorithmus



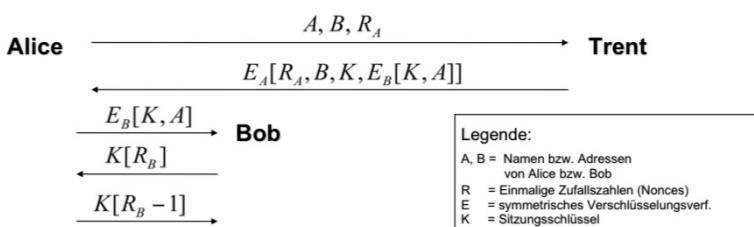
$$HMAC(m) = H[(K^+ \oplus opad) || H[(K^+ \oplus ipad) || m]]$$

- $K^+$  := Schlüssel  $K$  auf Länge von  $b$  Bits gebracht
  - $b$  Bits langer Block  $S_i := K^+ XOR ipad$
  - Nachricht  $m$  mit dem Block  $S_i$  konkatenieren
  - Hash-Wert von  $S_i || m$  berechnen
  - $b$ -Bit-Block  $S_o := K^+ XOR opad$
  - $S_o$  mit dem Ergebnis von 4. konkatenieren
  - 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.

### 3.3 Authentisierungsprotokolle

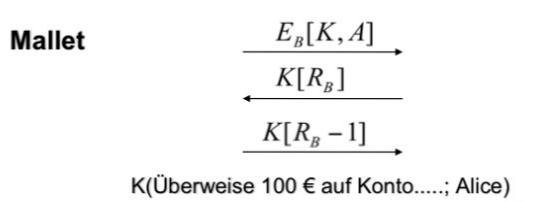
#### Needham-Schröder

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



## Needham - Schröder - Protokollschwäche

- > Problem: Alte Sitzungsschlüssele 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 ( $K_{A,T}, K_{B,T}$ )



## Lösungs Idee:

- Seq.nr. oder Timestamps einführen
- Gültigkeitsdauer von Sitzungsschlüsseln festlegen

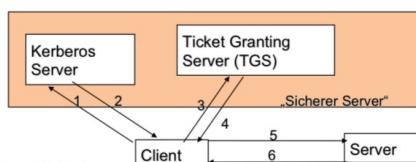
## Kerberos

- > 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 o. Software) kann sich ü. ein Netz bei Server(n) authentisieren
- > Kerberos-Server kennt Schlüssel aller Clients
- > Basiert auf symm. Verschlüsselung
- > Abgeleitet vom Needham-Schröder-Protokoll
- > Hierarchie von Authentisierungsservern möglich; jeder Server verwaltet einen bestimmten Bereich (sog. Realm)
- > Über koop. Mechanismen der Kerberos-Server kann Single-Sign-On realisiert werden

## Kerberos - Authentisierungsdaten

- > Authentisierung basiert auf gemeinsamen (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) $T = s, c, \text{addr}, \text{timestamp}, \text{lifetime}, K_{c,s}$
- > Authenticator:
  - „Ausweis“ zur Authentisierung; damit Server ein Ticket verifizieren kann
  - vom Client selbst erzeugt
  - wird zusammen mit dem Ticket verschickt $A_{c,s} = c, \text{addr}, \text{timestamp}$

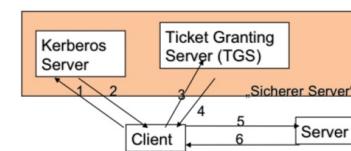
## Kerberos Modell



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]

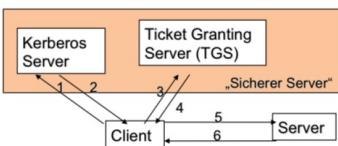


## Kerberos: Initiales Ticket (ein Mal pro Sitzung)



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

$c$ =	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 $y$



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

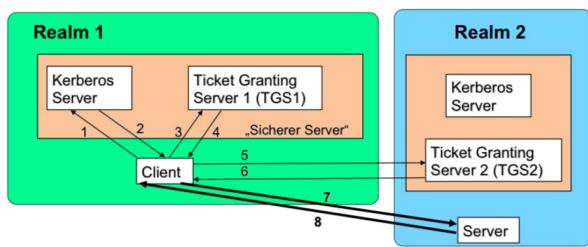
3. Request für Server Ticket:  
 $s, K_{c,tgs}[A_{c,tgs}], K_{tgs}[T_{c,tgs}]$  mit  $A_{c,tgs} = c, a, t$     $T_{c,tgs} = tgs, c, a, t, v, K_{c,tgs}$

4. Server Ticket:  
 $K_{c,tgs}[K_{c,s}], K_s[T_{c,s}]$  mit  $T_{c,s} = s, c, a, t, v, K_{c,s}$

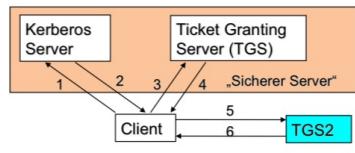
## Multi-Domain-Kerberos

- 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: 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

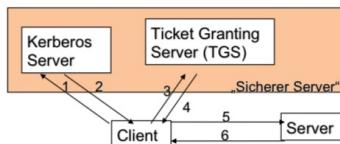
3. Request for Server Ticket for foreign TGS (TGS2):  
 $tgs2, K_{c,tgs1}[A_{c,tgs1}], K_{tgs1}[T_{c,tgs1}]$   
mit  $A_{c,tgs1} = c, a, t$ ;  $T_{c,tgs1} = tgs1, c, a, t, v, K_{c,tgs1}$

4. Server Ticket:  
 $K_{c,tgs1}[K_{c,tgs2}], K_{tgs2}[T_{c,tgs2}]$  mit  $T_{c,tgs2} = tgs2, c, a, t, v, K_{c,tgs2}$

## Kerberos

### Bewertung

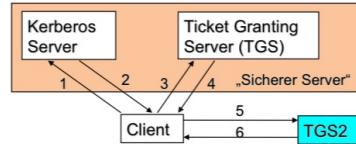
- Sichere netzweite Authentisierung auf Ebene der Dienste
- Authentisierung basiert auf IP-Adresse
  - IP-Spoofing u.U. möglich
  - Challenge Response Protokoll zur Verhinderung nur optional
- Sicherheit hängt von der Stärke der Passwörte ab (aus dem Passwort wird der Kerberos-Schlüssel abgeleitet)
- 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



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:  $s, K_{c,tgs2}[A_{c,tgs2}], K_{tgs2}[T_{c,tgs2}]$  mit  $A_{c,tgs2} = c, a, t$     $T_{c,tgs2} = tgs2, c, a, t, v, K_{c,tgs2}$
6. Server Ticket:  $K_{c,tgs2}[K_{c,s}], K_s[T_{c,s}]$

- > Domänenübergreif. Authentisierung
- > Erfordert Schlüsselaustausch zw. TGS1 und TGS2: KTGS1, KTGS2
- > Vertrauen (Trust) erforderlich:
  - Besuchende Domäne muss Authenticator und TGS der Heimatdomäne vertrauen
  - Beide Domänen müssen sich auf „sichere“ Implementierung verlassen
- > Skalierungsproblem: n Realms erfordern  $n^*(n-1)/2$  Schlüssel, d.h.  $O(n^2)$



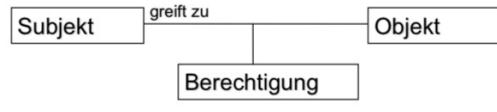
c	=	Client
s	=	Server
a	=	Adresse
v	=	Gültigkeitsdauer
t	=	Zeitstempel
$K_x$	=	Schlüssel von x
$K_{x,y}$	=	Sitzungsschlüssel von x u. y
$T_{x,y}$	=	Ticket für x um y zu nutzen
$A_{x,y}$	=	Authenticator von x für y

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

## 4. Autorisierung und Zugriffskontrolle

### Autorisierung und Zugriffskontrolle

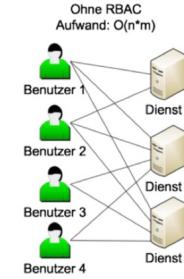
- > Autorisierung: Vergabe / Spezifikation von Berechtigungen
- > Zugriffskontrolle: Durchsetzung dieser Berechtigungen
- > Häufig werden Autorisierung und Zugriffskontrolle zusammengefasst
- > Handelnde werden als Sbj. bezeichnet
- > Berechtigungen werden an Sbj. erteilt
- > Berechtigungen gelten für Obj.
- > Obj. e sind die schützenswerten Einheiten im System



## Klassifikation

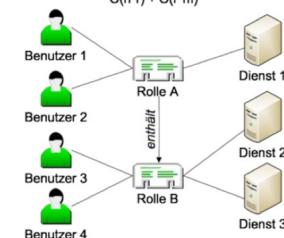
- DAC (Discretionary Access Control)
  - Basieren auf dem Eigentümerprinzip
  - Eigentümer spezifiziert Berechtigungen an seinen Objekten
  - Zugriffsrechte auf Basis der Objekte vergeben
- MAC (Mandatory Access Control)
  - Regelbasierte Festlegung der Rechte
  - Systemglobal
  - Z.B. Bell-LaPadula; 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)

Kontinuierlich zu pflegende Berechtigungszuordnungen bei n Benutzern und m Diensten:



Mit RBAC (r = Anzahl Rollen)

Aufwand bei statischer Rollenhierarchie:



## 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	read	read	
Prozess 2		read, write	signal
Prozess 3	read, write, owner		kill

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

## Referenzmonitor

- > zur Realisierung der Zugriffskontrolle ist eine sichere „vertrauenswürdige“ Systemkomponente erforderlich
- > Häufig als Referenzmonitor oder Access Control Monitor bezeichnet
- > Erfüllt folgende Anforderungen:
  - Zugriff auf Obj.e nur über den Monitor möglich
  - Monitor kann Aufrufenden (Subj.) zweifelsfrei ident. (Authentisierung)
  - Monitor kann Obj.zugriff unterbrechen bzw. verhindern

## 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

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

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

## DFN-PKI Zertifikat: Be

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

## DFN-PKI Zertifikat: Beispie

<b>Öffentlicher Schlüssel</b>	
<b>Algorithmus</b>	RSA-Verschlüsselung ( 1.2.840.113549.1.1 )
<b>Parameter</b>	Ohne
<b>Öffentlicher Schlüssel</b>	512 Byte : A4 1C 04 D6 30 EE A3 95 ...
<b>Exponent</b>	65537
<b>Schlüssellänge</b>	4.096 Bit
<b>Schlüsselverwendung</b>	Verschlüsseln, Überprüfen, Einpacken, Ableiten
<b>Signatur</b>	256 Byte : 52 67 F1 81 1D 42 DD 98 ...
<b>Erweiterung</b>	Schlüsselverwendung ( 2.5.29.15 )
<b>Kritisch</b>	JA
<b>Verwendung</b>	Digitale Signatur, Verschlüsseln von Schlüsseln
<b>Erweiterung</b>	Basiseinschränkungen ( 2.5.29.19 )
<b>Kritisch</b>	JA
<b>Zertifizierungsinstanz</b>	NEIN
<b>Erweiterung</b>	Erweiterte Schlüsselverwendung ( 2.5.29.37 )
<b>Kritisch</b>	NEIN
<b>Zweck #1</b>	Serverauthentifizierung ( 1.3.6.1.5.5.7.3.1 )
<b>Zweck #2</b>	Clientauthentifizierung ( 1.3.6.1.5.5.7.3.2 )

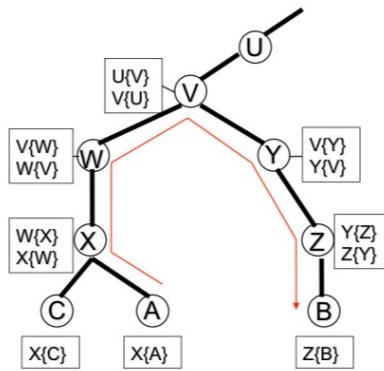
## DFN-PKI Zertifikat: Beispie

**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.aoisphere.lrz.de  
**DNS-Name** di46tel-pre.aoisphere.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** enavicon.webdb.devweb.mwn.de

DFN-PKI Zertifikat: Beispie

## Kopplung von Realms; Zertifizierungspfade

- Bisher wurde nur eine CA betrachtet, nun CA Hierarchie:
  - Legende:  $X(A)$  = Zertifikat ausgestellt von X für A ( $X$  zertifiziert  $A$ )



- 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?

## 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

lrz

#### 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
  - Zertifikatkette muss vom Client geprüft werden  
(lässt sich ggf. über Server-based Certification Validation Protocol (SCVP) an den Server auslagern)

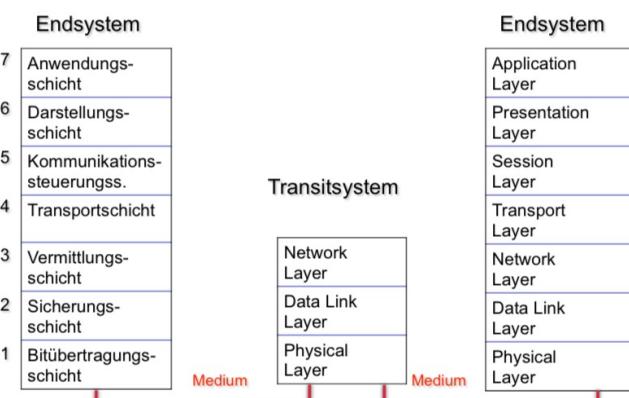
# KAPiTEL 11

## 11 Netz Sicherheit - Schicht 2: Data Link Layer

### Virtual (Private) Network

- > Grundidee: Nachbildung einer log. Netzstruktur („Local Area Network“ oder eines „nicht öffentl.“ Netzes) in beliebigen Topologien / Technologien, z.B. auch über das Internet
- > Das „virtuelle“ Netz soll u.a. bzgl. Vertraulichkeit und Datenintegrität mit physischen LANS vergleichbar sein
- > Virtualisierung auf jeder Schicht des OSI-Modells mögl.

### ISO/OSI Schichtenmodell



### 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
- > Bsp.: 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

### Virtual Network auf Schicht 2/3/4

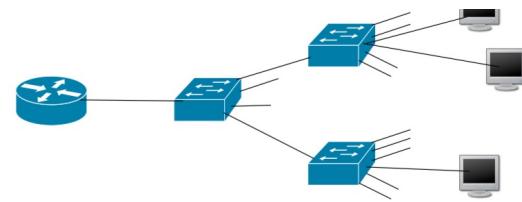
- > Schicht 2:
  - Virtual LAN (VLAN)
    - Mehrere LAN Broadcast Domains über den selben physischen Link
    - Standard: VLAN Tagging (IEEE 802.1Q)
  - Virtual Private LAN Services (Achtung: Abkürzung auch VPLS)
    - Verbindet physisch getrennte (V)LANS miteinander
  - Point-to-Point Verbindungen
  - Layer 2 Tunneling Protocol
  - ...
- > Schicht 3 und höher:
  - IPsec
  - SSL/TLS
  - OpenVPN, eduroVPN
  - ...

## Aufgaben der Schicht 2

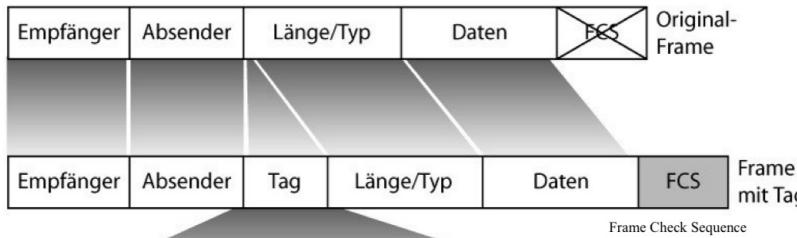
- > Fehlerfreie Übertragung von Frames (Rahmen)
  - Aufteilung von Bitströmen in Frames
  - Fehlerkontrolle über Prüfsummen (z.B. Cyclic Redundancy Check, CRC)
- > Flusskontrolle (Verhindert, dass der Empfänger mit Frames überflutet wird und diese verwerfen muss)
- > Medienzugriffsvorfahren für gemeinsam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

## Virtual LAN (VLAN)

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

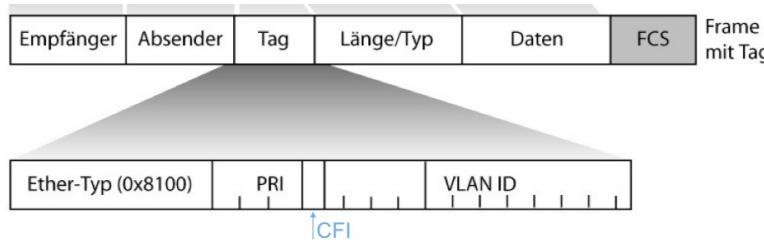


## VLAN - Datenpakete



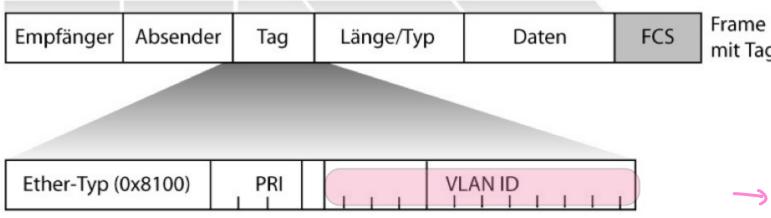
- > Virtual Local Area Network (VLAN); IEEE 802.1Q
- > VLAN definiert Broadcast-Domäne → Pakete e.g. welche MAC-Adresse zu welcher IP-Adresse
- > Idee: Erweiterung des Ethernet-Frame um sog. Tag

## VLAN - 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 kanon. Form (1 Bit); bei Ethernet 0; sonst (z.B. Token Ring) 1



### Erweiterung des Ethernet-Frame um 32-bit Tag:

- VLAN-ID: Ident. d. VLANs („VLAN NR.“) (12 Bit)
  - ID 0 = „kein VLAN“, ID 0xFFFF ist reserviert
  - Somit 4094 versch. VLANs möglich

#### → Berücksichtigen:

1. phys. Sicherheit
2. Administratoren können alles machen

## Point to Point Protokoll (PPP)

- > Punkt-zu-Punkt Protokoll: Entwickelt für Verbindungsauflauf über Wähleinrichtungen
  - DSL, ISDN, Modem, Mobilfunk, Funk, serielle Leitungen, ...
  - WAN-Verbindungen zw. 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:
    - Verbindungsauflauf und abbau
    - Test
    - Aushandeln der Konfiguration (u.a. Nutzdatenlänge pro Frame)
  - Network Control Protocol (NCP):
    - Aushandeln der Konfig. der unterstützten Schicht 3 Protokolle (z.B. IP, IPX, Appletalk, ...), versch. Schicht 3 Protokoll über einen PPP-Link möglich
- > Weitere Varianten: PPPoE (over Ethernet), PPPoA (over ATM)

## PPP - Sicherheitsdienste

- > Authentifizierung opt.
- > Im Rahmen der LCP-Aushandlung der Konfig. kann jeder Partner eine Authentifiz. fordern
- > Def. Authentifiz.protokolle:
  - 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.“

Irz

### 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:
  - 1, id,  $R_A, A$
  - 2, id,  $H(id, R_A, K_{AB}), B$
  - 3|4, id
- 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

B

### Sicherheitsrisiko PAP-Fallback

- 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

Irz

### 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 Authentication Prot.
  - ....

Irz

### EAP

#### Grundlagen

- 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



### Point to Point Tunneling Protocol (PPTP)

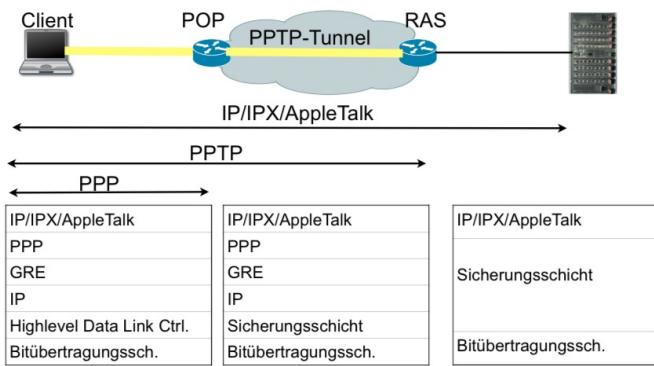
- > PPP wurde für „direkt“ verbund. Systeme entwickelt
- > Idee von PPTP (RFC 2637):
  - Ausdehnung von PPP über Internet
  - PPTP realisiert Tunnel durch/über das Internet
  - Transport von PPP PDUs in IP-Paketen
  - Dazu werden PPP PDUs mit Generic Router Encapsulation Protocol (GRE) gekapselt
  - GRE ist ein Schicht 4 Protokoll

PPP Protocol Data Unit (PPP PDU)
GRE
IP
Sicherungsschicht
Bitübertragungsschicht (Physical Layer)

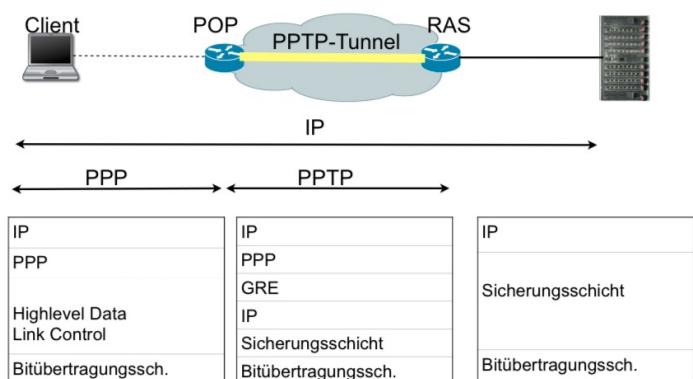
### PPTP - Anwendungsfälle

- > 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

## Voluntary Tunneling

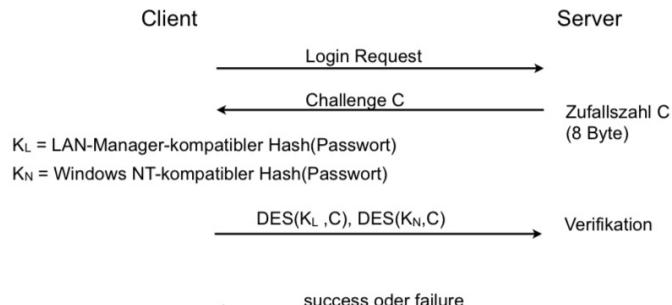


## Compulsory Tunneling



## Vergleich MSCHAP v1 und v2

## ■ Version 1:



## MSCHAP v2

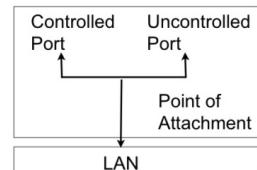
## Sicherheit

- 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

## 802.1X

## Grundlagen

- 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



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

## 802.1X

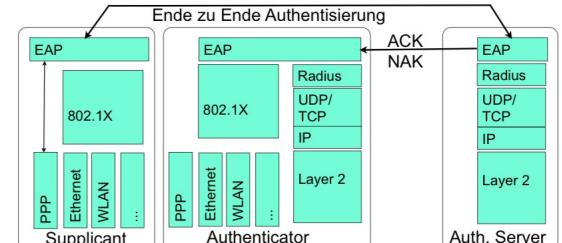
## Ablauf der Protokolle

- Möglicher Ablauf:
  1. Supplicant fordert Controlled Port
  2. Authenticator fordert Authentisierung
  3. Nach erfolgreicher Authentisierung wird der Port freigeschaltet
- Supplicant oder Authenticator können Authentisierung initiiieren
- 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

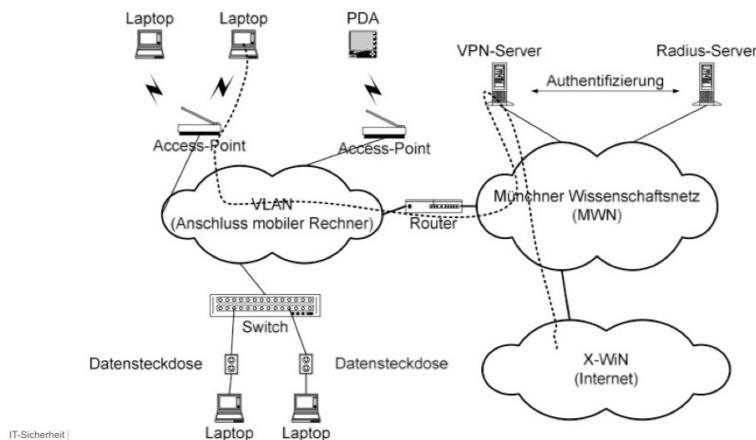
- Unterstützt verschiedene Auth.-Mechanismen
- Aushandlung erst während der Authentisierung mit Auth.-Server
- Authenticator ist nur Vermittler der Nachrichten



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

## Beispiel

### Datenzugang in öffentlichen Bereichen im MWN



### 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)

lrz

### @BayernWLAN

- 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 (Ende 2023 wieder V. gewonnen)
- 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

@BayernWLAN



### Kooperationsmodell

- 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 Winter 2024/25: >50.000 APs , davon 40 % (-20.000) von Unis und Hochschulen (gut 6.800 vom LRZ ;-)

lrz

### Beispiel aus dem MWN eduVPN



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



# EINSCHUB

einschub (Kap. 8 - QC)

### Shor Algorithmus zur Faktorisierung (1997)

- > probab. Algorithmus, Eingabe:  $n$ , Ausgabe: ein Teiler von  $n$
- > Laufzeit:  $O((\log n)^3)$
- > Benötigte Qubits zur Faktorisierung einer Zahl  $n$  mit  $N$  Binärstellen, d.h.  $n < 2^N$ 
  - Ursprüngl. Algorithmus:  $3N$  Qubits
  - bester bekannter Algorithmus:  $2N+3$  Qubits
  - gilt nur für einen fehlerfreien QC
- > Auf realen QC wird ein Faktor  $M$  mehr Qubits zur Fehlerkorrektur benötigt
- > Für  $N=2048$  ist die Annahme, dass ~20 Mio. physik. Qubits benötigt werden
- > Aktuell ist Shor deshalb noch nicht anwendbar, aber
  - Fortschritte bei der Fehlerkorrektur
  - Fortschritte bei QC
- > IBM hat 2001 auf einem 7 Qubit Rechner die Zahl 15 faktorisiert, d.h.  $N=4, M=3$

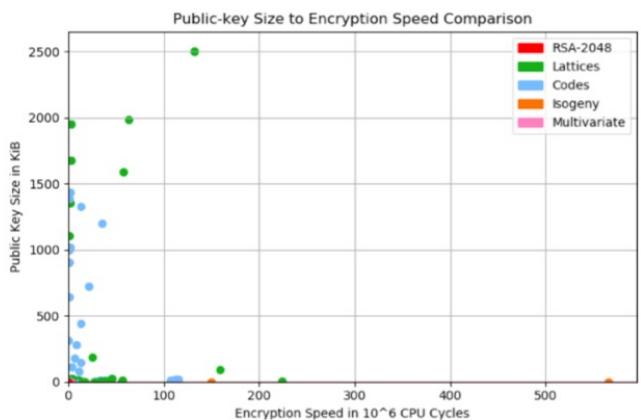
## NIST process: Post quantum cryptography (PQC)

- Research on PQC started in the 2000s
- Efforts in research, application and standardization, especially:

The US-American „National Institute of Standards and Technology“ started a process to determine the best post-quantum crypto-schemes to be used in the near future.



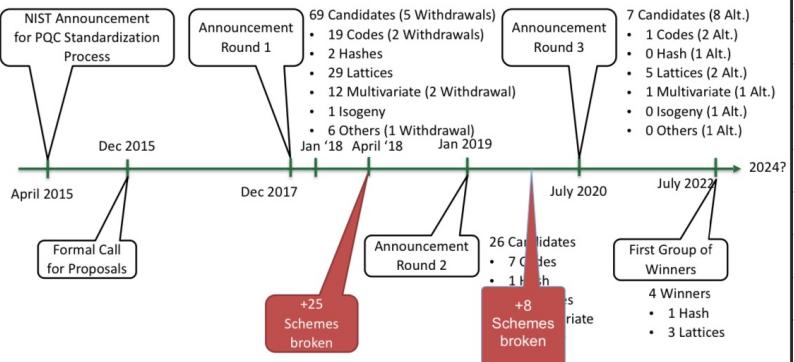
- Why a standard? → So there is some common ground for everyone.
- Goal: Standardization by 2023
- Covers Key exchange, Encryption and Signing



## NIST PQC Standardization process: Security

Level	Classical Security	Quantum Security	Asymmetric Security (DiscLog, RSA)	Examples
I	128 bits	64 bits	3,024 bits	At least as hard to break as AES128 (exhaustive key search)
II	128 bits	80 bits		At least as hard to break as SHA256 (collision search)
III	192 bits	96 bits	8,192 bits	At least as hard to break as AES192 (exhaustive key search)
IV	192 bits	128 bits		At least as hard to break as SHA384 (collision search)
V	256 bits	128 bits	15,336 bits	At least as hard to break as AES256 (exhaustive key search)

## NIST PQC Standardization process



## NIST PQC Standardization

### Patent und Intellectual Property Probleme

- After NIST announced the first group of winners
- Concerns about intellectual property rights, surrounding lattice-based schemes
  - Kyber
  - New-Hope
- NIST indicates there are two patent-portfolios relating to CRYSTALS-KYBER: <https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/selected-algos-2022/nist-pqc-license-summary-and-excerpts.pdf>
- NIST claims to take this under consideration

## Quantensichere Cryptographie

### Standardisierung in der IETF

- IETF - Internet Engineering Task Force - Standardisierung von Internet-Protokollen
  - IETF Draft - kann jeder einbringen, wird diskutiert, verbessert und dann vielleicht zu
  - RFC - Request for Comments (deFakto Standard)
  - STD - „echter“ IETF Standard
- IETF und IRTF (Internet Research Task Force) und Quanten-Krypto
  - Verschiedene Working Groups zu verschiedenen Themen, z.B. Post Quantum Cryptography (PQC) transition
  - <https://wiki.ietf.org/group/sec/PQCAgility>
  - [RFC 8554](#) - Leighton-Micali Hash-Based Signatures
  - [RFC 8391](#) - XMSS: eXtended Merkle Signature Scheme

## NIST PQC Round 3 Comparison – Security Level 1

Method	Scheme	Secret Key Sizes	Public Key Sizes	Ciphertext/Signature Sizes (32Byte Plaintext)
Encryption / Key Encapsulation	McEliece (Code)	6 KB	260 KB	128 B
	Crystals-Kyber (Lattice)	1.6 KB	800 B	768 B
Signatures	NTRU (Lattice)	1.2 KB	1 KB	1 KB
	SABER (Lattice)	1.5 KB	672 B	736 B
Signatures	Crystals-Dilithium (Lattice)	n/A	1.4 KB	2 KB
	Falcon (Lattice)	n/A	897 B	666 B
	Rainbow (Multivariate)	100 KB	157 KB	66 B

- Additional call for digital signature proposals until June 2023
- NIST Projekt: Migration to Post-Quantum Cryptography <https://csrc.nist.gov/pubs/pd/2021/08/04/migration-to-postquantum-cryptography/final>





