



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

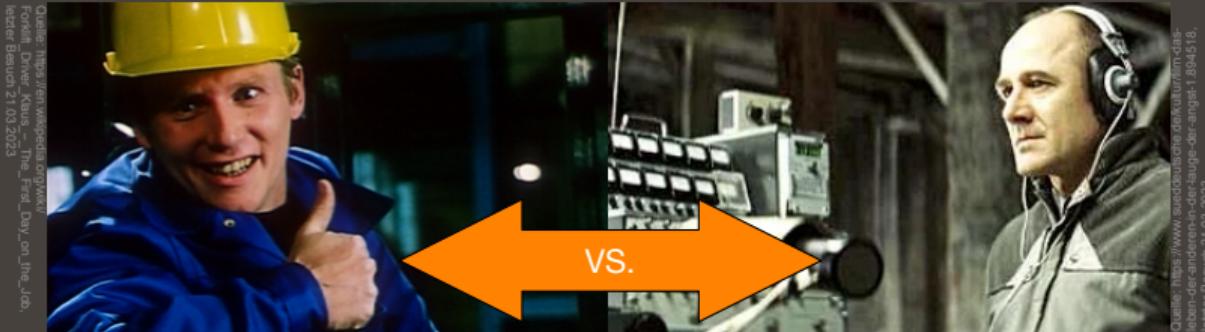
SECURITY

Einführung

April 13, 2023

Marc Stöttinger





DIE SICHERHEIT ...

Funktionssicherheit (engl. safety) zielt auf Übereinstimmung der Ist-Funktionalität der Komponenten mit der spezifizierten Soll-Funktionalität ab (Gefahrenabweitung, Ausfallsicherheit, Schutz von Leib und Leben).

Sicherheit (engl. security) umfasst alle Vorkehrungen zum Schutz von elektronisch gespeicherten Informationen sowie informationstechnischen Systemen (SW & HW).

Datenschutz (engl. privacy) regelt die Verwendung und Weitergabe personenbezogener Daten (informationelles Selbstbestimmungsrecht gemäß BDSG & DSGVO).

GLOBALE MELDUNG



37 Millionen Kunden betroffen

Hackerangriff auf T-Mobile US

20.01.2023 - 10:31 Uhr

Die Mobilfunktochter der Deutschen Telekom in den USA ist abermals Opfer eines Hackerangriffs geworden. Die mutmaßlichen Täter könnten private Informationen wie Telefonnummern und Adressen erbeutet haben.



Nach Russland-Resolution

EU-Parlament von Hackern angegriffen

23.11.2022 - 17:59 Uhr

Das EU-Parlament ist nach seiner Russland-Resolution Ziel eines Hackerangriffs geworden, bei dem Server zum Absturz gebracht wurden. Eine kremlnahe Gruppe bekannte sich offenbar zu dem Angriff.

Quelle: <https://www.tagesschau.de/thema/hackerangriff/> - letzter Besuch 21.03.23

LOKALE MELDUNG



Hunderte Unternehmen betroffen

Globale Hackerwelle trifft Deutschland

06.02.2023 - 14:27 Uhr

Eine globale Welle von Cyberangriffen hat auch deutsche Unternehmen und Institutionen lahmgelegt. Laut dem zuständigen Bundesamt könnten Hunderte Firmen betroffen sein. Eine Software-Aktualisierung könnte die Sicherheitslücke schließen.



Bericht des "Handelsblatt"

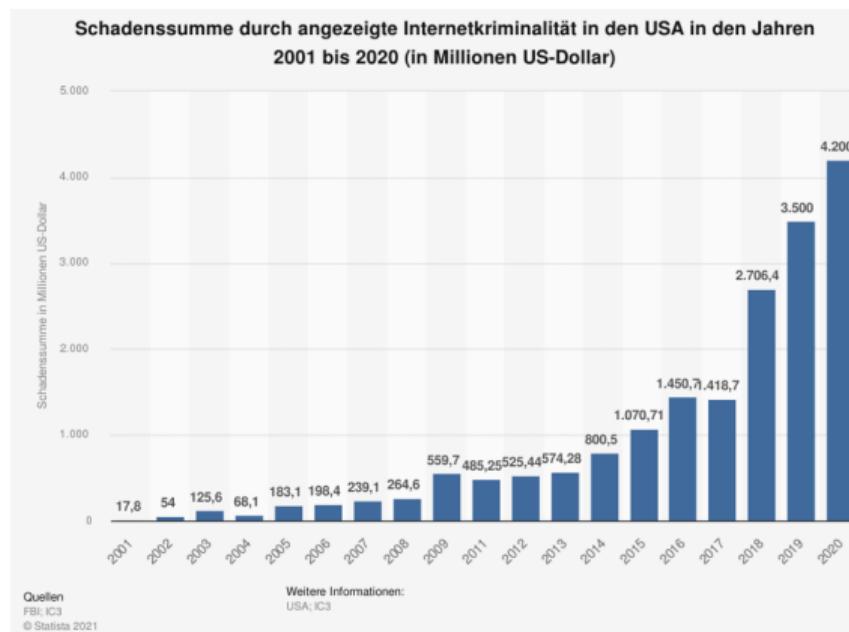
Gehackte Daten von Continental im Darknet

15.11.2022 - 12:22 Uhr

Als Continental Ende August von Hackern attackiert wurde, hieß es noch, der Cyberangriff sei erfolgreich abgewehrt worden. Doch nun werden gestohlene Daten des Automobilzulieferers im Darknet zum Kauf angeboten - für 50 Millionen Dollar.

Quelle: <https://www.tagesschau.de/thema/hackerangriff/> - letzter Besuch 21.03.23

WIRTSCHAFTLICHE AUSWIRKUNGEN



Quelle: <https://de.statista.com/statistik/daten/studie/151979/umfrage/schadenssumme-durch-internetkriminalitaet-in-den-usa-seit-dem-jahr-2001/> - letzter Besuch 21.03.23

GRÜNDE FÜR MEHR ANGRIFFE

- Zunahme der Digitalisierung - Abhängigkeit von IT-Infrastruktur
 - Im globalen (und auch lokalen) Markt ist die Vernetzung nicht mehr wegzudenken
 - Viele Geschäftsprozesse sind digitalisiert und automatisiert
 - Viele Daten und Informationen sind nur noch rein digital gespeichert
- Angriffe auf die Gesellschaft nehmen zu
 - Manipulation der Informationen in sozialen Medien
 - Wirtschaftliche Einflussnahme über die Gesellschaft

Grund und Auswirkung

- Angriffe gegen IT-Systeme werden "rentabler"
- Erhöhter Sicherheitsbedarf in Wirtschaft und Gesellschaft

DISKUSSION IN KLEINEN GRUPPEN

Tauschen Sie sich mit Ihrem Nachbarn 5 Minuten aus:

- Hatten Sie schon mal Berührungspunkte mit Sicherheitsvorfällen?
 - Wenn ja, wo und wie?
- Warum sind Ihrer Meinung nach Angriffe so oft erfolgreich?

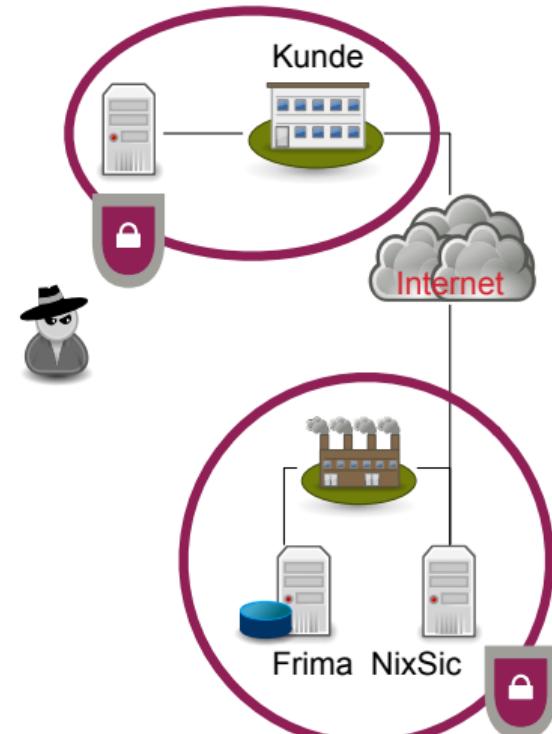
(IT-)SECURITY IST EINE SYSTEMEIGENSCHAFT

→ Sicherheitsmechanismen - Technologien

- Verschlüsselung
- Authentifizierung
- Zugriffskontrolle/Rechtemanagement
- Firewall
-

→ Qualitätssicherung - Prozesse

- Personalschulung
- Regelmäßige (Security-)Wartung
- Regelmäßige Audits und Reviews
- Hohe Testabdeckung
-



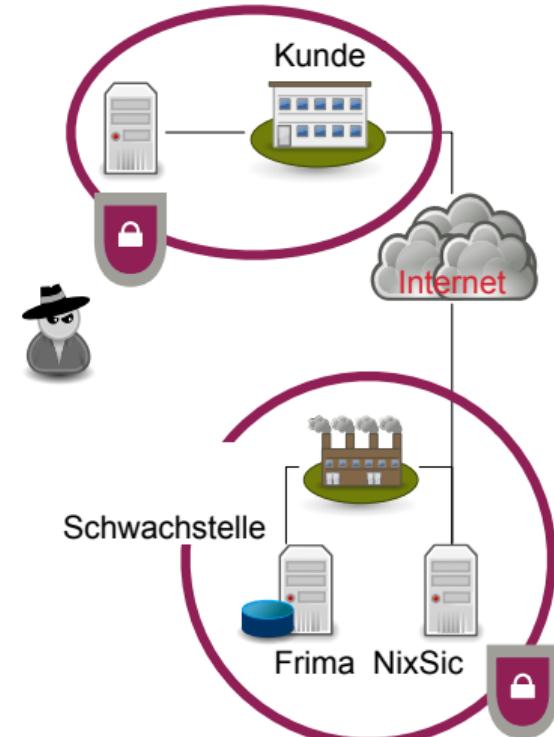
SCHWACHSTELLE - WEAKNESS

Schwachstelle

“Eine Schwachstelle ist ein sicherheitsrelevanter Fehler eines IT-Systems oder einer Institution.”
BSI

Beispiel für Schwachstellen:

- Schwache Passwörter
- Keine Softwarewartung
- Sicherheitslücke in der Software
- Offene Netzwerkports
- Ungeschultes Personal



BEDROHUNG - THREAT

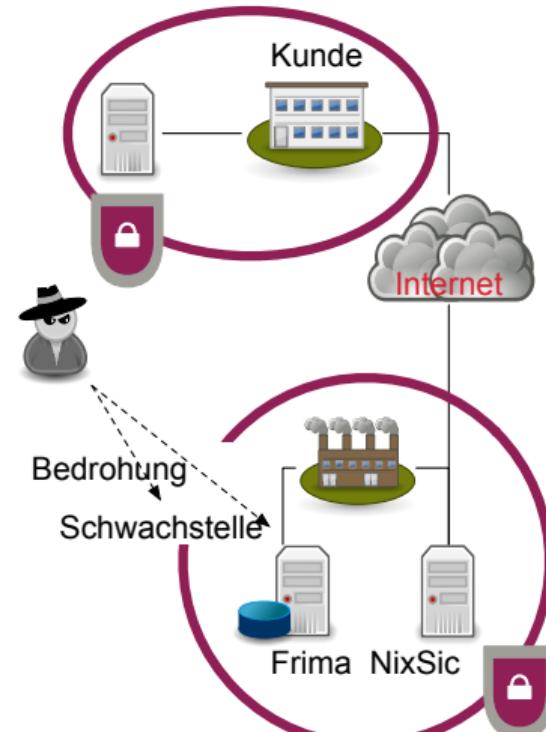
Bedrohung

“Eine Bedrohung ist ganz allgemein ein Umstand oder Ereignis, durch den oder das ein Schaden entstehen kann.”

BSI

Beispiel für Bedrohung durch:

- Passwort raten
- Ausnutzen einer (bekannten) Schwachstelle
- Durchführen von Netzwerkscans



GEFÄHRDUNG - HAZARD

Gefährdung

“Eine Gefährdung ist eine Bedrohung, die konkret über eine Schwachstelle auf ein Objekt einwirkt.”
BSI

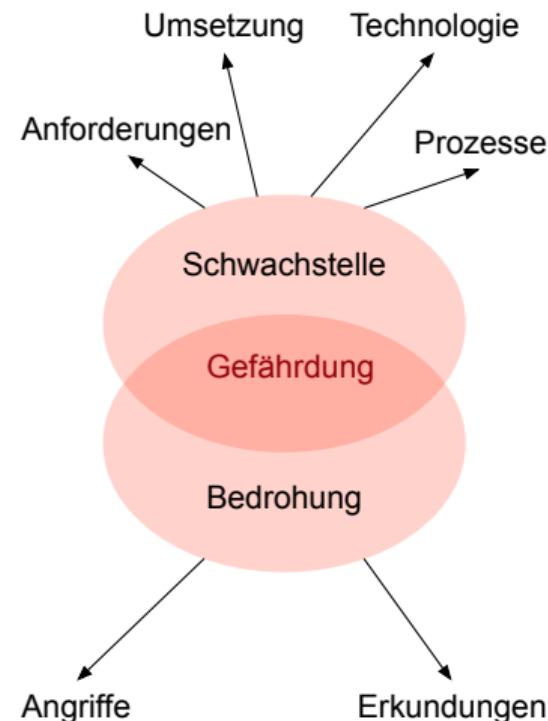
Beispiel für Bedrohung durch:

- Angreifer rät das Passwort und erhält Zugang zum System
- Angreifer nutzt bestehende Sicherheitslücke in alter Software mit einem bestehende nHack-Skript und legt die IT lahm



SICHERHEITSVORFÄLLE

- Jede Gefährdung kann zu einem Sicherheitsvorfall führen
- Gefährdungen hängen von der Anzahl der Schwachstellen und Bedrohungen ab
 - Je mehr Schwachstellen, desto mehr Gefährdung
 - Je mehr Bedrohungen, desto mehr Gefährdung
- Schwachstellen können reduziert werden
- Bedrohungen können nur schwer reduziert werden, da diese externe Faktoren sind



URSACHEN FÜR SCHWACHSTELLEN - TEIL 1

1. Voranschreiten der Vernetzung und Digitalisierung

- Mehr Software führt zu mehr potentiellen Fehlern
- Stärkere Vernetzung führt zu mehr/neuen Angriffsmöglichkeiten

2. Komplexität der Systeme

- Detailgrad ist zu hoch für den kompletten Überblick
- Auswirkungen von kleinen Änderungen sind schwer vorhersehbar

3. Haftungskomplexität

- Haftung und Sicherungsgarantien werden über Lieferketten ausgelagert

4. Vielfältige der Sicherheitsmechanismen und -technik

- Gut klingende aber ineffektive Sicherheitsmaßnahmen
- Effektive und akkurat hinreichende Mechanismen werden übersehen

URSACHEN FÜR SCHWACHSTELLEN - TEIL 2

5. Schatten-IT und veraltete Systeme

- Vergessene (und ungewartete) System werden zum Einfallstor

6. Fehlende Standardisierung

- Sicherheitsmechanismen meist markttypisch umgesetzt
- Fehlender marktübergreifender Standard

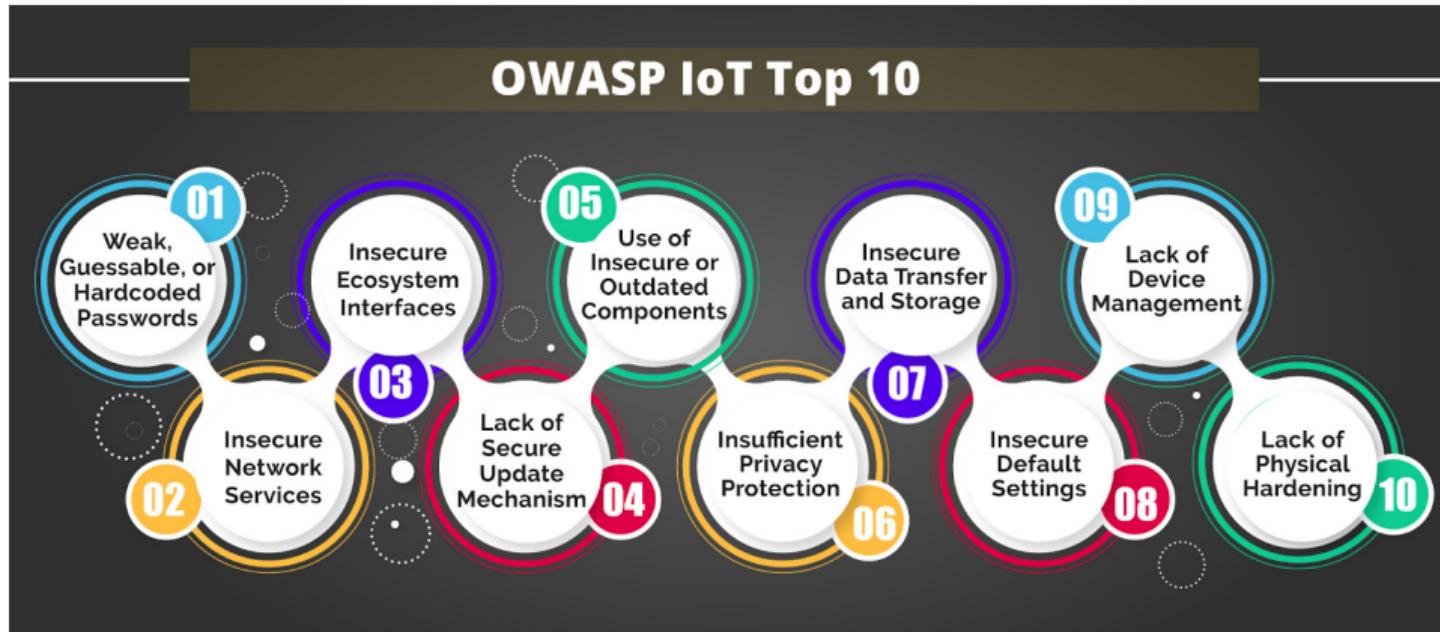
7. Produktions- und Wirtschaftsdruck

- Kürzere Entwicklungs- und Produktionszyklen, weniger Personal
- Wenig Geld für "nicht sichtbare" Sicherheits-"Features"
- Bei neuen Produkten/Technologien fehlt die Erfahrung

8. Sicherheit nachrüsten ist schwerfällig

- Benutzungskomplexität steigt, führt zu Ablehnung bei den Benutzern
- Ein komplettes Redesign wird oft aus Kostengründen abgelehnt

BEISPIEL FÜR ``BELIEBTE'' SCHWACHSTELLEN IN IOT



Quelle: <https://www.appsealing.com/owasp-iot-top-10/> - letzter Besuch 22.03.23

URSACHEN FÜR STEIGENDE BEDROHUNGEN

1. Professionalisierung der Angreifer

- Etablierung von business- und vertriebsartigen Strukturen bei den Angreifern
- Modularisierung von Angriffen und bereitstellen von Angriffframeworks
- Bereitstellen von Angriffen und die Durchführung von Angriffen als eine Dienstleistung, z.B. RaaS

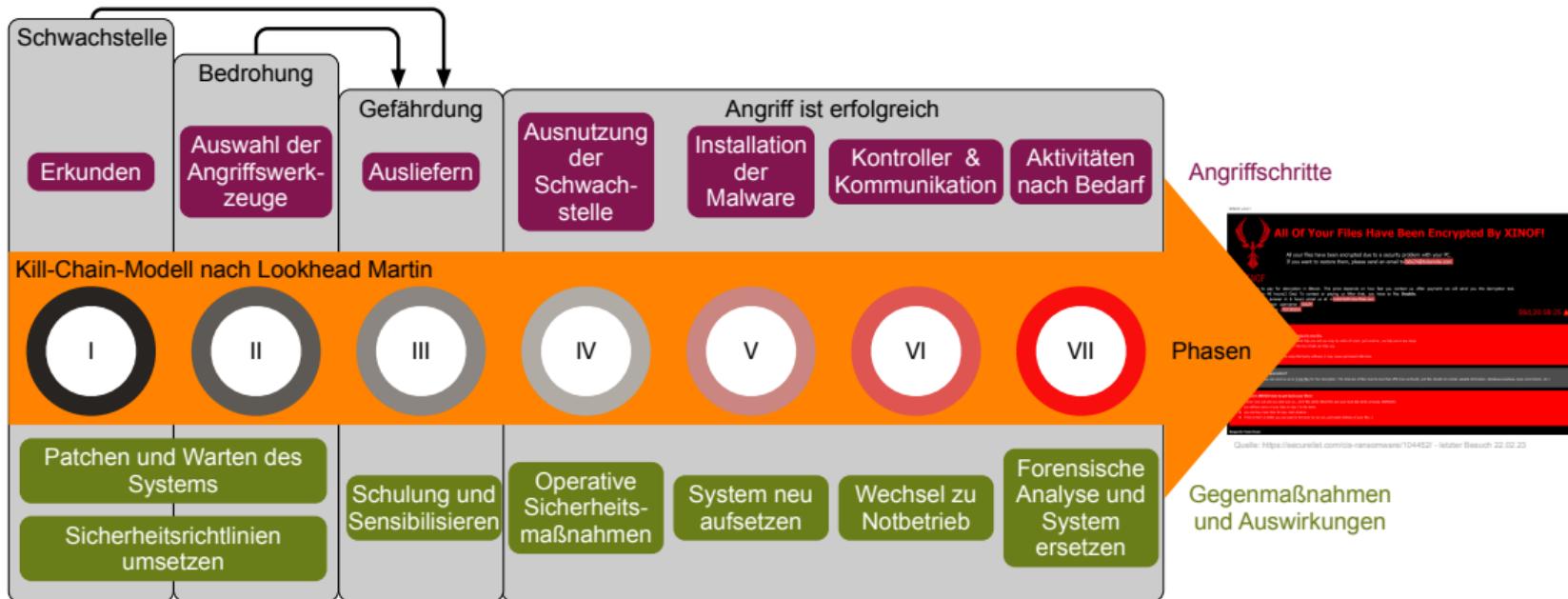
2. Steigerung der Automatisierung

- Automatische Scanner zur globalen Suche nach Schwachstellen
- Ausgereifte Schadsoftware verbreitet sich eigenständig weiter
- Sinkende menschliche Interaktion beim "Opfer"

3. Weltweit und Branchenübergreifend

- Strafverfolgung wird schwierig im Bezug auf die Ländergrenzen

VON DER SCHWACHSTELLE BIS ZUR MONETARISIERTEN GEFAHRDUNG



ASPEKTE ZUR ETABLIERUNG VON SECURITY

- Generelle Problematik beim langfristigen Absichern von Systemen:
 - Wachsende Komplexität der Systeme
 - Neue, variantenreiche und effektive Angriffe und Angriffsstrategien
- Schwachstellen können auf ganz interdisziplinären und sich ändernden Ursachen basieren
 - technische
 - organisatorische, prozessuale
 - anwendungsspezifische
 - rechtliche
 -

ASPEKTE BEI DER UMSETZUNG UND IMPLEMENTIERUNG

Beispiel:

Implementierung eines Programms mit denen Dateien beliebig verschlüsselt werden können, zum Schutz gegenüber Dritter.

- Welcher Verschlüsselungsalgorithmus wird gewählt?
schwache Sicherheitsparameter
- Wie wird sichergestellt, dass der Benutzer sich nicht vertippt bei der Passwort Eingabe?
Potentielle Benutzerfehler
- Verwaltung der geheimen Schlüssel, wie werden diese gespeichert?
Implementierungsschwachstelle
- Welcher Prozess darf die Verschlüsselung und Entschlüsselung durchführen?
Missbrauch bei Schadsoftware

REALISIERUNG VON SECURITY

Grundsätzliche Herangehensweise:

- Ein adäquates Schutzniveau mit verfügbaren Ressourcen erreichen
- Das nötige Sicherheitswissen in relevanten Bereiche propagieren
- Security kontinuierlich adaptieren und überwachen

Den universellen
Security-Experten
gibt es nicht

Security ist eine Systemeigenschaft

- Je nach Tätigkeitsfeld andere Berührungspunkte
- Schwachstellen finden und beheben benötigt Domänen-Expertise
- Klare Kommunikation der Konsequenzen wird benötigt



Basiert auf: https://www.mein-bezirk.at/schaerding/c-lokales/gibt-es-den-wolpertinger-wirklich_a1109097/ - letzter Besuch 22.03.23

ZUSAMMENFASSUNG

- Security \neq Safety
- Je mehr ein System potentiellen Gefährdung ausgesetzt ist, desto mehr kann es zu potentiellen Sicherheitsvorfällen kommen
- Angriffsstrategien und Angreiffer werden professioneller
- Schwachstellen können auf ganz interdisziplinären und sich ändernden Ursachen basieren
- Security ist eine Systemeigenschaft



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Grundlagen und Definitionen

April 15, 2023

Marc Stöttinger



Im Mittelpunkt jeder Sicherheitsbetrachtung steht menschliches Handeln und Unterlassen.

Sebastian Klipper

WIEDERHOLUNG: SECURITY VS. SAFETY

Tauschen Sie sich mit Ihrem Sitznachbar 3 Minuten aus:

- Überlegen Sie am Beispiel eines Autos oder Getränkeautomatens, wie jeweils ein Security-Vorfall und ein Safety-Vorfall aussehen könnte.
- Können Sie auf Basis der beiden Vorfälle eine generelle Aussage formulieren, die den Unterschied zwischen Security und Safety klar stellt?

DEFINITION VON IT-SICHERHEIT

IT-Sicherheit

"IT-Sicherheit beschäftigt sich mit der Absicherung von technischen Systemen durch angemessene Maßnahmen auf ein tragbares Maß." - BSI

IT-Sicherheit beschränkt sich in der Regel auf die Absicherung informationstechnischer Systeme:

- Netzwerk
- Server
- eMail
- ...



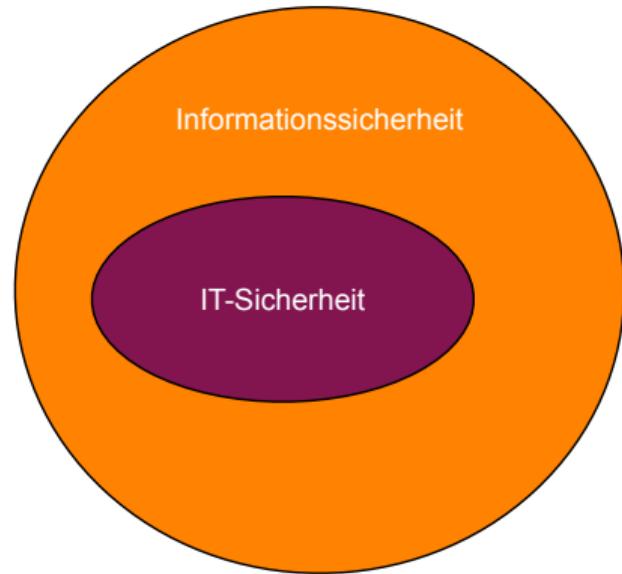
DEFINITION VON INFORMATIONSSICHERHEIT

Informationssicherheit

"Informationssicherheit beschäftigt sich mit der Sicherheit von technischen oder nicht-technischen Systemen zur Informationsverarbeitung, -speicherung und -lagerung." - BSI

Informationssicherheit betrachtet die Sicherung von Informationen generell:

- Verschlussakten
- Personenkontrolle
- Geschäftsmodelle
- ...



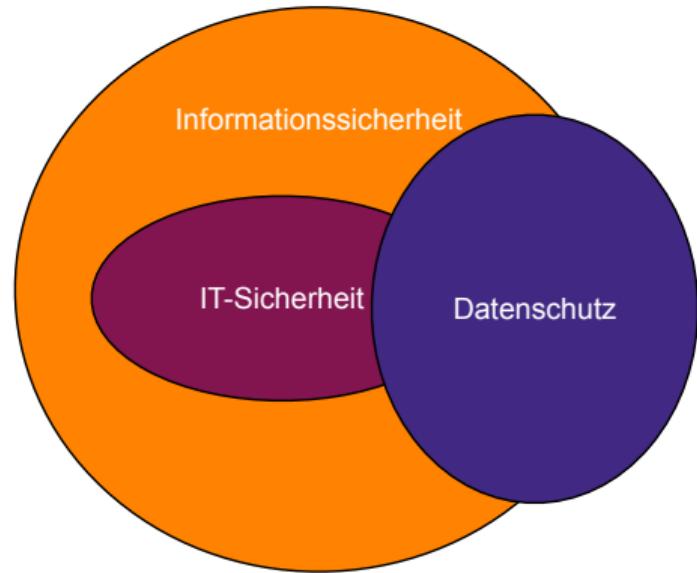
DEFINITION VON DATENSCHUTZ

Informationssicherheit

“Datenschutz soll das Individuum davor schützen, durch den Umgang mit den eigenen personenbezogenen Daten im Persönlichkeitsrecht beeinträchtigt zu werden. Mit Datenschutz wird daher der Schutz personenbezogener Daten vor etwaigem Missbrauch durch Dritte bezeichnet.” - BSI

Zusätzliche Aspekte zum Schutz der Informationen:

- Rechte und Genehmigungen von Datenerhebung
- Verwendung und Löschung der erhobenen Daten
- ...

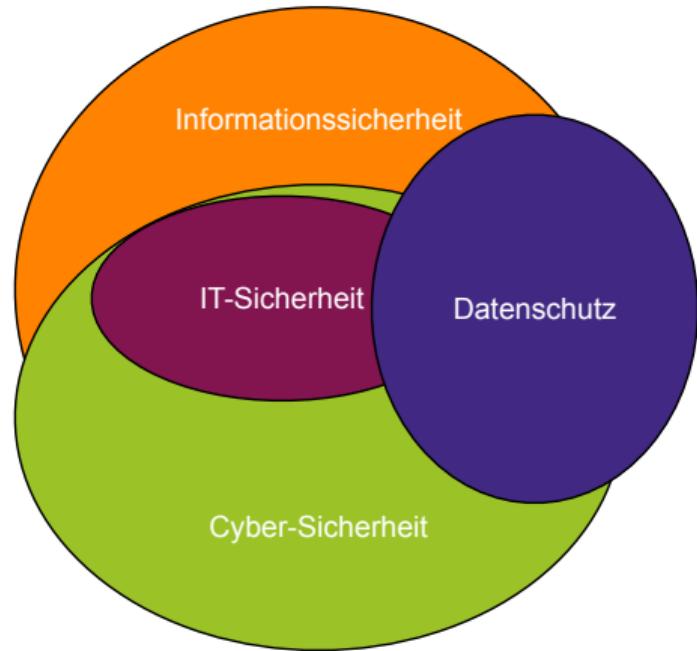


DEFINITION VON CYBER-SICHERHEIT

Informationssicherheit

"Cyber-Sicherheit befasst sich mit allen Aspekten der Sicherheit in der Informations- und Kommunikationstechnik. Das Aktionsfeld der Informationssicherheit wird dabei auf den gesamten Cyber-Raum ausgeweitet. Dieser umfasst sämtliche mit dem Internet und vergleichbaren Netzen verbundene Informationstechnik... ." - BSI

Cyber-Sicherheit fokussiert sich auf mit dem Internet verbundene Geräte und erweitert den Sicherheitsbegriff auf gesellschaftliche Werte.



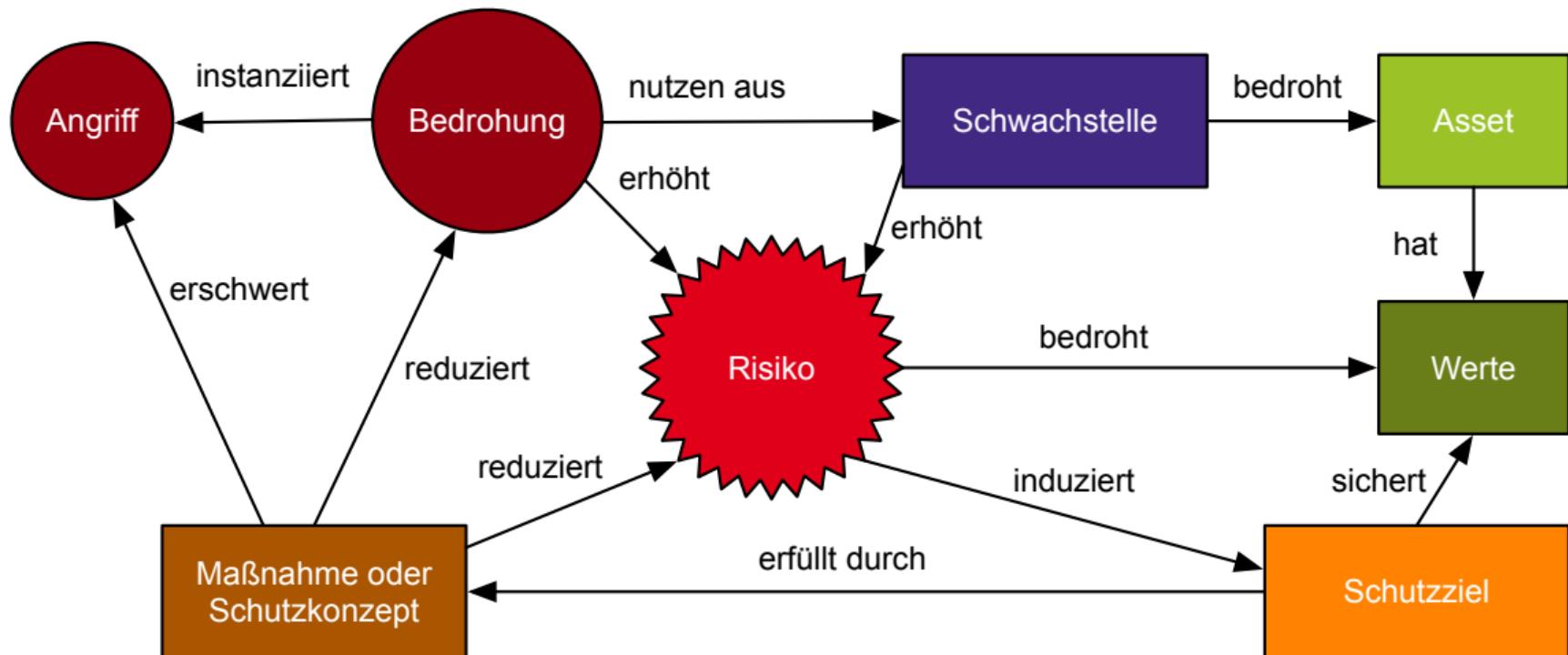
BEGRIFFLICHKEIT IN DER SECURITY

Wie Sie gesehen haben, gibt es verschiedene Bereiche mit **verschiedenen Schwerpunkten** im Kontext von Security. In den vier Bereichen gibt es **gemeinsame Begriffe**, um Sachverhalte zum Thema **Angriffe** und **Schutz** genauer zu spezifizieren.

In einem risikobasierten Ansatz ist es möglich, mit diesen Begriffen Angriffe zu quantifizieren, um beispielsweise folgende Fragen zu beantworten:

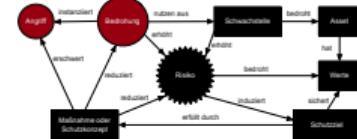
- Was muss ich vor einem Angriff schützen?
- Wie wahrscheinlich ist es, dass ein Angriff stattfindet?
- Was bedeutet es, wenn der Angriff erfolgreich ist?
- Wie verhindere ich, dass ein Angriff erfolgreich ist?

RISIKO-ZENTRISCHE SICHERHEITSBEGRIFFE



ANGRIFF UND BEDROHUNG

Ein Angriff ist eine Instanziierung einer Bedrohung, welche auf Basis konkreter Techniken und Vorgehensweisen eine Schwachstelle ausnutzen will.

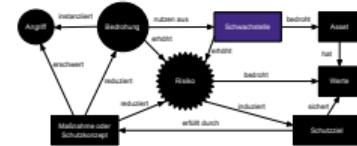


Oft wird keine Unterscheidungen von Bedrohungen und Angriffen gemacht. Jedoch kann dies nützlich sein, wenn viele potentielle Angriffe auf eine Schwachstelle existieren und diese durch eine Bedrohung zusammengefasst werden können.

- Beispiel: Für ein System existiert die Bedrohung, dass das Passwort gebrochen wird, da es nur aus 8 Zeichen besteht.
 - Einfacher Brutforce-Angriff
 - Wörterbuch-Angriff
 - Passwort-Spraying
 - Phising-Angriff
 - Keylogger-Angriffe

SCHWACHSTELLEN

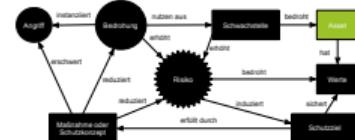
Schwachstellen sind eine Gefährdung für die Assets. Schwachstellen können somit eine Gefährdung für das System darstellen, wenn diese im Rahmen eines Angriffs ausgenutzt werden.



→ Siehe Vorlesung **Einführung** Folie 10 bis 12.

ASSETS

Eine Asset ist jede Komponente, jedes System, alle Daten, jede Anwendung oder jede Ressource, die für ein System, ein Unternehmen oder eine Organisation von immenser Bedeutung ist und geschützt werden muss.



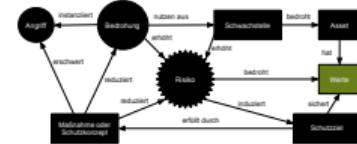
Ein Asset ist alles (materielle oder immaterielle) von besonderem Wert.

- Datenbestände
- Businessplan
- Schlüssel

In manchen Fällen wird auch noch zwischen primären und sekundären Assets unterschieden. Sekundäre Assets hängen von primären Assets ab.

SCHÄDEN UND WERTE

Bei Verlust des ursprünglichen Werts eines Assets hat dies Auswirkungen auf das System, Unternehmen oder die Organisation. Der Schaden der mit dem Wertverlust einhergeht kann sich verschieden stark in verschiedenen Bereichen auswirken.



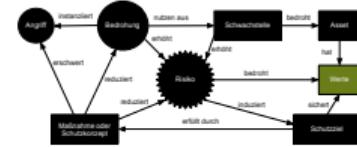
Schadensklassifikation nach BSI IT-Grundschutz:

- normal - Auswirkung ist begrenzt
- hoch - Auswirkung sind beträchtlich
- sehr hoch - Auswirkung ist existenziell bedrohlich und katastrophal

Beschreibung der Klassifizierung muss für jede betrachtete Schadenskategorie definiert sein. Klassifizierung und Schadensstufen sind oft domänen spezifisch.

SCHÄDEN UND WERTE

Bei Verlust des ursprünglichen Werts eines Assets hat dies Auswirkungen auf das System, Unternehmen oder die Organisation. Der Schaden der mit dem Wertverlust einhergeht kann sich verschieden stark in verschiedenen Bereichen auswirken.

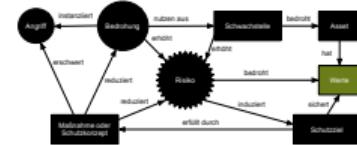


Schadenskategorie im BSI IT-Grundschutz:

- Verstoß gegen Gesetze/Vorschriften/Verträge
- Beeinträchtigung des informationellen Selbstbestimmungsrechts
- Beeinträchtigung der persönlichen Unversehrtheit
- Beeinträchtigung der Aufgabenerfüllung
- Negative Innen- oder Außenwirkung
- Finanzielle Auswirkungen

SCHÄDEN UND WERTE

Bei Verlust des ursprünglichen Werts eines Assets hat dies Auswirkungen auf das System, Unternehmen oder die Organisation. Der Schaden der mit dem Wertverlust einhergeht kann sich verschieden stark in verschiedenen Bereichen auswirken.



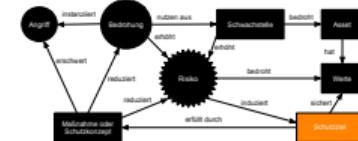
Schadenskategorie im Bereich Automotive nach der ISO21434:

- Safety Schaden
- Finanzieller Schaden
- Operativer Schaden
- Privatsphärenschaden

SCHUTZZIELE

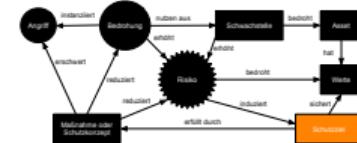
Schutzziele definieren abstrakte Sicherheitsanforderungen an ein Asset. Das Schutzziel muss erfüllt sein, damit das Asset nicht sein Wert verliert. Diese Verletzung kann sich auch auf die ursprünglichen Eigenschaften des Systems sich auswirken.

- Jedes Schutzziel konkretisieren den abstrakten Sicherheitseigenschaft die wichtig für die Eigenschaft und die Funktionalität des Assets ist.
- Jede Sicherheitseigenschaft eines Schutzzieles kann einen generische Art von Bedrohung gegenüber gestellt werden.
- Die Anzahl der Schutzziel und deren art hängt von dem jeweiligen Schutzzielmodell ab.



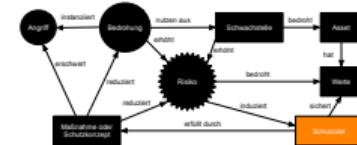
SCHUTZZIELE IM DETAIL (1/3)

- **Vertraulichkeit:** Schutz vor unbefugter Preisgabe von Informationen.
 - Veröffentlichung privater Bankdaten
- **Integrität:** Sicherstellung der Korrektheit (Unversehrtheit) von Daten und der korrekten Funktionsweise von Systemen.
 - Ändern des Betrages bei einer Paypal-Überweisung
- **Authentizität:** Kommunikationspartner ist tatsächlich diejenige Person, die sie vorgibt zu sein bzw. die Informationen wurden tatsächlich von der angegebenen Quelle erstellt.
 - Senden von Messenger-Nachricht unter falschem Namen



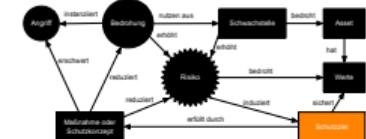
SCHUTZZIELE IM DETAIL (2/3)

- **Verfügbarkeit:** Sicherstellung des vorgesehenen Nutzbarkeit eines IT-Systems.
 - Stören der Compass Systems
- **Autorisierung:** Freischaltung der eingeräumten Rechte für eine erfolgreich authentifizierte Person oder Identität.
 - Cheat-Code im Spiel eingeben
- **Nicht-Abstreitbarkeit:** Empfangen/Senden einer Nachricht oder Durchführen einer Aktivität kann nicht abgestritten werden.
 - Beschuldigung des Autopiloten am Absturz des Flugzeugs



SCHUTZZIELE IM DETAIL (3/3)

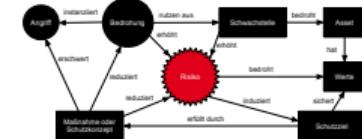
- Verschiedene gängige Sicherheitsziel-Modelle:
 - **CIA**: Confidentiality, Integrity, Availability
 - **CIAA**: CIA + Authenticity
 - **STRIDE**: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privileges



Sicherheitsziel	Sicherheitsziel (eng.)	Bedrohung	Bedrohung (eng.)
Vertraulichkeit	Confidentiality	Unbefugtes Auslesen	Information Disclosure
Integrität	Integrity	Manipulation	Tampering
Authentizität	Authenticity	Fälschen	Spoofing
Verfügbarkeit	Availability	Störung des Betriebs	Denial-of-Service
Autorisierung	Authorization	Erhöhung von Rechten	Elevation of Privileges
Verbindlichkeit	Non-Repudiation	Abstreiten von Aktionen	Repudiation

RISIKO

Alleine die Existenz einer Schwachstelle mit einer potentiellen Bedrohung führt nicht zu einer 100% Ausnutzung. In der Regel wird für einen Angriff der Weg des geringsten Widerstands gewählt.

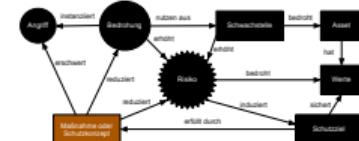


Das Risiko kann zur Priorisierung der Umsetzung der Maßnahmen genutzt werden

- Alle Maßnahmen umzusetzen steht oft nicht im Kosten/Nutzen-Verhältnis
- Systematische Betrachtung der gesamten Bedrohungslandschaft
- In die Risikobewertung geht sowohl die Wahrscheinlichkeit eines Angriffs ein als auch der zu erwartende Schaden auf das Asset oder Gesamtsystem

SCHUTZMASSNAHME

Maßnahmen können entweder technischer, prozessualer oder organisatorischer Art sein. Final wird mit der Maßnahme erreicht, dass Risiko soweit zu verringern, dass es akzeptierbar ist oder transferiert wird.

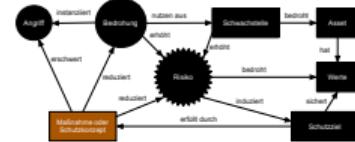


Maßnahmen für die verschiedenen Stadien eines Angriffs

- **Präventive Maßnahmen** - Abwehr des Angriffs und Erhalt der Schutzziele
- **Detektierende Maßnahme** - Maßnahme zur Detektion, wenn eine präventive Maßnahme den Angriff nicht abwehren konnte
- **Reaktive Maßnahme** - Operative Maßnahme zum Wiederherstellen des Soll-Zustandes nach Detektion eines Sicherheitsereignisses

SCHUTZMASSNAHME

Maßnahmen können entweder technischer, prozessualer oder organisatorischer Art sein. Final wird mit der Maßnahme erreicht, dass Risiko soweit zu verringern, dass es akzeptierbar ist oder transferiert wird.



Maßnahme	präventiv	detektierend	reakтив
prozessual und organisatorisch	Schulungen, Richtlinien, Vulnerability-Management	Audit, SOC	CERT-Team, Blue und Red Team, Security Response Prozess
technisch	Verschlüsselungs-technologie, Firewall, VirensScanner, DMZ	Intrusion Systeme	Reaktive DMZ und Backup-System

WIEDERHOLUNG: SECURITY VS. SAFETY

Tauschen Sie sich mit Ihrem Sitznachbar 5 Minuten aus:

- Identifizieren Sie zwei Assets von Ihrem Beispiel Heute morgen (Auto oder Getränkeautomat).
- Identifizieren Sie pro Asset mindestens zwei Schutzziele basierend auf potentiellen Bedrohungen.

ANREIFERMODELLE

Es ist wichtig für die Sicherheitsbetrachtung im Bezug auf potentielle Angriffe und Bedrohungen, verschiedene Arten von Angreifern zu berücksichtigen.

- Grundsätzlich geschieht ein Angriff immer aus einer Motivation heraus.

Angreifer

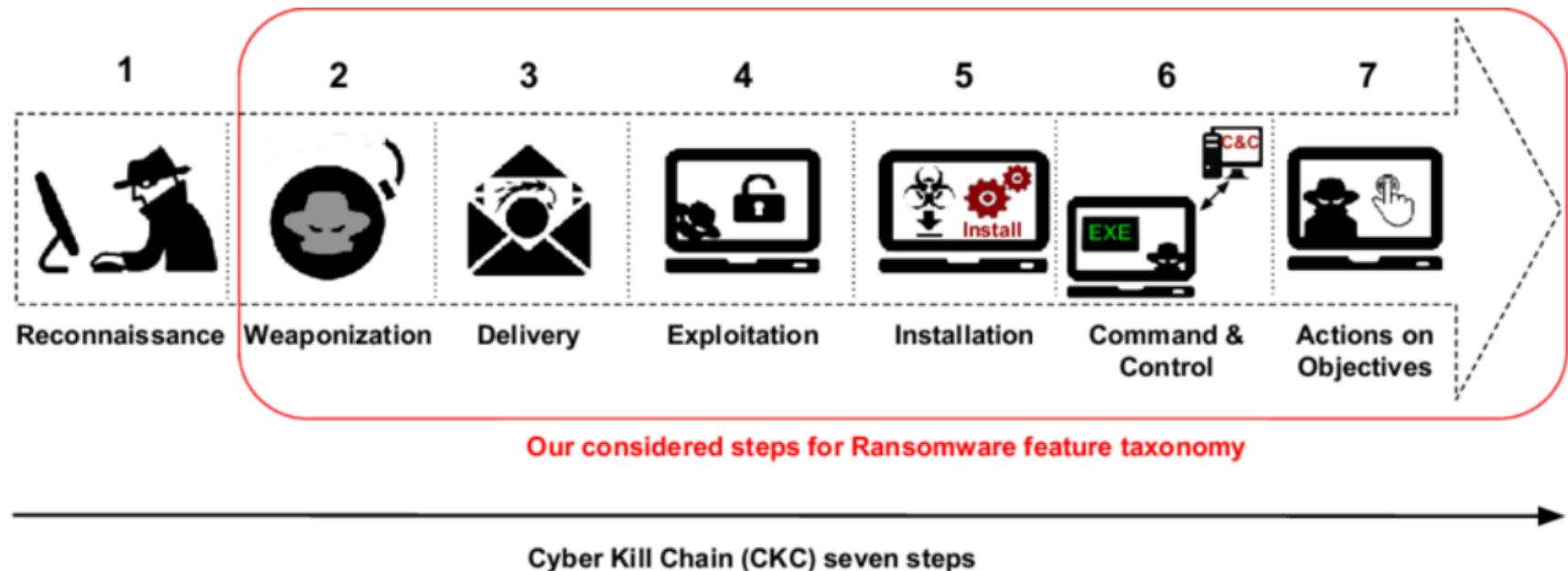
Ein Angreifer ist immer ein Mensch; auch Organisationen bestehen aus Menschen.

- Die Fähigkeiten und Ressourcen zum Durchführen eines Angriffs hängt auch vom Angreifertyp ab.
- Die Motivation der Angreifer ist wichtig, da diese Rückschlüsse auf potentielle Ziele preisgibt.

ANGREIFER UND IHRE MOTIVATION

Angreifer	Motivation	Vorgehen
Anwender	Persönlicher Vorteil	Anwendung von Tools oder Anleitungen; Anheuerung organisierter Krimineller
Mitarbeitende	Rache; Geld; Ideologie	Zugriff auf und Kompromittierung von internen Systemen
Ethische Hacker	Anerkennung; Herausforderung; Geld; Ethische Überzeugung	Identifikation von Schwachstellen; Ausnutzung unter ethischen Richtlinien
Hacktivisten	Anerkennung; Herausforderung; Politische oder ideologische Ziele; Vandalismus; Geld	Identifikation und Ausnutzung von Schwachstellen; Offenlegung des Eindringens
Kriminelle	Geld	Identifikation und Ausnutzung von Schwachstellen; Kompromittierung des Systems; Monetarisierung
Konkurrenz	Störung; Entwendung von Technologie; Diskreditierung	Reverse-Engineering von Produkten; Anheuerung organisierter Krimineller
Organisierte Kriminelle	Geld	Systematische Identifikation und Ausnutzung von Schwachstellen; Kompromittierung des Systems; Bereitstellen von Services; Monetarisierung
Staaten / Geheimdienste	Wirtschaftliche Vorteile, Destabilisierung	Kompromittierung der Infrastruktur, Komponenten oder Standards; Tarnung vor Entdeckung

VORGEHENSWEISE NACH DER CYBER KILL CHAIN VON LOCKHEED MARTIN



Quelle: T. Dargahi et al., A Cyber-Kill-Chain based taxonomy of crypto-ransomware features. Journal of Computer Virology and Hacking Techniques, Springer, 2019

CYBER KILL CHAIN - RECONNAISSANCE: ZIELE SUCHEN

Aktivität der Angreifer:

- Gezieltes Suchen nach Angriffszielen und sammeln von Informationen über einen längeren Zeitraum.
- Aktives Afangen von Daten und Ausnutzen menschlicher Schwächen.
- Oft werden offene Tools wie Nmap und OpenVAS oder selbst entwickelte Tools verwendet.
- Ebenso werden Informationen aus dem Darknet genutzt, um Angriffe vorzubereiten.

Mögliche Abwehr:

- Die Reconnaissance ist schwer zu erkennen, wenn sie ausgeführt wird.
- Unauffälliges Verhalten in Social Media. Prüfen ob Accounts auf Leaking-Plattformen wie "haveibeenpwned" auftauchen

CYBER KILL CHAIN - WEAPONIZATION: AUSWAHL DES WERKZEUGS

Aktivität der Angreifer:

- Gezieltes Suche nach Schwachstellen im Unternehmen.
- Nutzen leicht zugänglicher Angriffstools oder -anleitungen oder Erstellen eigener Tools mit Hilfe von Informationsmaterialien.
- Botnets werden oft gemietet, um temporäre oder dauerhafte Angriffe durchzuführen.
- Zusätzlich können auf dem Schwarzmarkt erwerbliche Zero-Day-Exploits zum Angriff genutzt werden.

Mögliche Abwehr:

- Keine Gegenmaßnahme möglich

CYBER KILL CHAIN - DELIVERY: AUSLIEFERN

Aktivität der Angreifer:

- Ausliefern der Malware durch:
 - Spam-Mail, Phishing-Mails
 - Dateien in Antworten auf offizielle Anfragen (Bewerbungen und Angebote)
 - Dateien und Links in Social Media
 - Präparierte Webseiten

Mögliche Abwehr:

- Awareness-Schulungen des Personals, Umsetzen von Sicherheitsrichtlinien
- Verwenden von Antivirus Software, Email- und Webfilter

CYBER KILL CHAIN- EXPLOITATION: AUSNUTZEN DER SCHWACHSTELLE

- Ausnutzen von Schwachstellen, um später vollständigen Zugang zum Unternehmensnetzwerk zu erhalten.
- Dies kann entweder still erfolgen oder direkt zu einer aktiven Beeinflussung der produktiven Systeme führen.
- Angreifer nutzen neben technischen Werkzeugen oft auch Social Engineering, um Zugang zu erhalten.

Mögliche Abwehr:

- Aktives Patch-Management, um das System up-to-date zu halten
- Durchführen von Vulnerability Management mit automatischen Schwachstellenscans im System

CYBER KILL CHAIN - INSTALLATION: ZUGANGSPERSISTIERUNG

- Persistieren der Malware und des Zugriffs auf das System.
- Oft werden Backdoors installiert und Reverse Shell genutzt und etabliert.
- Translative Bewegung im System zum Auskundschaften des Netzwerks durch den dauerhaften Zugang zum infizierten System.
- Ein Advanced Persistent Threat (APT) kann sich über einen längeren Zeitraum im System verstecken und dadurch auch Lieferanten und Kunden in der Lieferkette infiltrieren.

Mögliche Abwehr:

- Sicherheitslösungen zur Erkennung, Protokollierung oder Whitelisting von erlaubter Software
- Mehrfaktor-Autentifizierung, gutes Rechtemanagement

CYBER KILL CHAIN - C2: KONTROLL UND KOMMUNIKATION

- Verdeckte Kommunikation zu einem Control und Command Server wird aufgebaut um:
 - Sensible Informationen und Daten zu exfiltrieren; werden oft im Darknet per Erpressungsversuch zum Rückkauf angeboten
 - Nachladen von weiterer Schadsoftware und zur "Pflege" und Wartung der bereits installierten Malware
 - Nutzen des infizierten Systems, um weitere Angriffe auf andere Systeme zu starten

Mögliche Abwehr:

- Unterbinden der Kommunikation zum CC-Server mit Firewalls, Intrusion Detektion Systemen
- Wechseln in den Notbetrieb (nur unbedingt notwendige Systeme und Kommunikationskanäle sind nutzbar), um den Schaden zu minimieren

CYBER KILL CHAIN - ACTION ON OBJECTIVE: AKTIVIERUNG NACH BEDARF

- Je länger die Persistierung im System stattfindet desto mehr Aktivitäten können die Angreifer durchführen
- Typische Aktivitäten zur Monetarisierung sind:
 - Veröffentlichung von sensiblen Daten
 - Verschlüsselung oder Manipulation von Daten
 - Verwendung des infizierten Systems als Bot in einem Botnetzwerk

Mögliche Abwehr:

- Durchführen einer IT-forensischen Analyse zur Rekonstruktion des Angriffs
- Backup-System und ein Notfallmanagementplan haben (Business Continuity Management und Incident Response Prozess)

ATT&CK-FRAMEWORK

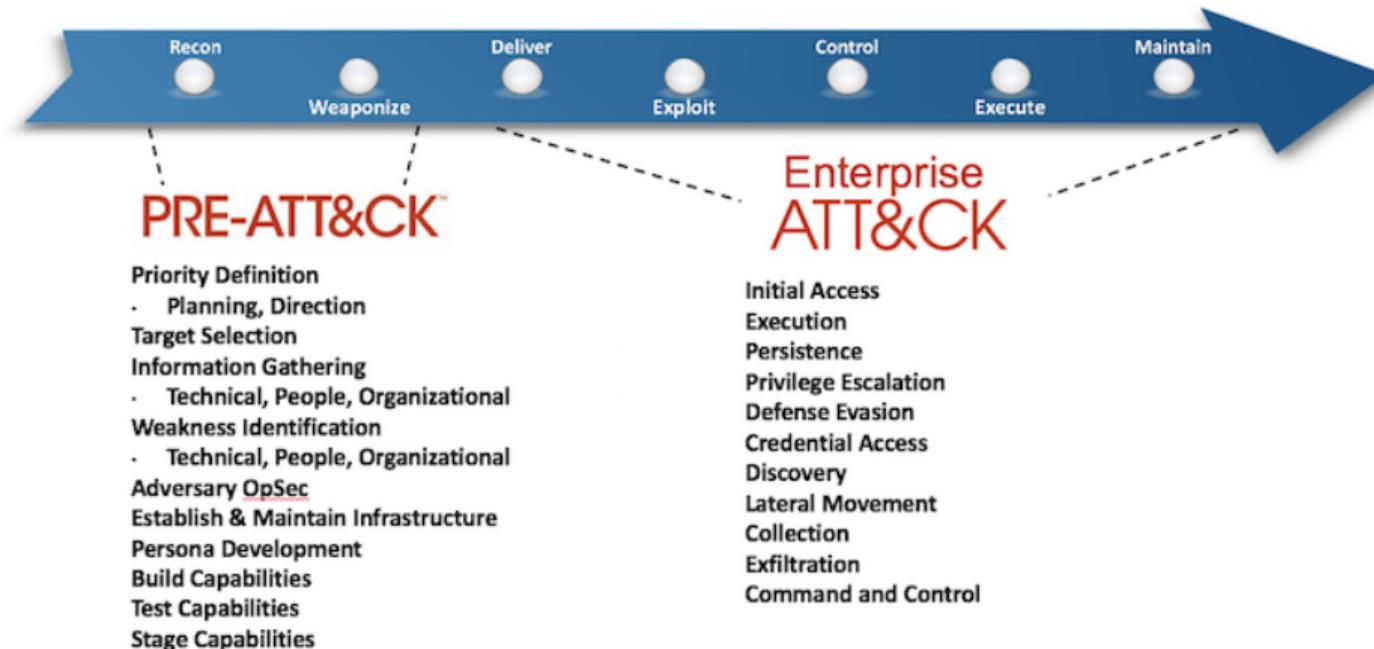


Figure: Att&ck-Framework

Quelle: <https://www.ecrimelabs.com/blog/2020/4/5/mitre-attampck-for-improved-metrics-and-kpi-on-detection-capabilities> - letzter Besuch 26.03.2023

ZUSAMMENFASSUNG

- Angreifer sind immer Menschen oder Organisationen
- Alle Sicherheitsarten (IT, Cyber, Informartionssicherheit) haben Gemeinsamkeiten aber auch Schwerpunkte
- Schutzziel und Gegenmaßnahmen können mit einem risikobasierten Ansatz zur Abwehr von Bedrohungen identifiziert und priorisiert werden
- Es gibt verschiedene Angreifertypen mit unterschiedlicher Motivation
- Alle Schritte eines Angriffes auf IT-Systeme lassen sich generische durch die Phasen der Cyber Kill Chain abbilden



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Standards und ISMS

April 27, 2023

Marc Stöttinger



Security is a process not a product.

Bruce Schneier

MOTIVATION SICHERHEITSSTANDARDS

→ **Bisher:** Identifikation von Bedrohungen und Vorgehen von Angreifern

→ **Aber:**

- Wo fangen wir an, IT-Sicherheit umzusetzen?
- Wo hören wir auf, IT-Sicherheit umzusetzen?
- Wie stellen wir eine sinnvolle Umsetzung sicher?
- Wie kommunizieren wir IT-Sicherheit intern/extern?

→ **Beispiel:**

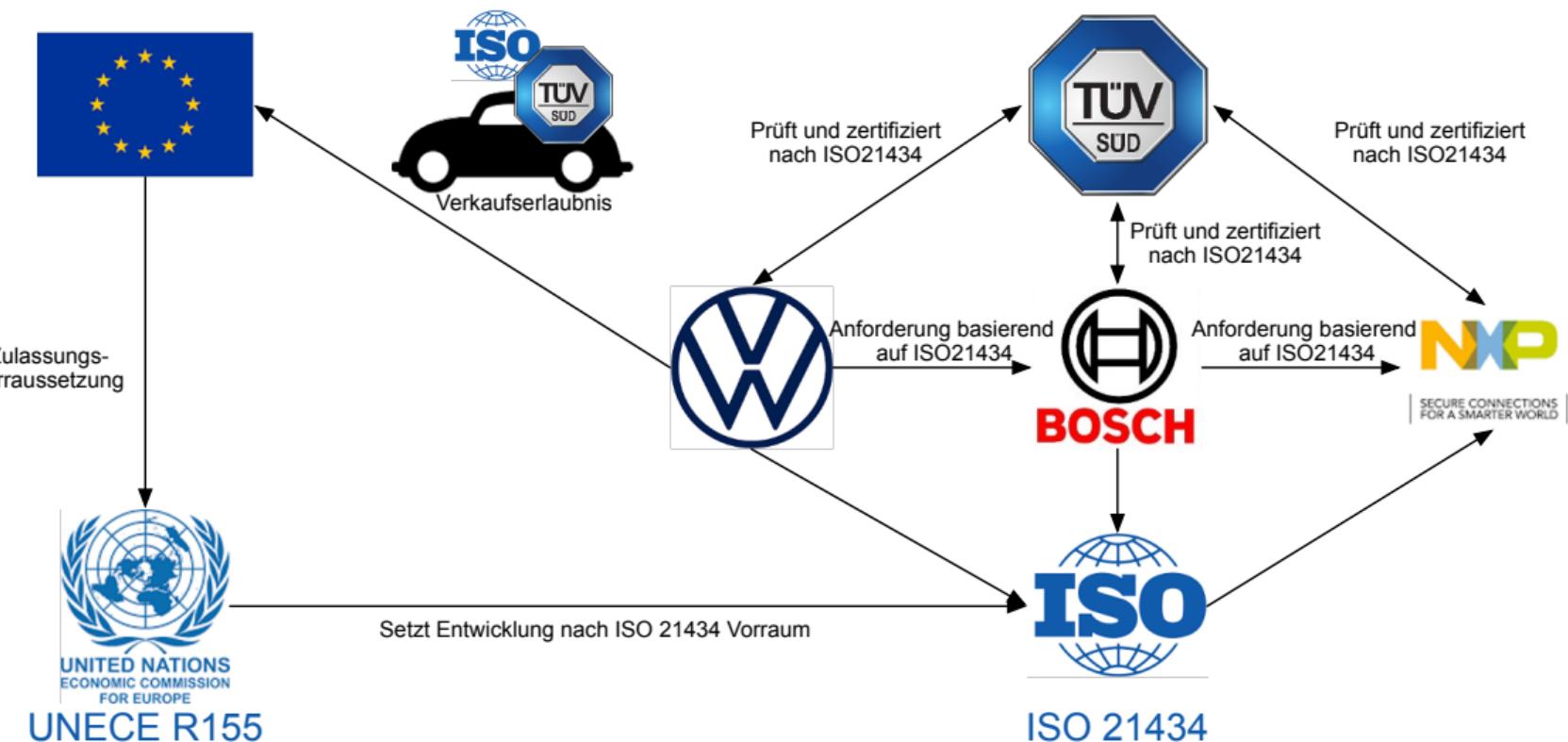
- Entwicklung der IT-Sicherheit für Fahrzeuge im Fall von [FCA] Jeep Cherokee-Hack in 2015



BEISPIEL IM FALL JEEP CHEROKEE-HACK

1. Sicherheitsforscher analysieren Multimedia System und WLAN Interface
 - Schwachstellen existieren, um wahlfreie Befehle auf dem Multimedia System auszuführen
 - Schwachstellen können via GSM ausgenutzt werden
 - Ca. 300.000 anfällige Jeeps werden via GSM identifiziert
2. Weitere Sicherheitslücken identifiziert, um wahlfreie Nachrichten im Fahrzeugnetzwerk zu senden
 - Fahrzeugnetzwerk enthält: Bremsen, Lenkung, Türsteuerung, ...
 - Senden von Nachrichten an Fahrzeugnetzwerk ist möglich via GSM
3. Sie demonstrieren den Angriff via Remote Hack mit Reportern am Steuer
4. Rückrufaktion zum Patchen der Fahrzeugsoftware kostet FCA 1.4 Millionen Dollar

VERANTWORTUNGSKETTE IT-SICHERHEIT IM BEREICH AUTOMOTIVE



RECHTSFORMEN ZUR IT-SICHERHEIT

- Rechtsnormen mit Fokus auf IT-Sicherheit sind u.a.:
 - IT-Sicherheitsgesetz 2.0 (IT-SiG 2.0)
 - EU-Datenschutzgrundverordnung (DSGVO/GDPR)
- Viele weitere Rechtsnormen enthalten Vorgaben zum Thema IT-Sicherheit
 - Telekommunikationsgesetz (TKG): Verbot des Abhörens oder Veränderns von Kommunikation durch TK-Diensteanbieter
 - E-Health-Gesetz: Absicherung des Netzwerkes für medizinische Datenkommunikation
 - ...
- Anforderungen zur Erfüllung von Rechtsnormen können sowohl direkt vom Gesetzgeber als auch transitiv vom Kunden erhalten werden

IT-SICHERHEITSGESETZ 2.0

- Das **IT-Sicherheitsgesetz 2.0** (IT-SiG 2.0) von 2021
 - Umfasst Kritische Infrastrukturen (Energie, Gesundheit, Ernährung,)
 - Das BSI fungiert als zentrale Prüf- und Kontrollbehörde
 - Bis zu 2 Mio. Euro Bußgeld bei vorsätzlich fahrlässiger Handlung
- Das IT-SiG definiert verschiedene Anforderungen an Organisationen
 - Verfahren zur Angriffserkennung müssen umgesetzt werden
 - IT-Sicherheitsvorfälle müssen dem BSI gemeldet werden
 - Eingekaufte kritische Komponenten müssen vom Innenministerium genehmigt werden
 - Technische Maßnahmen nach branchenspezifischen Sicherheitsstandards (B3S) werden empfohlen

DATENSCHUTZGESETZE (DSGVO UND GDPR)

- Die Datenschutzgrundverordnung (DSVGO, Englisch GDPR) verlangt u.a:
 - **Zweckbindung:** Nur benötigte private Daten dürfen erhoben und verarbeitet werden
 - **Speicherbegrenzung:** Daten müssen gelöscht werden, wenn der Zweck verfällt
- Nutzer haben ein Rechte auf:
 1. **Information** zur Erhebung und Verarbeitung privater Daten
 2. **Zugriff, Änderung und Löschung** der gespeicherten privaten Daten
 3. **Einschränkung und Mitnahme** der gespeicherten privaten Daten
 4. **Widerspruch** gegen die Speicherung privater Daten
 5. **Vermeidung automatisierter Entscheidungsfindung** basierend auf privaten Daten
- Unternehmen müssen gespeicherte private Daten gegen Angriffe schützen und sind für Schäden haftbar

FOLGEN EINER DSGVO VERLETZUNG

- Verletzungen des DSGVO werden mit **bis zu 4%** des jährlichen Einkommens geahndet [Fine, ENF]:

Höhe [Euro]	Angeklagter	Grund
405.000.000	Meta	Instagram Daten von Kindern nachlässig behandelt (z.B. Profil standardmäßig öffentlich)
35.258.708	H&M	Erfassung privater Urlaubs- und Gesundheitsdaten von Mitarbeitenden
50	Privatperson	Unerlaubter Einsatz einer Dashcam

- Verarbeitung privater Daten im Unternehmen muss kontrolliert werden:
- Bewusstsein der Mitarbeitenden für Umgang mit privaten Daten
 - Zentrales und sicheres Speichern privater Daten
 - Kontrolle und Protokollierung des Zugriffs auf private Daten
 - Review der erhobenen Daten sowie der Konzepte zur Sicherung mit Juristen

RECHTSNORMEN UND STANDARDS

- Rechtsnormen (z.B. UNECE R155) sind verpflichtende Richtlinien
 - Gesetzgebung ist ein langwieriger Prozess
 - Rechtsnormen können den "Stand der Technik" nicht zeitnah abbilden
- Standards (z.B. ISO21434) sind empfehlende Richtlinien
 - Bilden den "Stand der Technik" einer Branche ab
 - Erlaubt Unternehmen einer Branche eine effiziente Prüfung auf Einhaltung von Anforderungen
 - Standardisierung kann "relativ" flexibel durch Unternehmen einer Branche angepasst werden
- Rechtsnormen verweisen häufig auf umzusetzende Standards

UNECE R155



Verweist auf
Einhaltung



ISO 21434

STANDARDS IM BEREICH IT-SICHERHEIT

- Es existieren verschiedene Standards in der IT-Sicherheit
- IT-Sicherheit im Unternehmen:
 - **ISO/IEC27000 Familie:** Anforderung und Implementierung eines ISMS
 - **BSI Grundschutz (BSI 200-x):** Empfehlungen zu Methoden, Prozessen und Prozeduren
 - **B3S:** Branchenspezifische Sicherheitsstandards im Rahmen des IT-SiG 2.0
- IT-Sicherheit für Produkte:
 - **Common Criteria:** Sichere Produktentwicklung und Anforderungen an Zertifizierung
 - **ETSI EN 303 645:** Sichere Entwicklung von IoT Geräten
 - **ISO21434:** Sichere Entwicklung von Fahrzeugen

EINSCHUB: INFORMATIONSSICHERHEITS-MANAGEMENTSYSTEM (ISMS)

- Grundphilosophien der IT-Sicherheit
 - IT-Sicherheit muss an Unternehmen angepasst und regelmäßig überprüft werden
 - IT-Sicherheit muss sowohl auf technischer- als auch auf Prozessebene implementiert werden
- Ein **ISMS** ist ein System zur Definition, Überprüfung, Erhalt und Verbesserung der IT-Sicherheit
 - Betrachtet sowohl technische Maßnahmen als auch Prozesse
 - Wird von der Unternehmensleitung vorgegeben
 - Wird auf das gesamte Unternehmen angewendet
- Ein ISMS nutzt den **Plan-Do-Check-Act (PDCA)** Zyklus zur ständigen Verbesserung

WARUM EIN ISMS?

- Sicherheit ist kein Zustand sondern ein Prozess
 - Sicherheit unterliegt einer kontinuierlichen Dynamik (Änderung von Gesetzen, neue Angriffe oder technischer Fortschritt)
- Sicherheit muss aktiv gewartet, aufrecht erhalten und verbessert werden
 - Systemeinführung planen
 - Sicherheitsmaßnahmen definieren und umsetzen
 - Erfolgskontrollen durchführen
 - Schwachstellen und Verbesserungsmöglichkeiten finden
 - Maßnahmen verbessern
 - Sicherheitsaspekte bei Außerbetriebnahme berücksichtigen



ISMS RELEVANTE KOMPONENTEN UND STANDARDS

→ Komponenten

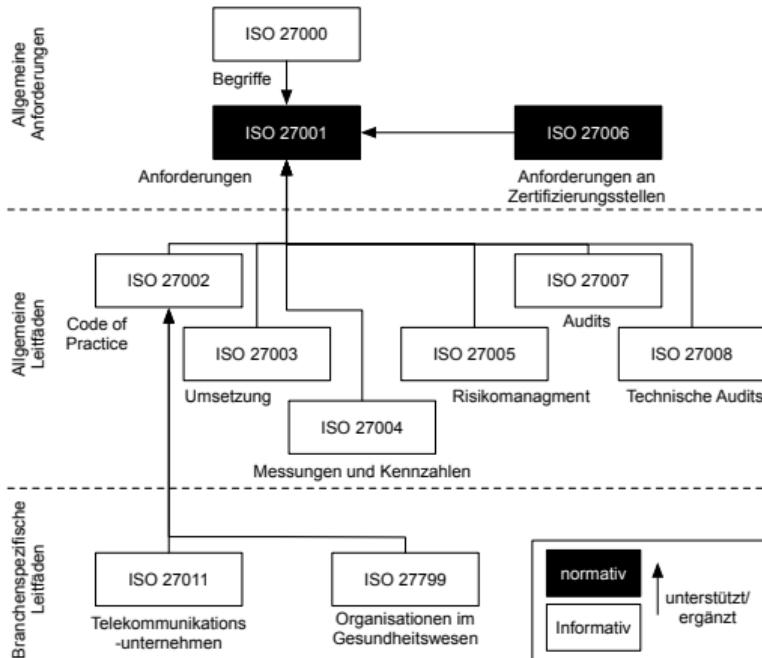
- Management-Prinzipien
- Ressourcen
- Mitarbeiter
- Sicherheitsprozess
 - Sicherheitsrichtlinen
 - Sicherheitskonzept

→ Standards

- ISO 27000
 - Zertifizierung nach ISO/IEC 27001
 - Organisationen
 - Personen
- BSI-Standard 200 (kompatibel ISO/IEC 27001)

ISO/IEC 27000 FAMILIENÜBERSICHT

- Informationen zum ISMS sind in der ISO 27000 Familie spezifiziert
- Die ISO 27000 Familie umfasst mehrere, sich gegenseitig unterstützende, Standards
- Ein ISMS kann mittels der ISO 27000 Familie von externen Gutachtern zertifiziert werden



ISO/IEC 27002

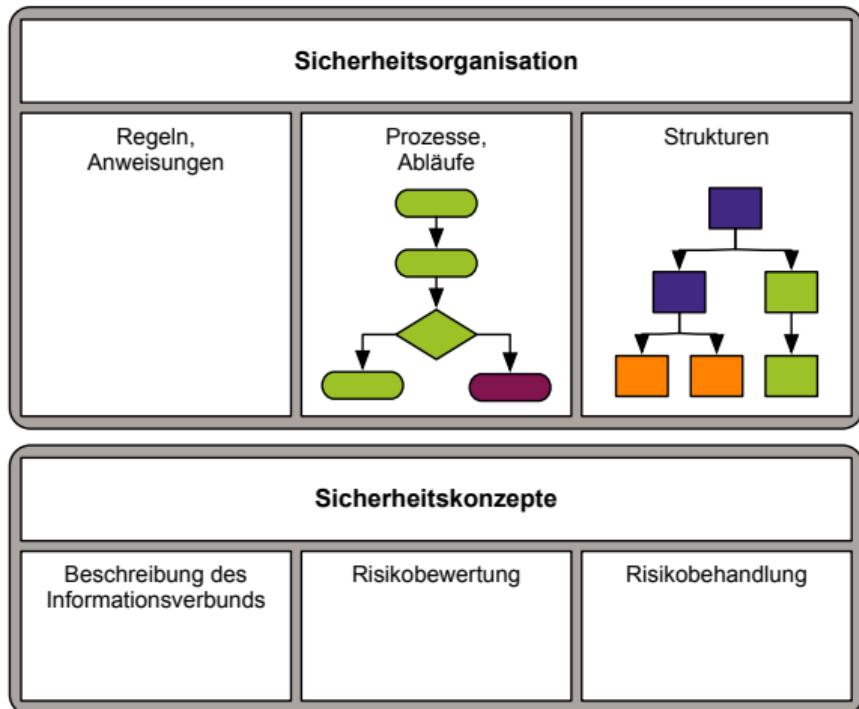
Best Practice Sammlung zur Umsetzung eines ISMS

→ Enthält Abschnitte zu:

- Weisungen und Richtlinien zur Informationssicherheit
- Organisatorische Sicherheitsmaßnahmen und Managementprozesse
- Verwaltung und Klassifizierung von Assets
- Personelle Sicherheit
- Physikalische Sicherheit und öffentliche Versorgungsdienste
- Netzwerk- und Betriebssicherheit (Daten und Telefonie)
- Zugriffskontrolle
- Systementwicklung und Wartung
- Umgang mit Sicherheitsvorfällen
- Notfallversorgung
- Einhaltung rechtlicher Vorgaben, der Sicherheitsrichtlinien und Audits

BSI STANDARD 200-X ÜBERSICHT

- **200-1:** Managementsystem für Informationssicherheit
- **200-2:** IT-Grundschutz-Methodik
- **200-3:** Risikomanagement
- **100-4:** Notfallmanagement
- **200-4:** Business Continuity Management (Community Draft)



COMMON CRITERIA

- Standard zur Bewertung der Sicherheit von IT-Produkten
 - Zertifizierung aus Eigeninitiative (z.B: Alleinstellungmerkmal)
 - Zertifizierung nötig für den Einsatz in manchen Branchen
- CC Zertifizierungen sind zweigeteilt in:
 - **Protection Profile (PP):** Beschreibung der Sicherheitsfunktionalität
 - **Evaluation Assurance Level (EAL):** Vertrauenswürdigkeit in die Umsetzung der Sicherheitsfunktionalität (EAL1 bis EAL7)
- Beispiele für EAL Stufen
 - **EAL1:** Produkt wurde gegen die Spezifikation getestet und eine Dokumentation existiert
 - **EAL3:** Es werden zusätzlich methodische Security-Tests durchgeführt
 - **EAL7:** Produkt wurde formal designed, verifiziert und getestet. Beispiel: [Diod]

COMMON CRITERIA - BEISPIEL

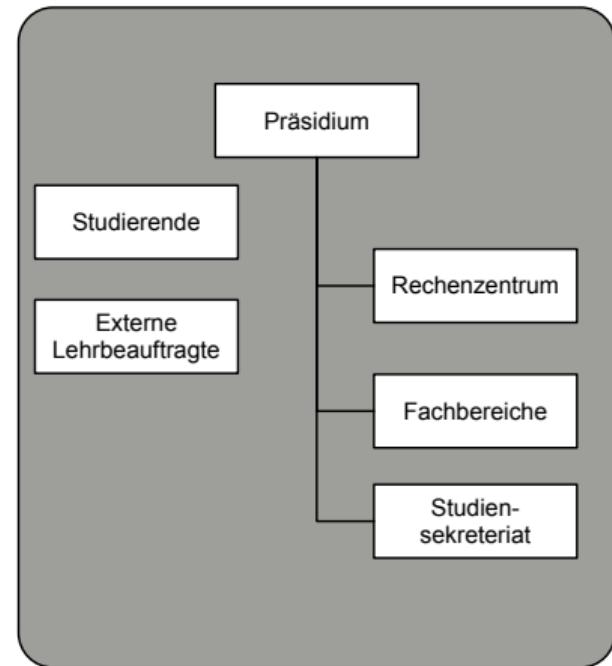
- Gesundheitsanwendungen in Deutschland werden mittels „sicherer Router“ an die Telematikinfrastruktur (TI) angebunden [KoCoBox]
- Für diese Router existiert das CC Profil [PP0098]
 - Umfasst u.a. die Funktion „Sichere Verbindung“
 - Router müssen nach EAL3 zertifiziert sein



Wert	zu schützende Eigenschaften des Wertes	Erläuterung, ⇒ davon abgeleitete Bedrohungen und Annahmen
Authentisierungsgeheimnisse bei der Speicherung und Bearbeitung im EVG	Integrität, Vertraulichkeit	<p>Die Vertraulichkeit und Integrität von Authentisierungsgeheimnissen (z. B. Passwort für Administratorauthentisierung, evtl. PIN für die gSMC-K) ist zu schützen.</p> <p>⇒ A.AK.Konnektor, A.AK.Admin_EVG, A.AK.phys_Schutz</p>

IMPLEMENTIERUNG EINES ISMS FÜR COMPASS

- Zertifizierung der IT-Sicherheit wird in Zukunft größere Rolle spielen [KoalVertrag]:
"Wir verpflichten alle staatlichen Stellen ... sich regelmäßig einer externen Überprüfung ihrer IT-Systeme zu unterziehen."
- Wir wollen ein ISMS implementieren, um für zukünftige Rechtsnormen gewappnet zu sein



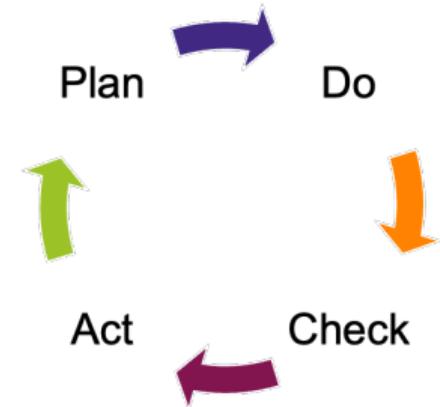
UMSETZEN EINES ISMS

1. Initial: ISMS Definieren

- 1.1 Management Support einholen und Rollen besetzen
- 1.2 Relevante Gesetze identifizieren
- 1.3 Umfang des ISMS definieren

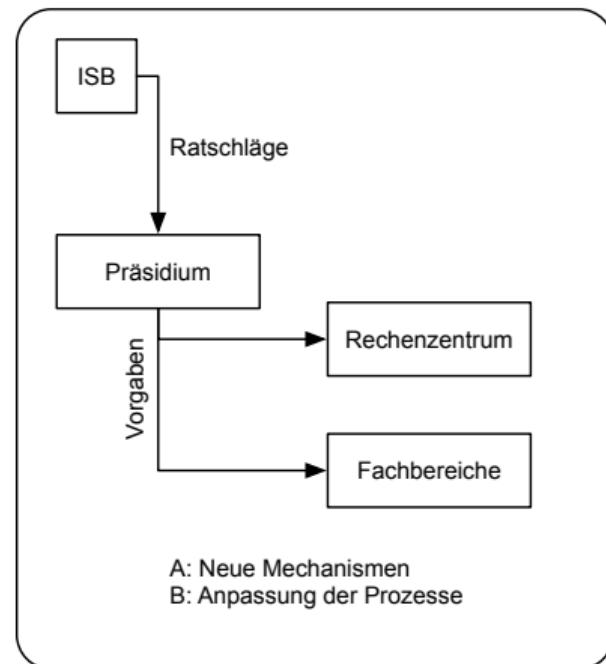
2. Wiederkehrend: ISMS Durchlaufen

- 2.1 **Plan:** Risikomanagement durchführen
- 2.2 **Do:** Maßnahmen implementieren, Ressourcen allozieren und Mitarbeitende schulen
- 2.3 **Check:** ISMS überwachen und Maßnahmen gegen definierte Kennzahlen prüfen
- 2.4 **Act:** Verbesserungen am ISMS identifizieren



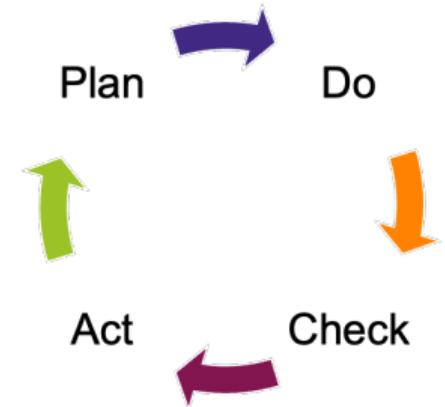
INITIAL: MANAGEMENT SUPPORT EINHOLEN UND ROLLEN BESETZEN

- Ein ISMS wird Top-Down implementiert
 - Unternehmensleitung spezifiziert grobe **Richtlinien**, um Bedrohungen zu adressieren
 - Betroffene Bereiche müssen **Prozesse** und **technische Maßnahmen** implementieren, um Konformität mit Richtlinie zu erreichen
- Relevante Rollen vergeben
 - Informationssicherheitsbeauftragter (ISB)
 - Ansprechpartner für Organisationseinheiten



INITIAL: UMFANG DES ISMS DEFINIEREN

- Der ISMS Umfang definiert **schützenswerte Kernprozesse** und **organisatorische Einheiten**, die Maßnahmen umsetzen müssen
- Umfang des ISMS sollte von der Organisationsleitung abgenommen werden



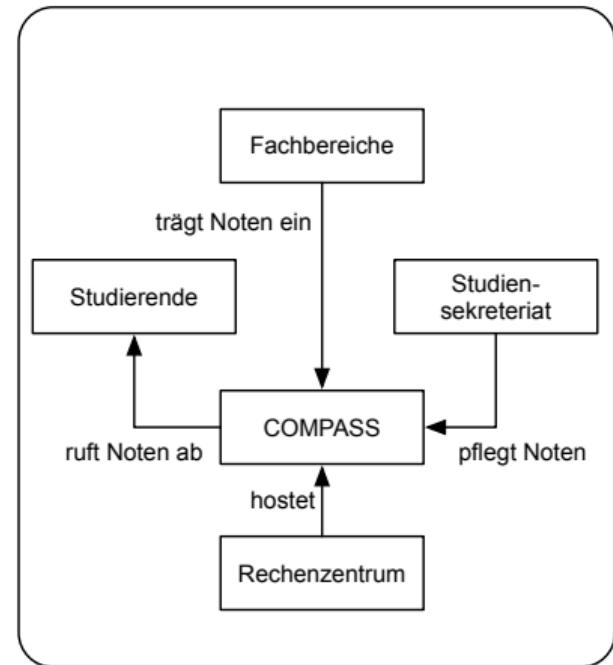
KERNPROZESSE UND UMFÄNGE FÜR COMPASS

Tauschen Sie sich mit Ihrem Sitznachbar 3 Minuten aus:

- Überlegen Sie, was die Kernprozesse von COMPASS sind und welche Schutzziele für die Kernprozesse benötigt werden.

INITIAL: UMFANG DES ISMS FÜR COMPASS DEFINIEREN

- Kernprozesse einer Hochschule:
 - Bewerbung und Zulassung
 - Studierendenmanagement
 - Lehre, Prüfungen (und Forschung)
- Beispielprozess:
 - Notenmanagement
- Bestätigung des Umfangs:
 - "Ein Ziel des ISMS der HSRM ist der Schutz der **Vertraulichkeit, Verfügbarkeit, Integrität und Authentizität** des Kernprozesses **Lehre und Prüfungen** und umfasst das Studienbüro, das Rechenzentrum und die Fachbereiche."



PLAN: RISIKOMANAGEMENT DURCHFÜHREN

→ **Zentrale Aufgabe:**

- Identifikation der konkreten Angriffe und Bedrohungen für den ISMS
- Umfang und Aufstellen einer Maßnahmenplanung

→ **Ergebnis:**

- Priorisierte Liste an technischen- und Prozessmaßnahmen zum Schutz vor Bedrohungen
- Verifikationskriterien für die Maßnahmen

→ **Beispiel:**

- Maßnahme: Sicheres Backup zum Wiederherstellen des COMPASS Notensystems
- Verifikation: Probefeldlauf Wiederherstellung von COMPASS in <= 1 Tag



DO: MASSNAHMEN IMPLEMENTIEREN

→ **Zentrale Aufgabe:**

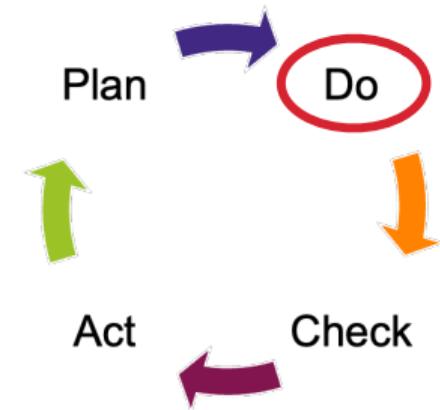
- Identifizierte Maßnahmen implementieren und Erkenntnisse zur Umsetzung gewinnen

→ **Ergebnis:**

- Technische- und Prozessmaßnahmen sind auf Basis der Vorgaben umgesetzt

→ **Beispiel:**

- Backuplösung wurde angeschafft, in Infrastruktur integriert und läuft täglich
- Prozesse zum Wiedereinspielen wurden definiert



CHECK: MASSNAHMEN ÜBERPRÜFEN

→ **Zentrale Aufgabe:**

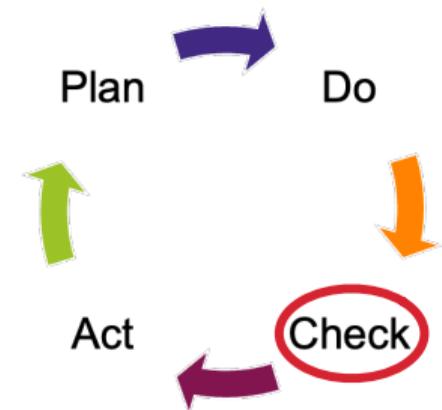
- Effektivität und Einhaltung der Maßnahmen überprüfen
- Verbesserungspotential identifizieren

→ **Ergebnis:**

- Feedback über Effektivität und Verbesserungspotential

→ **Beispiel:**

- Backup benötigt 2 Tage statt, wie geplant maximal 1 Tag
- Fehlende Einträge in COMPASS Notendatenbank beim Re-import
- Noch nicht bewertete Klausuren werden vom Backupsystem nicht gespeichert



ACT: VERBESSERUNGEN UMSETZEN

→ **Zentrale Aufgabe:**

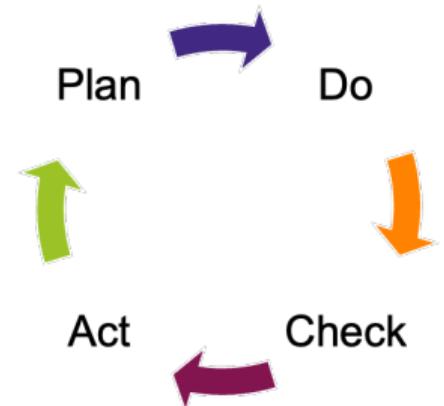
- Identifizierte Verbesserungen am ISMS Prozess umsetzen
- Änderungen kommunizieren und prüfen

→ **Ergebnis:**

- Änderungen am ISMS Umfang und Vorgehen

→ **Beispiel:**

- Es wird mehr Bandbreite zu den **Backupsystemen** benötigt
- Digitales Prüfungssystem, betrieben von Dienstleistung Lehre & Studium, muss auch vom ISMS Umfang abgedeckt werden



LANGFRISTIGES ZIEL DES ISMS

- Ziel des ISMS ist eine langfristige Absicherung durch inkrementelle Verbesserungen
 1. Passt sich mit der Zeit an die individuellen Bedürfnisse der Organisation an
 2. PDCA Zyklus sollte z.B. alle 1-2 Jahre durchgeführt werden
 3. Bestehende Maßnahmen sollten regelmäßig überprüft werden
- Alle Schritte des ISMS sollten ausreichend dokumentiert werden
- Die Qualität eines ISMS kann mittels Reifegradmodellen gemessen werden



Quelle: RGM- letzter Besuch 26.03.2023

ZUSAMMENFASSUNG

- Gesetze und Standards im Bereich der IT-Sicherheit
- Zusammenspiel zwischen Gesetzen und Standards
- Relevante Rollen sowie den Inhalt des Umfangs im ISMS
- Hintergrund des PDCA Zyklus im ISMS



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Risikomanagement

May 5, 2023

Marc Stöttinger



More people are killed every year by pigs than by sharks, which shows you how good we are at evaluating risk.

Bruce Schneier

MOTIVATION RISKOMANAGEMENT

- **Bisher:** Aufsetzen eines ISMS und Durchführen des PDCA Zyklus zum strukturierten Behandeln von IT-Sicherheit

- **Beispiel:** Kernprozess "Notenpflege in COMPASS"
 - Plan: Identifikation und Priorisierung der IT-Sicherheitsmaßnahmen
 - Do: Implementierung des Backup Systems
 - Check: Verifikation der Backup Lösung
 - Act: Implementierung der ISMS Verbesserungen



MOTIVATION RISKOMANAGEMENT

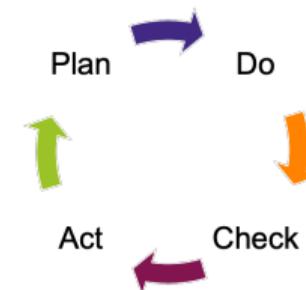
- **Bisher:** Aufsetzen eines ISMS und Durchführen des PDCA Zyklus zum strukturierten Behandeln von IT-Sicherheit
- **Beispiel:** Kernprozess "Notenpflege in COMPASS"
 - Plan: Identifikation und Priorisierung der IT-Sicherheitsmaßnahmen
 - Do: Implementierung des Backup Systems
 - Check: Verifikation der Backup Lösung
 - Act: Implementierung der ISMS Verbesserungen



Prio	Maßnahme
?	Backupsystem
?	Schulung
?	Multi-Faktor Authentifizierung

MOTIVATION RISKOMANAGEMENT

- **Bisher:** Aufsetzen eines ISMS und Durchführen des PDCA Zyklus zum strukturierten Behandeln von IT-Sicherheit
- **Beispiel:** Kernprozess "Notenpflege in COMPASS"
 - Plan: Identifikation und Priorisierung der IT-Sicherheitsmaßnahmen
 - Do: Implementierung des Backup Systems
 - Check: Verifikation der Backup Lösung
 - Act: Implementierung der ISMS Verbesserungen
- **Heute:**
 - Inhalt: Identifikation und Priorisierung von Maßnahmen im Plan Schritt



Prio	Maßnahme
?	Backupsystem
?	Schulung
?	Multi-Faktor Authentifizierung

WIRTSCHAFTLICHKEIT VON IT-SICHERHEIT

- Einsatz von IT-Systemen soll den Profit erhöhen, indem z.B.
 - Geschäftsprozesse optimiert und Kosten reduziert werden
 - der Umsatz gesteigert wird

WIRTSCHAFTLICHKEIT VON IT-SICHERHEIT

- Einsatz von IT-Systemen soll den Profit erhöhen, indem z.B.
 - Geschäftsprozesse optimiert und Kosten reduziert werden
 - der Umsatz gesteigert wird
- IT-Sicherheitsmaßnahmen reduzieren typischerweise weder Kosten noch steigern sie den Umsatz

WIRTSCHAFTLICHKEIT VON IT-SICHERHEIT

- Einsatz von IT-Systemen soll den Profit erhöhen, indem z.B.
 - Geschäftsprozesse optimiert und Kosten reduziert werden
 - der Umsatz gesteigert wird
- IT-Sicherheitsmaßnahmen reduzieren typischerweise weder Kosten noch steigern sie den Umsatz
- IT-Sicherheit verhindert Schäden, die mit gewisser Eintrittswahrscheinlichkeit anfallen

WIRTSCHAFTLICHKEIT VON IT-SICHERHEIT

- Einsatz von IT-Systemen soll den Profit erhöhen, indem z.B.
 - Geschäftsprozesse optimiert und Kosten reduziert werden
 - der Umsatz gesteigert wird
- IT-Sicherheitsmaßnahmen reduzieren typischerweise weder Kosten noch steigern sie den Umsatz
- IT-Sicherheit verhindert Schäden, die mit gewisser Eintrittswahrscheinlichkeit anfallen
- Benötigt wird also eine Wirtschaftlichkeitsbetrachtung von IT-Sicherheit, die potentielle Schäden ihrer Eintrittswahrscheinlichkeit gegenüberstellt

RISIKOMANAGEMENT AM BEISPIEL VERSICHERUNGEN

Verschiedene Versicherungsmodelle für Fahrzeuge

→ Haftpflicht



Quelle: [https://www.check24.de/kfz-
versicherung/automarken/bmw/1er/](https://www.check24.de/kfz-versicherung/automarken/bmw/1er/)

RISIKOMANAGEMENT AM BEISPIEL VERSICHERUNGEN

Verschiedene Versicherungsmodelle für Fahrzeuge

- Haftpflicht
- Teilkaskoversicherung



Quelle: <https://www.check24.de/kfz-versicherung/automarken/bmw/1er/>

RISIKOMANAGEMENT AM BEISPIEL VERSICHERUNGEN

Verschiedene Versicherungsmodelle für Fahrzeuge

- Haftpflicht
- Teilkaskoversicherung
- Vollkaskoversicherung



Quelle: <https://www.check24.de/kfz-versicherung/automarken/bmw/1er/>

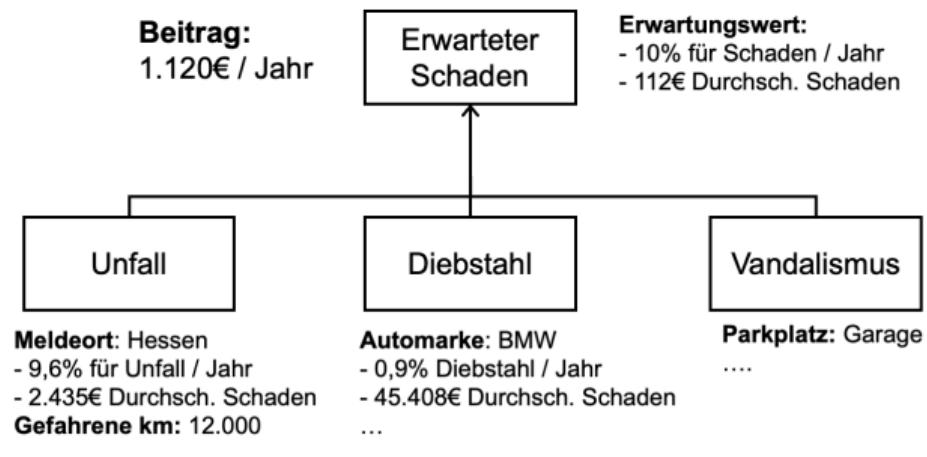
RISIKOMANAGEMENT AM BEISPIEL VERSICHERUNGEN

Verschiedene Versicherungsmodelle für Fahrzeuge

- Haftpflicht
- Teilkaskoversicherung
- Vollkaskoversicherung

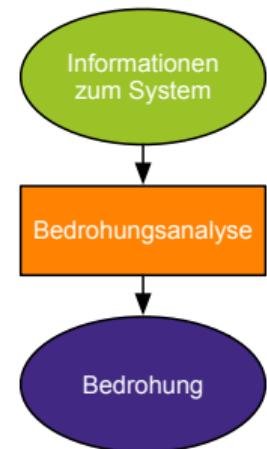


Quelle: <https://www.check24.de/kfz-versicherung/automarken/bmw/1er/>



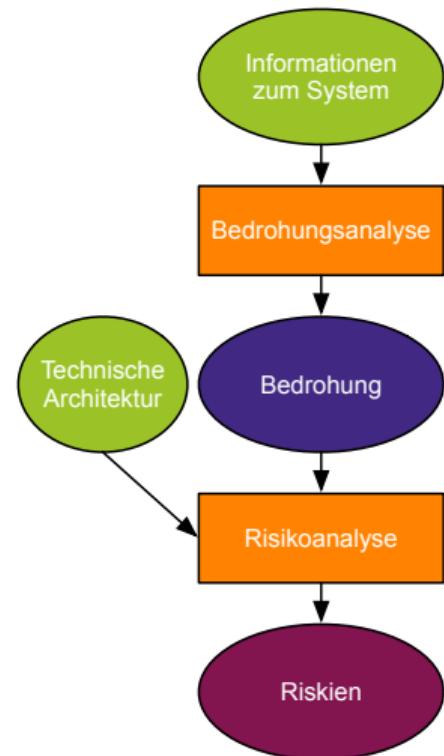
BEDROHUNGS- UND RISIKOANALYSE (TARA)

- Eine **Bedrohung** ist ein Umstand, der zu einem Schaden führen könnte
 - Beispiel: Passwort raten
- Eine **Gefährdung** ist eine Bedrohung, die konkret eine Schwachstelle ausnutzt
 - Beispiel: Angreifer rät schwaches Passwort



BEDROHUNGS- UND RISIKOANALYSE (TARA)

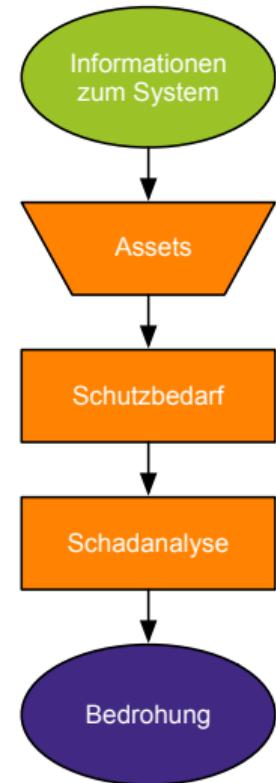
- Eine **Bedrohung** ist ein Umstand, der zu einem Schaden führen könnte
 - Beispiel: Passwort raten
- Eine **Gefährdung** ist eine Bedrohung, die konkret eine Schwachstelle ausnutzt
 - Beispiel: Angreifer rät schwaches Passwort
- Ein **Risiko** ist die Kombination aus dem Ausmaß des Schadens einer Gefährdung und deren Eintrittswahrscheinlichkeit
- Die **Bedrohungs- und Risikoanalyse** (auch **Threat Analysis** and **Risk Assessment**, TARA) ist ein strukturierter Vorgang, um Risiken zu identifizieren und priorisieren



BEDROHUNGSDANALYSE

- Eine **Bedrohungsanalyse** als ein strukturierter Prozess, um potentielle Bedrohungen möglichst vollständig zu identifizieren

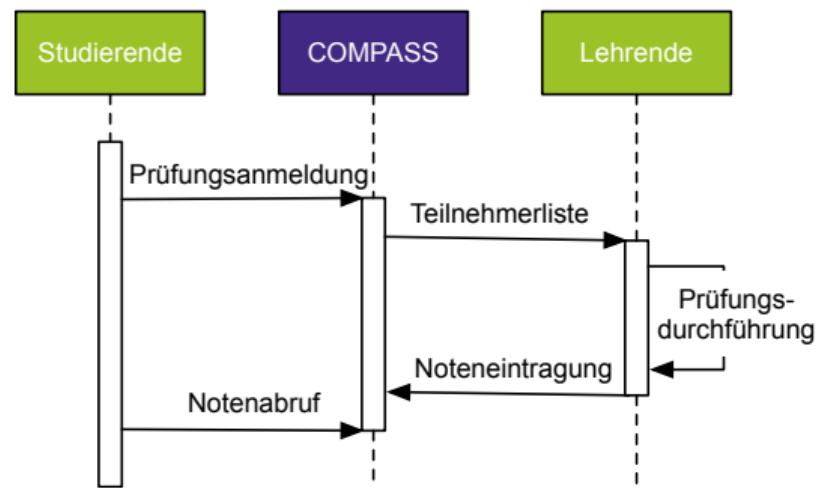
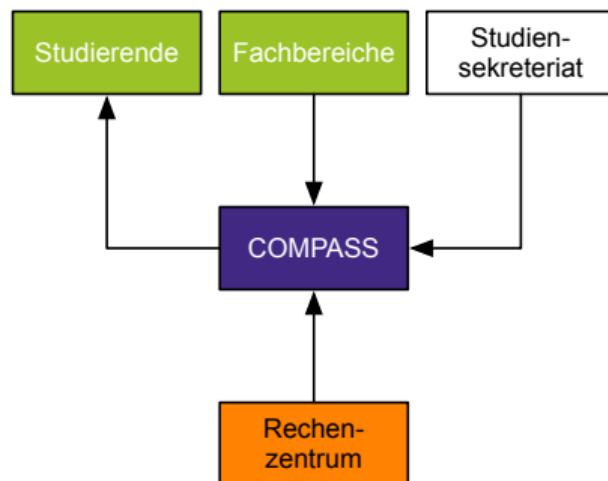
- Vorgehen bei der Bedrohungsanalyse:
 - Welche Vermögenswerte (**Assets**) sind am System beteiligt
 - Welchen **Schutzbedarf** haben die Assets?
 - Wie hoch ist der **Schaden** bei Verlust des Schutzbedarfs?
 - Was sind abstrakte **Bedrohungen**?



TECHNISCHE VORRAUSETZUNG - KERNPROZESS LEHRE UND PRÜFUNG

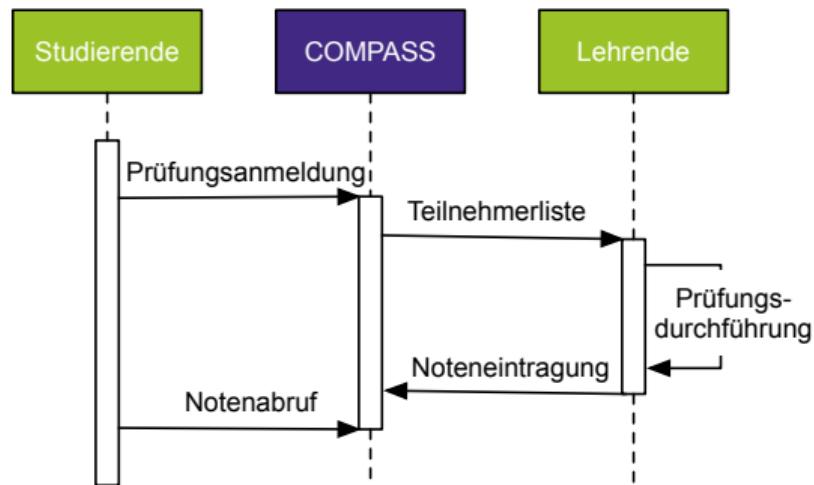
→ Kernprozess Lehre und Prüfungen mit Use-Cases

- Studierende melden sich zu einer Prüfung an
- Lehrende tragen Noten ein
- Studierende rufen Noten ab



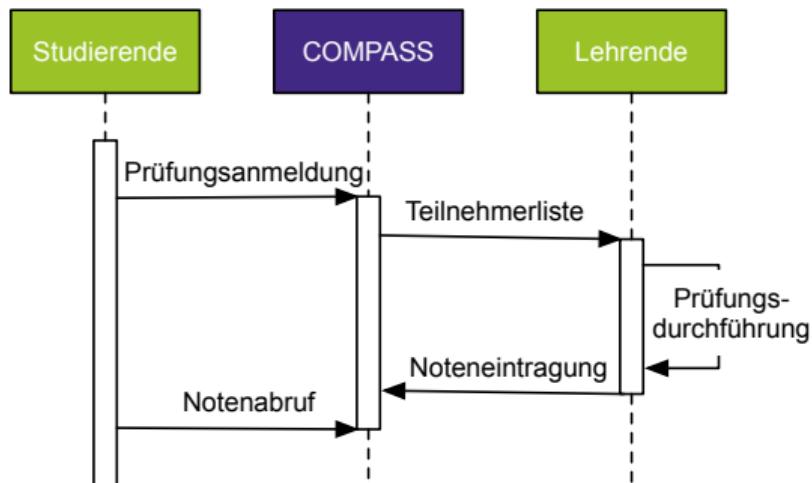
ASSEST IDENTIFIZIEREN

- **Assets:** Etwas von (ideellem) Wert für die Teilnehmenden
- Use-Cases für Lehre und Prüfung
 - Prüfungsanmeldung
 - Notenmanagement



ASSEST IDENTIFIZIEREN

- **Assets:** Etwas von (ideellem) Wert für die Teilnehmenden
- Use-Cases für Lehre und Prüfung
 - Prüfungsanmeldung
 - Notenmanagement
- Assets für Lehre und Prüfung
 - Noten (Daten)
 - Prüfungsanmeldungen (Daten)
 - COMPASS Funktionen (System)



SCHUTZBEDARFS- UND SCHADENSANALYSE

- Der Schutzbedarf liefert eine Unterteilung für mögliche Schäden an Assets
 - Unterteilung anhand ausgewählter Sicherheitsziele (z.B. CIA, CIAA, STRIDE, ...)
- Für jede Kombination aus (Sicherheitsziel x Asset) sollte der mögliche Schaden mittels Schadensnormen (z.B. HEAVENS Standard) abgeschätzt werden

[HEAVENS] Bewertung	Beschreibung
Keine	Keine Verluste
Niedrige	Geringe Verluste
Mittel	Tolerierbare Verluste
Hoch	Substantielle Verluste
Kritisch	Verluste bedrohen Existenz

SCHUTZBEDARFS- UND SCHADENSANALYSE

- Der Schutzbedarf liefert eine Unterteilung für mögliche Schäden an Assets
 - Unterteilung anhand ausgewählter Sicherheitsziele (z.B. CIA, CIAA, STRIDE, ...)
- Für jede Kombination aus (Sicherheitsziel x Asset) sollte der mögliche Schaden mittels Schadensnormen (z.B. HEAVENS Standard) abgeschätzt werden
- Beispiel: Verfügbarkeit der Noten
 - Studierenden könnten das Studium nicht abschließen
 - Imageschaden (Rückgang der Studierendenzahlen)
 - Schaden = Mittel

[HEAVENS] Bewertung	Beschreibung
Keine	Keine Verluste
Niedrige	Geringe Verluste
Mittel	Tolerierbare Verluste
Hoch	Substantielle Verluste
Kritisch	Verluste bedrohen Existenz

BEDROHUNGSANALYSE COMPASS ERGEBNIS FÜR ASSET NOTEN

Asset	Sicherheitsziel	Bedrohung	Schaden	Begründung
Noten	Vertraulichkeit	Veröffentlichung der Noten	Hoch	DSGVO Strafzahlungen, Imageschaden
	Integrität	Verfälschung der Noten	Mittel	Klagen durch Studierende, Studierende könnten Studium nicht abschließen, Imageschaden
	Authentizität	Note zu Modul nicht korrekt zugeordnet	Niedrig	Verfälschung des Notendurchschnitts für Studierende
	Verfügbarkeit	Noten nicht mehr verfügbar	Mittel	Studierende können Studium nicht abschließen, Imageschaden
	Autorisierung	Unbefugter Zugriff auf Noten	Hoch	(Verweis auf Vertraulichkeit und Integrität)
	Nicht-Abstreitbarkeit	Leugnung einer Prüfungsfunktion	Niedrig	Verbesserung der Note, Studierende nicht exmatrikulierbar

STRIDE: METHODIK

Nutzt die Schutzzielspezifischen Bedrohung und ordnet abstrakt dies einem Basiselement einer Datenfluss-Architektur zu. Die sogenannte **Stride-per-Element** Methode. Somit können Systeme abstrakt modelliert und analysiert werden.

STRIDE Typ	Datentransfer	Speicher	Prozess	Actor
S poofing			x	x
T ampering	x	x	x	
R eputation	x		x	x
I nformation Disclosure	x	x	x	
D enial of Service	x	x	x	
E valuation of Privlages	x	x	x	

STRIDE: BEISPIEL

Entering Grades

STRIDE CIA LINDDUN

```
graph TD; Lecturer[Lecturer] -- "grades" --> Write((Write grades)); Write -- "grade entered" --> Grade[Grades of lecture Security];
```

Edit Threat

Title
Lecturer gets spoofed by student

STRIDE threat type
Spoofing

Threat status
NA Open Mitigated

Priority
High Medium Low

Description
Student logs in as lecturer to change grades

Mitigations
Mitigation or prevention for the threat

Save Cancel

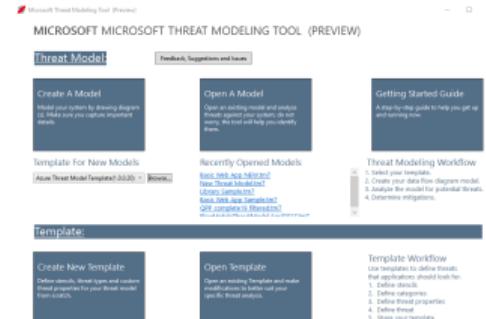
STRIDE: WERKZEUG

Das **Threat Modeling Tool** und **Threat-Dragon Tools** können zur Erstellung von Bedrohungsmodellen für die Bedrohungsanalyse genutzt werden.

- Modellierung des Systems per Datenflussgraph
- Bedrohungsanalyse mittels Stride-per-Element Ansatz
- Tools sind lizenzenfrei verfügbar:
 - Threat Modeling Tool (nur für Windows)
[Download]
 - Threat-Dragon (für Windows, Linux, Mac)
[Download]



Quelle: <https://owasp.org/www-project-threat-dragon/>



Quelle: <https://learn.microsoft.com/de-de/azure/security/develop/threat-modeling-tool-getting-started>

DISKUSSION IN KLEINEN GRUPPEN

Tauschen Sie sich mit Ihrem Nachbarn 5 Minuten aus:

- Welchen Schutzbedarf und Schadenshöhe sehen Sie für die Assets
 - Prüfungsanmeldungen (Daten)
 - COMPASS Funktion (System)

[HEAVENS] Bewertung	Beschreibung
Keine	Keine Verluste
Niedrige	Geringe Verluste
Mittel	Tolerierbare Verluste
Hoch	Substantielle Verluste
Kritisch	Verluste bedrohen Existenz

BEDROHUNGSMODELL - DREAD

Das DREAD Modell wird genutzt, um potentielle Bedrohungen besser zu Kategorisierung und zu bewerten. Dabei wird die Bedrohungen nach folgenden Kriterien bewertet:

→ **D**amage (Schaden):

Wie schwer ist der versuchte Schaden durch den Angriff?

BEDROHUNGSMODELL - DREAD

Das DREAD Modell wird genutzt, um potentielle Bedrohungen besser zu Kategorisierung und zu bewerten. Dabei wird die Bedrohungen nach folgenden Kriterien bewertet:

→ **D**amage (Schaden):

Wie schwer ist der versuchte Schaden durch den Angriff?

→ **R**eproducibility (Reproduzierbarkeit):

Wie leicht lässt sich der Angriff reproduzieren/anwenden/wiederholen?

BEDROHUNGSMODELL - DREAD

Das DREAD Modell wird genutzt, um potentielle Bedrohungen besser zu Kategorisierung und zu bewerten. Dabei wird die Bedrohungen nach folgenden Kriterien bewertet:

→ **D**amage (Schaden):

Wie schwer ist der versuchte Schaden durch den Angriff?

→ **R**eproducibility (Reproduzierbarkeit):

Wie leicht lässt sich der Angriff reproduzieren/anwenden/wiederholen?

→ **E**xloitability (Ausnutzbarkeit):

Wie schwer ist es, den Angriff durchzuführen?

BEDROHUNGSMODELL - DREAD

Das DREAD Modell wird genutzt, um potentielle Bedrohungen besser zu Kategorisierung und zu bewerten. Dabei wird die Bedrohungen nach folgenden Kriterien bewertet:

→ **D**amage (Schaden):

Wie schwer ist der versuchte Schaden durch den Angriff?

→ **R**eproducibility (Reproduzierbarkeit):

Wie leicht lässt sich der Angriff reproduzieren/anwenden/wiederholen?

→ **E**xloitability (Ausnutzbarkeit):

Wie schwer ist es, den Angriff durchzuführen?

→ **A**ffected users (Betroffene):

Wie viele Personen/Systeme/Komponenten sind vom Angriff betroffen?

BEDROHUNGSMODELL - DREAD

Das DREAD Modell wird genutzt, um potentielle Bedrohungen besser zu Kategorisierung und zu bewerten. Dabei wird die Bedrohungen nach folgenden Kriterien bewertet:

→ **D**amage (Schaden):

Wie schwer ist der versuchte Schaden durch den Angriff?

→ **R**eproducibility (Reproduzierbarkeit):

Wie leicht lässt sich der Angriff reproduzieren/anwenden/wiederholen?

→ **E**xloitability (Ausnutzbarkeit):

Wie schwer ist es, den Angriff durchzuführen?

→ **A**ffected users (Betroffene):

Wie viele Personen/Systeme/Komponenten sind vom Angriff betroffen?

→ **D**iscoverability (Auffindbarkeit):

Wie einfach kann die Angriffsprozedur gefunden werden?

BEWERTUNGSBEISPIEL - DREAD

Bewertung	Gering	Mittel	Hoch
D amage: Schaden	Verarbeitung unbedeutender Information ist möglich	Verbreitung relevanter Informationen ist möglich	Sicherheitslücke untergraben und vollständige Bescheinigungen erlangt
R eproducibility: Reproduzierbarkeit	Nur mit Kenntnis der Sicherheitslücke schwer reproduzierbar	Angriff kann innerhalb eines bestimmten Zeitfensters reproduziert werden	Angriff kann jederzeit reproduziert werden.
E xloitability: Ausnutzbarkeit	Nur Experten mit Fachwissen können den Angriff durchführen	Erfahrene Programmierer können den Angriff ausführen	Programmieranfänger kann den Angriff in kurzer Zeit durchführen.
A ffected users: Betroffene	Ein sehr geringer Prozentsatz von Benutzern ist betroffen	Einzelne sind betroffen; keine Standardkonfiguration	Alle Benutzer sind betroffen; Standardkonfiguration
D iscoverability: Auffindbarkeit	Der Fehler ist unbekannt und es ist unwahrscheinlich, dass Benutzer das Schadenspotential erkennen.	Die Sicherheitslücke befindet sich in einem selten verwendeten Teil des Produkts. Die bösartige Verwendbarkeit ist nur mit einem Aufwand erkennbar.	Angriff wird über öffentlich zugängliche Medien erklärt. Die Sicherheitslücke findet sich in einer viel verwendeten Funktion und ist leicht wahrnehmbar.

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN (1/2)

Ermittlung der Eintrittswahrscheinlichkeit am Beispiel der Versicherungen und von COMPASS

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN (1/2)

Ermittlung der Eintrittswahrscheinlichkeit am Beispiel der Versicherungen und von COMPASS

→ **Beispiel Versicherungen**

Unfall

- Rückschluss aus empirischen Datenmengen (Unfälle / Jahr)
- Unfallsituationen vergleichbar
- Unfall wird (meist) nicht durch Menschen beabsichtigt

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN (1/2)

Ermittlung der Eintrittswahrscheinlichkeit am Beispiel der Versicherungen und von COMPASS

→ **Beispiel Versicherungen**

Unfall

- Rückschluss aus empirischen Datenmengen (Unfälle / Jahr)
- Unfallsituationen vergleichbar
- Unfall wird (meist) nicht durch Menschen beabsichtigt

→ **IT-Sicherheit**

Verfügbarkeit der Noten

- Datenmengen recht klein bis sogar individuell
- Situation sehr individuell (wer hat Zugriff, wo wird gehostet, ...)
- Angreifer beabsichtigt Schaden

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN (2/2)

Alternative: Modellierung der Eintrittswahrscheinlichkeit durch Voraussetzungen für einen erfolgreichen Angriff (z.B. aus [HEAVENS])

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN (2/2)

Alternative: Modellierung der Eintrittswahrscheinlichkeit durch Voraussetzungen für einen erfolgreichen Angriff (z.B. aus [HEAVENS])

[HEAVENS] Faktoren	Kritisch(3)	Hoch(2)	Mittel(1)	Niedrig(0)
Zugriffsmöglichkeiten	Internet	Lokales Netzwerk	Systemzugriff	Physischer Zugriff
Expertise	Laie	Kompetent	Experte	Mehrere Experten
Wissen über das Ziel	Öffentlich	Branchenspezifisch	Unternehmensspezifisch	Geheim
Benötigte Geräte	Standard	Spezialisierte Geräte	Speziell Produzierte Geräte	Mehrere Speziell Produzierte Geräte

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN – BEISPIELE

- **Beispiel A:** Phishing eines Passwortes
 - Angreifer identifiziert Opfer via HSRM Homepage
 - Angreifer baut Login-Seite nach und sendet sie an Opfer

Faktor	Phising
Zugriffs-möglichkeiten	Internet (3)
Expertise	Laie (3)
Wissen über das Ziel	Öffentlich (3)
Benötigte Geräte	Standard (3)
Summe	12

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN – BEISPIELE

- **Beispiel A:** Phishing eines Passwortes
 - Angreifer identifiziert Opfer via HSRM Homepage
 - Angreifer baut Login-Seite nach und sendet sie an Opfer
- **Beispiel B:** USB-Stick mit Trojaner an COMPASS Server anschließen
 - Angreifer verschafft sich Zugang zum Serverraum
 - Angreifer schließt bösartigen USB Stick an, der einen Trojaner installiert

Faktor	Phising
Zugriffs-möglichkeiten	Internet (3)
Expertise	Laie (3)
Wissen über das Ziel	Öffentlich (3)
Benötigte Geräte	Standard (3)
Summe	12

EINTRITTSWAHRSCHEINLICHKEITEN VON ANGRIFFEN – BEISPIELE

- **Beispiel A:** Phishing eines Passwortes
 - Angreifer identifiziert Opfer via HSRM Homepage
 - Angreifer baut Login-Seite nach und sendet sie an Opfer
- **Beispiel B:** USB-Stick mit Trojaner an COMPASS Server anschließen
 - Angreifer verschafft sich Zugang zum Serverraum
 - Angreifer schließt bösartigen USB Stick an, der einen Trojaner installiert

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

BEISPIEL NOTENPFLÈGE VIA COMPASS

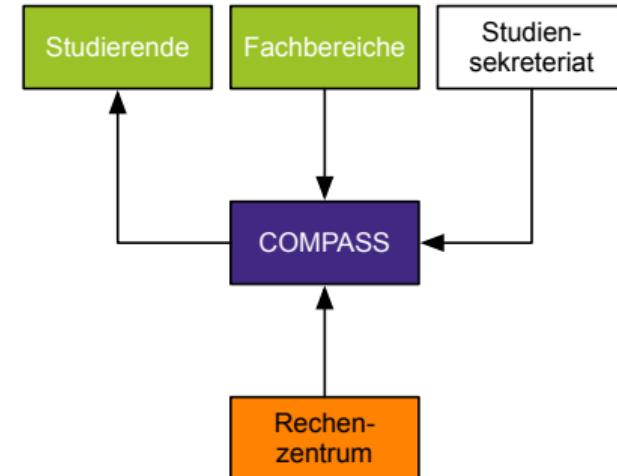
- Die Risikoanalyse setzt Kenntnisse technischer Details voraus
 - Umso mehr Details bekannt sind, desto besser die Abschätzung
 - Technisches Verständnis entwickelt sich über die ISMS Zyklen

BEISPIEL NOTENPFLUGE VIA COMPASS

- Die Risikoanalyse setzt Kenntnisse technischer Details voraus
 - Umso mehr Details bekannt sind, desto besser die Abschätzung
 - Technisches Verständnis entwickelt sich über die ISMS Zyklen

→ **Technische Details zu COMPASS**

- Die COMPASS Webseite authentifiziert alle User nur via Passwort
- Auf dem COMPASS Server existiert ein ssh Zugang für Administratoren
- Der Serverraum ist mit einem Gebäudeschlüssel zugänglich



ANGRIFFSBÄUME ZUR ABSCHÄTZUNG DER EINTRITTSWAHRSCHEINLICHKEIT

- Unterschiedlicher Detailgrad der Informationen in der Risikoanalyse
 - Abstrakte Bedrohung: Verlust der Verfügbarkeit der Noten
 - Konkrete Angriffe: Phishing oder Anschluss bösartiger Hardware

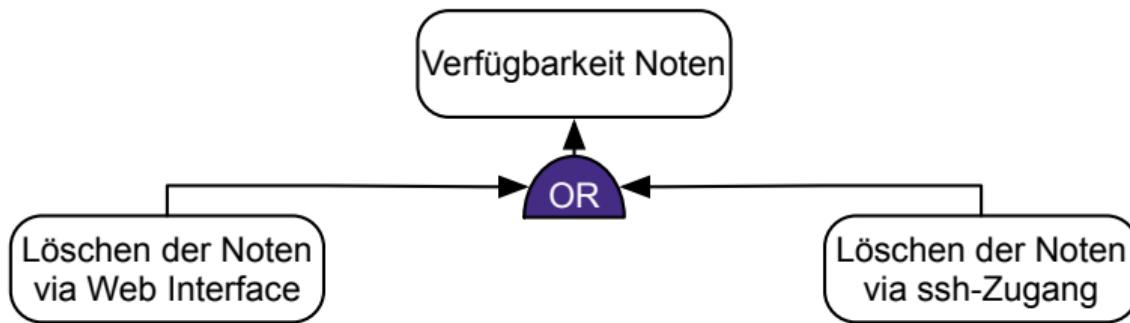
ANGRIFFSBÄUME ZUR ABSCHÄTZUNG DER EINTRITTSWAHRSCHEINLICHKEIT

- Unterschiedlicher Detailgrad der Informationen in der Risikoanalyse
 - Abstrakte Bedrohung: Verlust der Verfügbarkeit der Noten
 - Konkrete Angriffe: Phishing oder Anschluss bösartiger Hardware
- Methode zum Verknüpfen der Informationen: **Angriffsbäume**
 - Abstrakte Bedrohungen als Wurzelknoten
 - Konkrete und abschätzbare Angriffe als Blätter
 - Knoten als logische Unterteilung möglicher Angriffe
 - Knoten werden mittels logischer Verknüpfungen (AND oder OR) verbunden

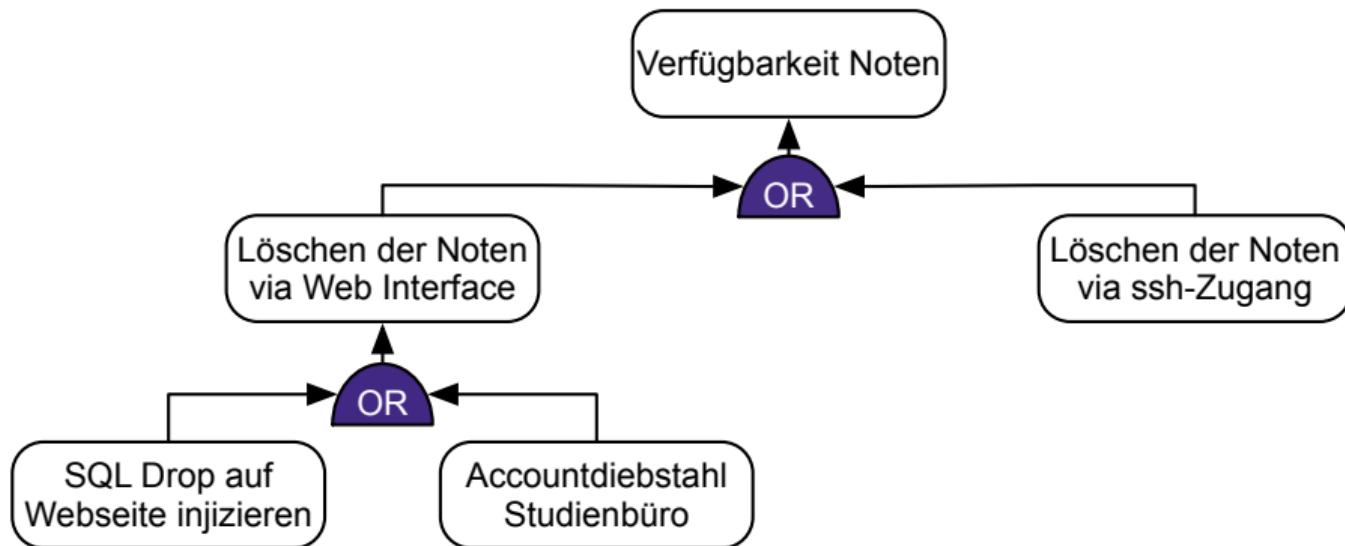
VERFEINERUNG DER BEDROHUNGEN VIA EINES ANGRIFFSBAUMS

Verfügbarkeit Noten

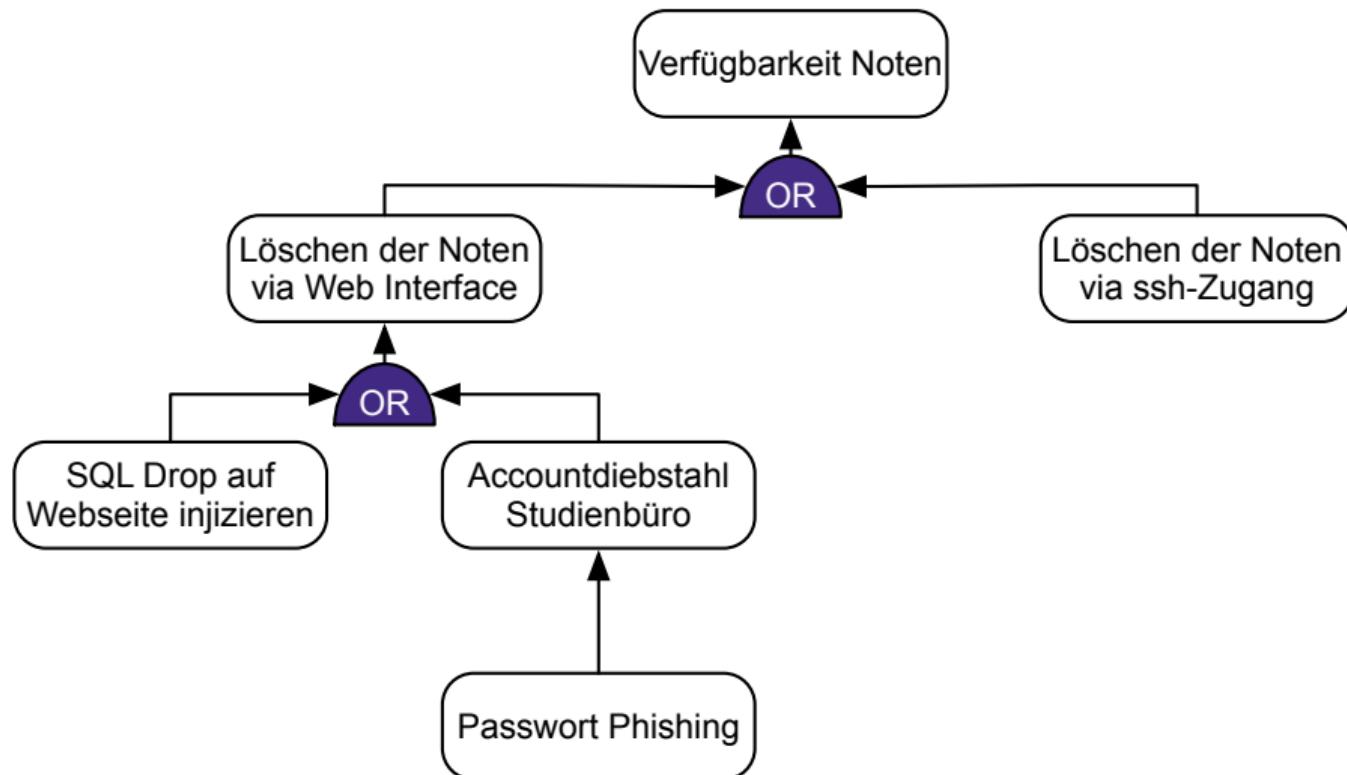
VERFEINERUNG DER BEDROHUNGEN VIA EINES ANGRIFFSBAUMS



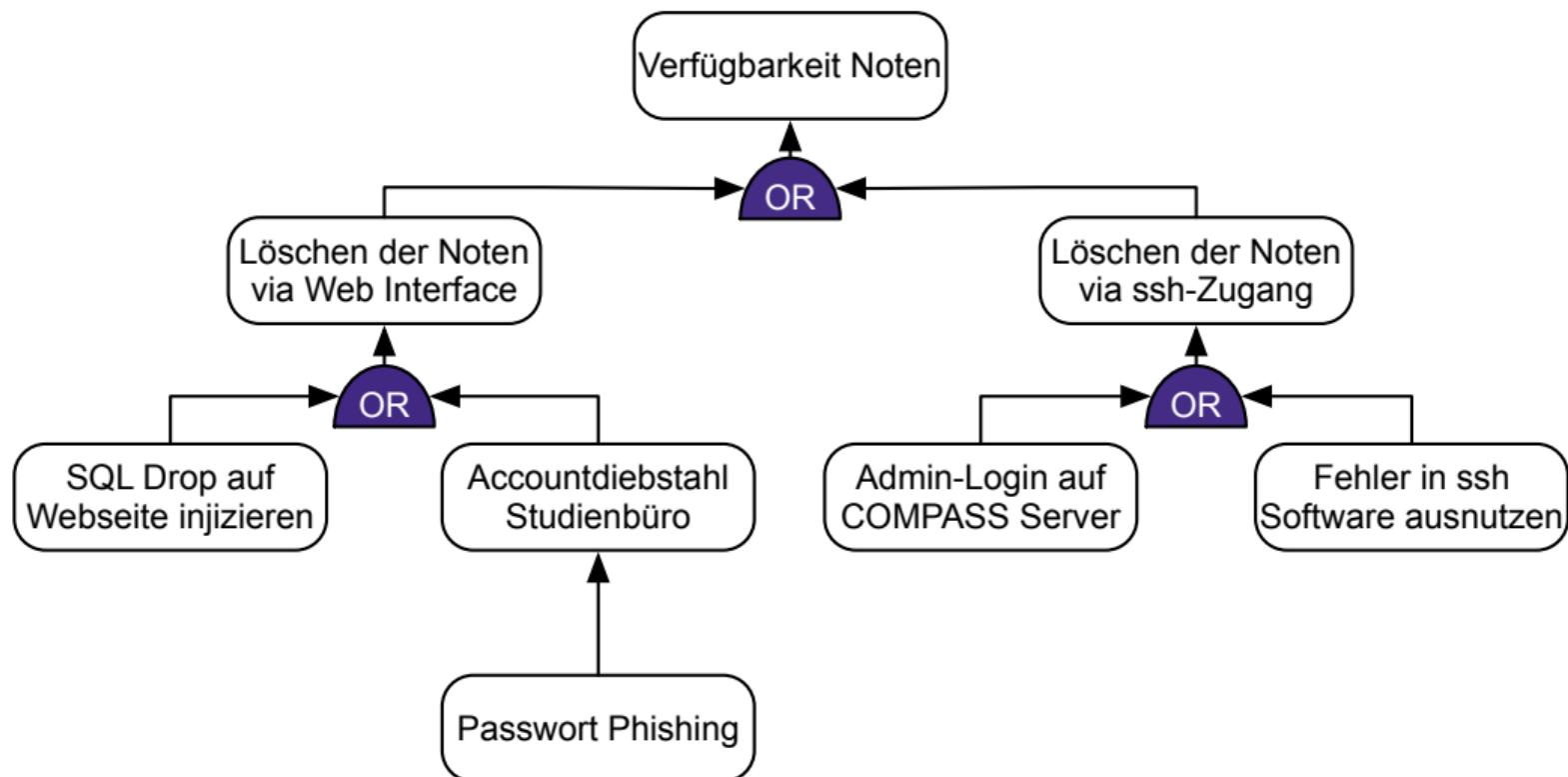
VERFEINERUNG DER BEDROHUNGEN VIA EINES ANGRIFFSBAUMS



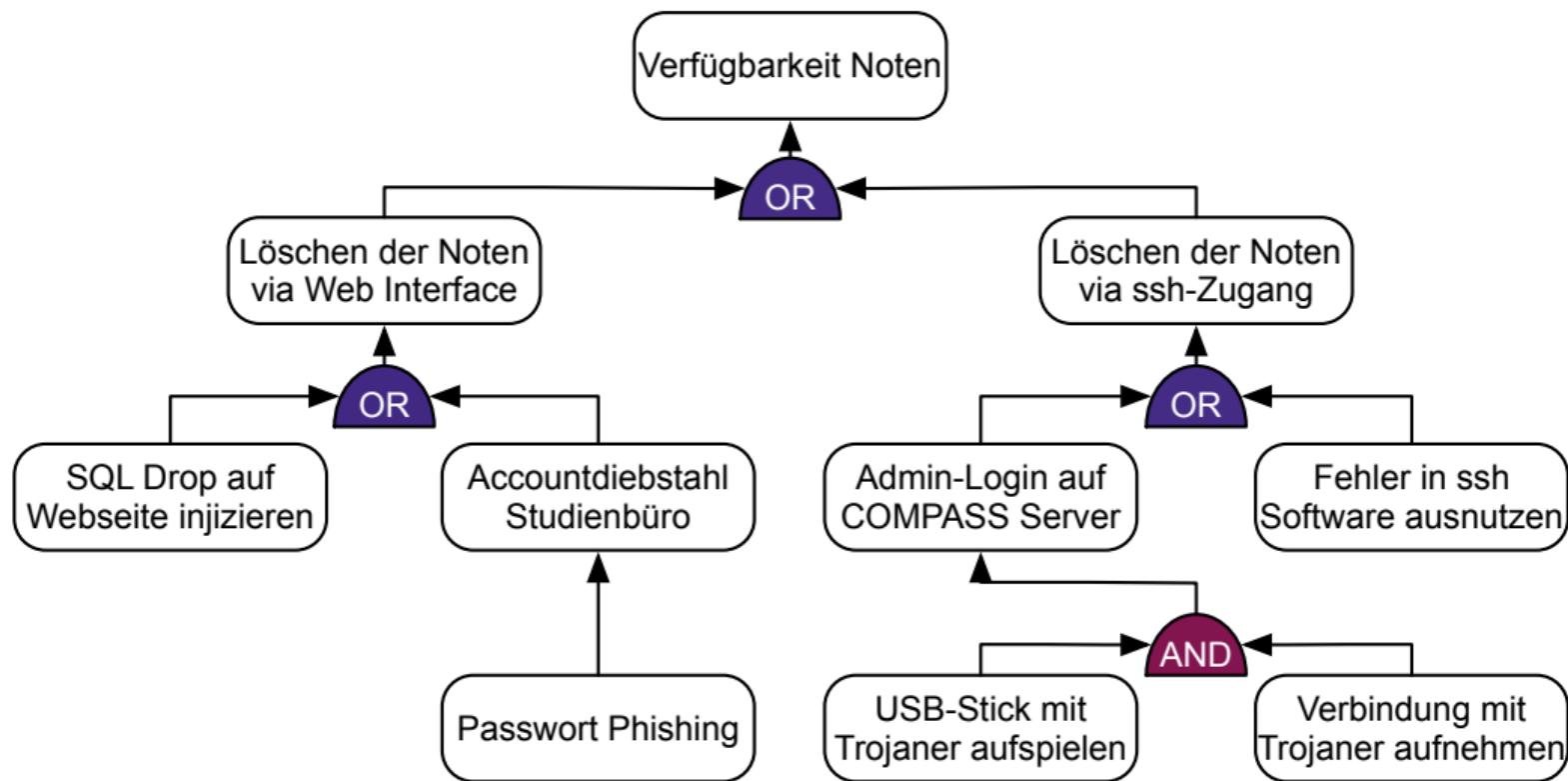
VERFEINERUNG DER BEDROHUNGEN VIA EINES ANGRIFFSBAUMS



VERFEINERUNG DER BEDROHUNGEN VIA EINES ANGRIFFSBAUMS



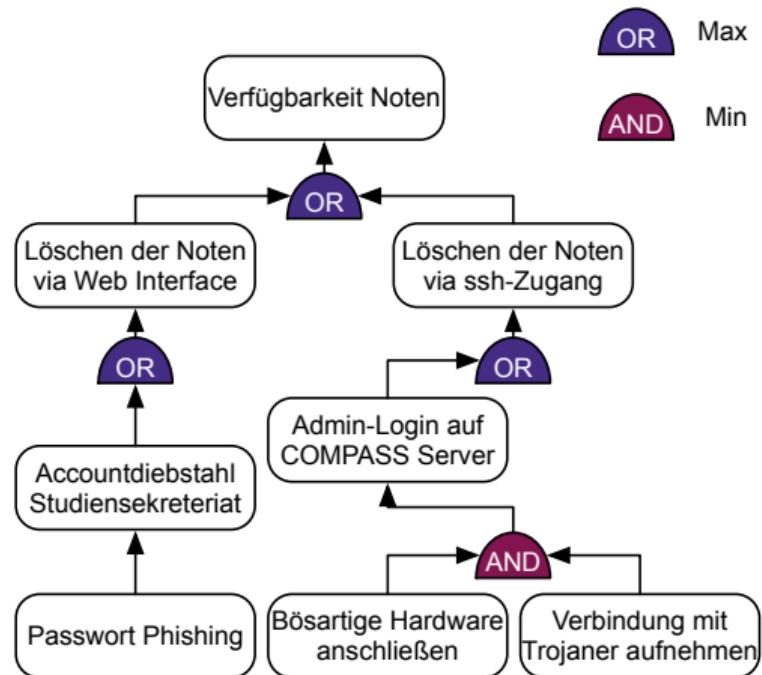
VERFEINERUNG DER BEDROHUNGEN VIA EINES ANGRIFFSBAUMS



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

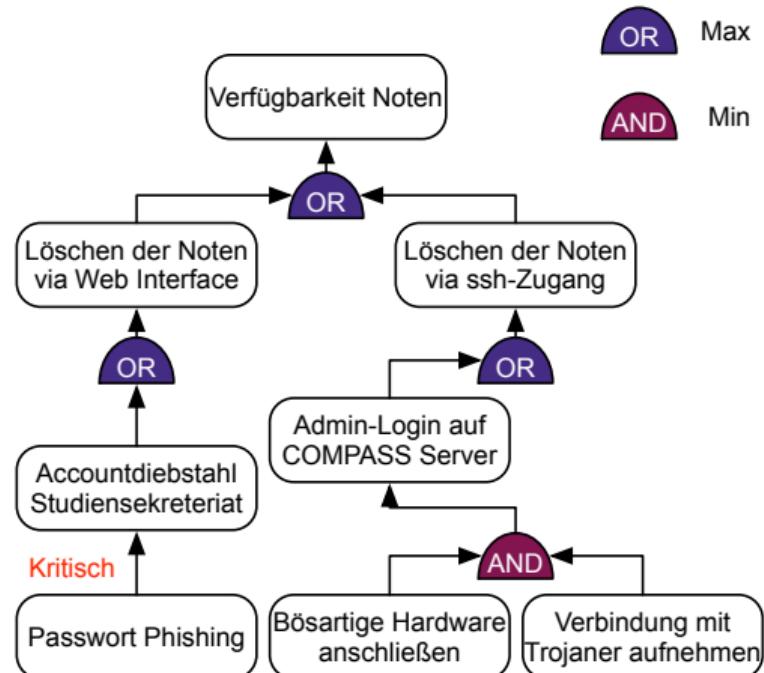
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

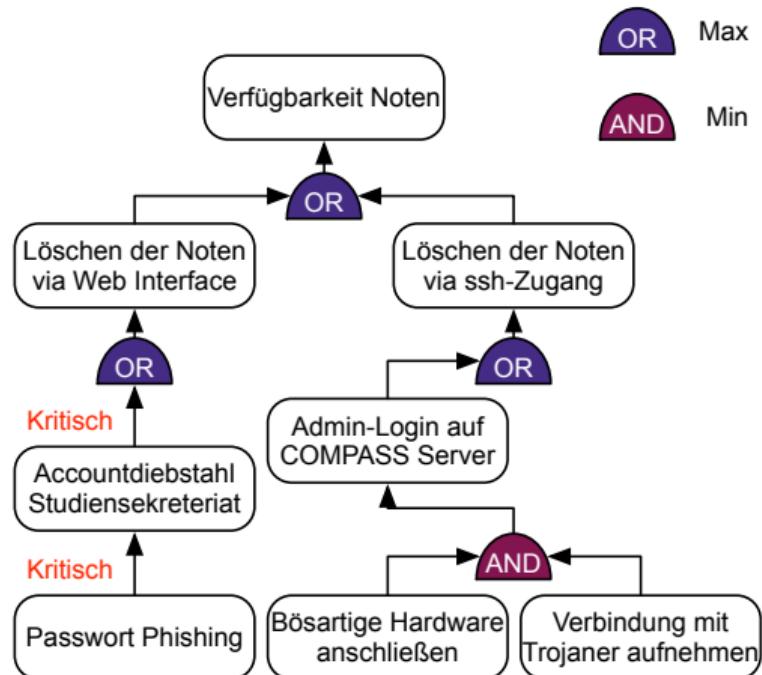
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

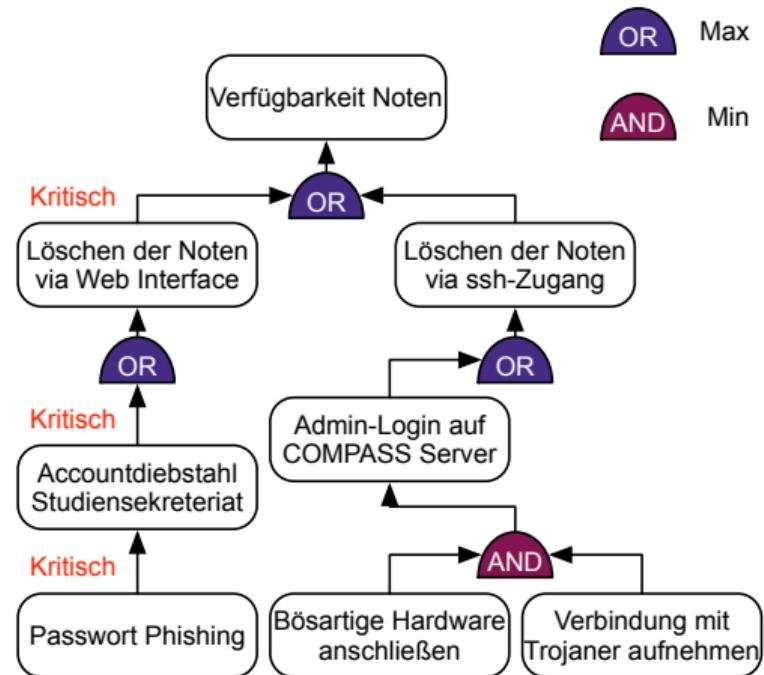
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

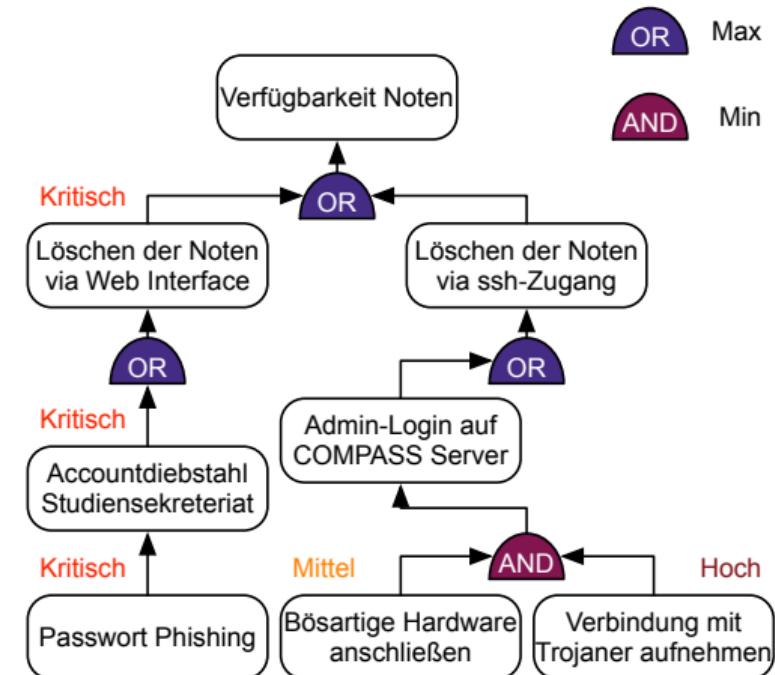
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

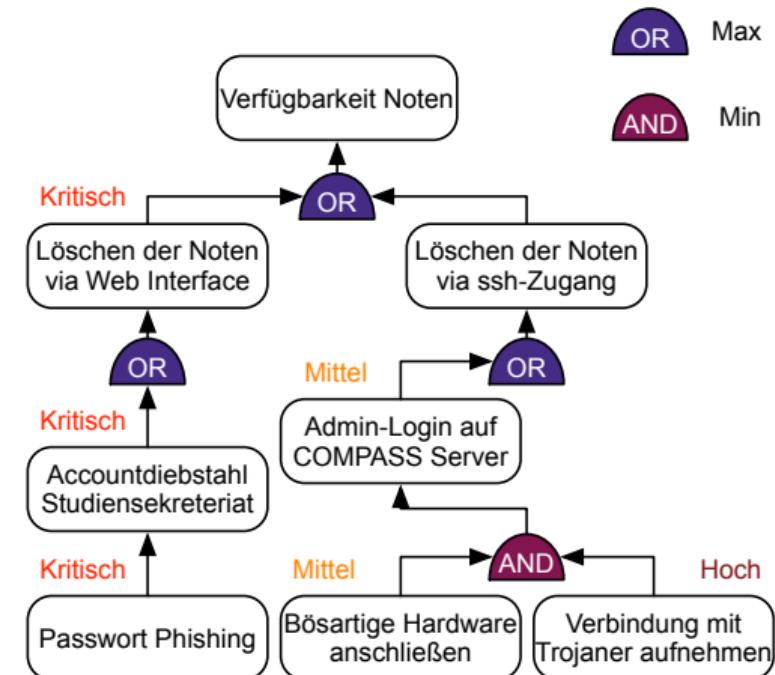
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

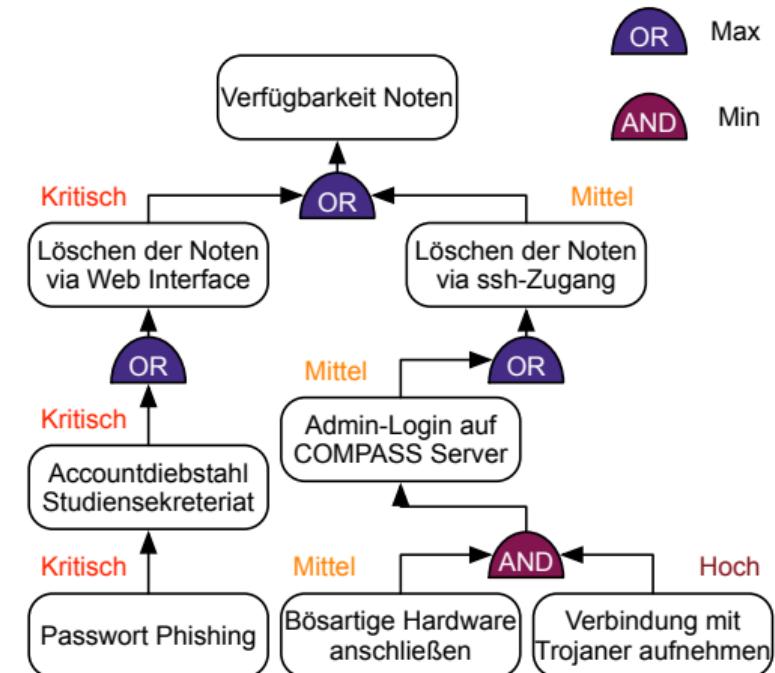
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

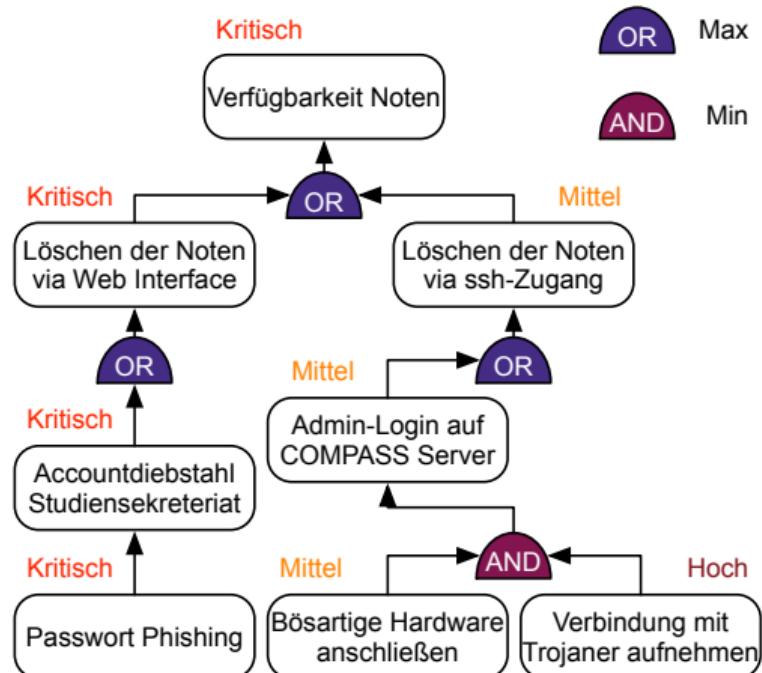
Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



PROPAGIEREN DER EINTRITTSWAHRSCHEINLICHKEIT

Faktor	Phising	USB-Stick mit Trojaner
Zugriffs-möglichkeiten	Internet (3)	Physisch (0)
Expertise	Laie (3)	Kompetent (2)
Wissen über das Ziel	Öffentlich (3)	Unternehmensspez. (1)
Benötigte Geräte	Standard (3)	Spezialisiert (2)
Summe	12	5

Wert	0-2	3-5	6-8	9-10	11-12
Bewertung	Keine	Niedrig	Mittel	Hoch	Kritisch



BERECHNUNG DES GESAMTRISIKOS

- Schaden und Eintrittswahrscheinlichkeit werden zum Risiko kombiniert
- **Risiko:** Verluste der Verfügbarkeit der Noten ist **Hoch**
 - Schaden: Mittel
 - Eintrittswahrscheinlichkeit: Kritisch

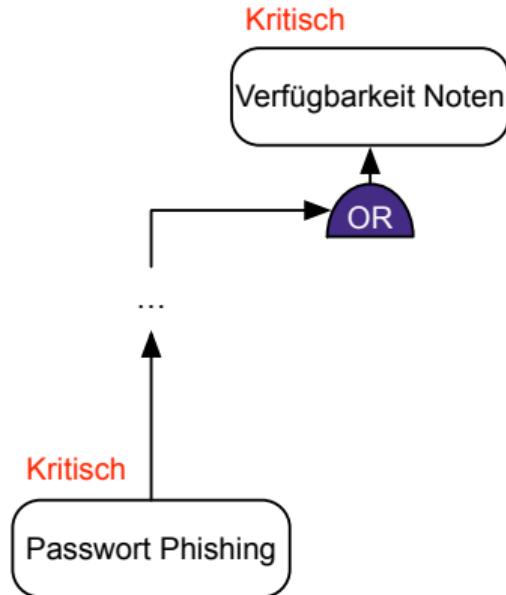
		Risiko				
		Klein	Niedrig	Mittel	Hoch	Kritisch
Eintrittswahrscheinlichkeit		Klein	Niedrig	Mittel	Hoch	Kritisch
Schaden	Klein	Klein	Klein	Klein	Klein	Niedrig
	Niedrig	Klein	Niedrig	Niedrig	Niedrig	Mittel
	Mittel	Klein	Niedrig	Mittel	Mittel	Hoch
	Hoch	Klein	Niedrig	Mittel	Hoch	Hoch
	Kritisch	Niedrig	Mittel	Hoch	Hoch	Kritisch

ZIEL DER RISIKOBEHANDLUNG

Die Bedrohungs- und Risikoanalyse liefert eine Liste an Risiken zusammen mit detaillierten Angriffsbäumen

- **Fokus der Risikobehandlung:** Die am höchsten priorisierten Risiken sinnvoll adressieren

Bedrohung	Bewertung
Noten nicht abrufbar	Hoch
Veröffentlichung der Noten	Hoch
Unbefugter Zugriff auf Noten	Hoch
Verfälschung der Noten	Mittel



MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Es gibt verschiedene Möglichkeiten mit einem Risiko umzugehen

MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Es gibt verschiedene Möglichkeiten mit einem Risiko umzugehen
- 1. **Mitigieren:** Mechanismen implementieren, um die Eintrittswahrscheinlichkeit und somit das Risiko zu reduzieren (z.B.: Zwei-Faktor Authentifizierung)

MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Es gibt verschiedene Möglichkeiten mit einem Risiko umzugehen
 - 1. **Mitigieren:** Mechanismen implementieren, um die Eintrittswahrscheinlichkeit und somit das Risiko zu reduzieren (z.B.: Zwei-Faktor Authentifizierung)
 - 2. **Vermeiden:** Daten oder Zugang entfernen, um die Bedrohung zu entfernen (z.B. Admin Zugang zum COMPASS Server nicht aus dem Internet erreichbar)

MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

→ Es gibt verschiedene Möglichkeiten mit einem Risiko umzugehen

1. **Mitigieren:** Mechanismen implementieren, um die Eintrittswahrscheinlichkeit und somit das Risiko zu reduzieren (z.B.: Zwei-Faktor Authentifizierung)
2. **Vermeiden:** Daten oder Zugang entfernen, um die Bedrohung zu entfernen (z.B. Admin Zugang zum COMPASS Server nicht aus dem Internet erreichbar)
3. **Transferieren:** Schaden durch Verträge abdecken (z.B. Versicherung für Ransomware oder Weitergabe des Risikos an Zulieferer)

MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

→ Es gibt verschiedene Möglichkeiten mit einem Risiko umzugehen

1. **Mitigieren:** Mechanismen implementieren, um die Eintrittswahrscheinlichkeit und somit das Risiko zu reduzieren (z.B.: Zwei-Faktor Authentifizierung)
2. **Vermeiden:** Daten oder Zugang entfernen, um die Bedrohung zu entfernen (z.B. Admin Zugang zum COMPASS Server nicht aus dem Internet erreichbar)
3. **Transferieren:** Schaden durch Verträge abdecken (z.B. Versicherung für Ransomware oder Weitergabe des Risikos an Zulieferer)
4. **Akzeptieren:** Keine Aktionen durchführen (z.B. sinnvoll bei niedrigem Risiko)

MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

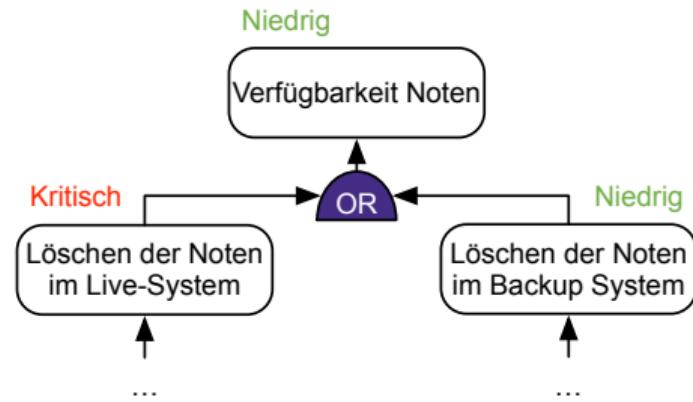
- Es gibt verschiedene Möglichkeiten mit einem Risiko umzugehen
 - 1. **Mitigieren:** Mechanismen implementieren, um die Eintrittswahrscheinlichkeit und somit das Risiko zu reduzieren (z.B.: Zwei-Faktor Authentifizierung)
 - 2. **Vermeiden:** Daten oder Zugang entfernen, um die Bedrohung zu entfernen (z.B. Admin Zugang zum COMPASS Server nicht aus dem Internet erreichbar)
 - 3. **Transferieren:** Schaden durch Verträge abdecken (z.B. Versicherung für Ransomware oder Weitergabe des Risikos an Zulieferer)
 - 4. **Akzeptieren:** Keine Aktionen durchführen (z.B. sinnvoll bei niedrigem Risiko)
- Entscheidung über Umgang mit Risiko muss von Person mit entsprechender Befugnis getroffen werden

MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Verschiedene Möglichkeiten im Beispiel
Verfügbarkeit der Noten

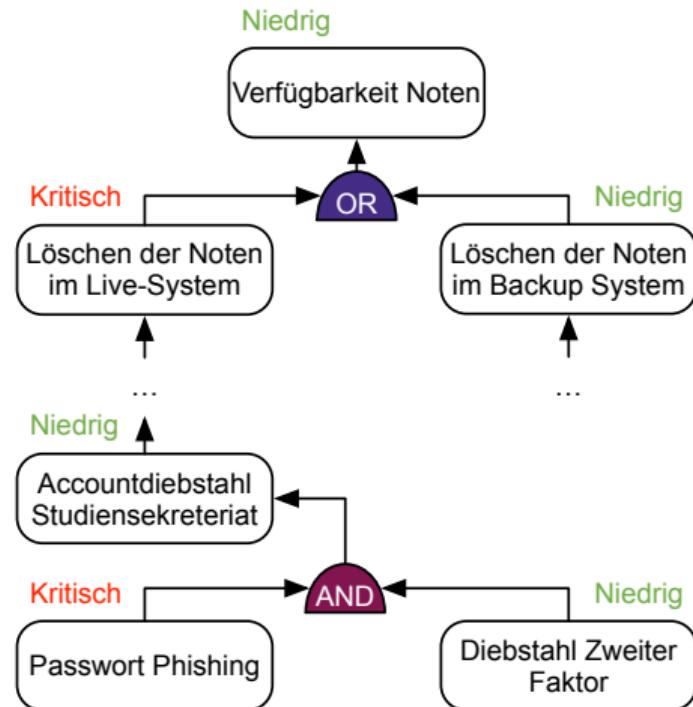
MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Verschiedene Möglichkeiten im Beispiel Verfügbarkeit der Noten
 - Implementierung eines Backup Systems



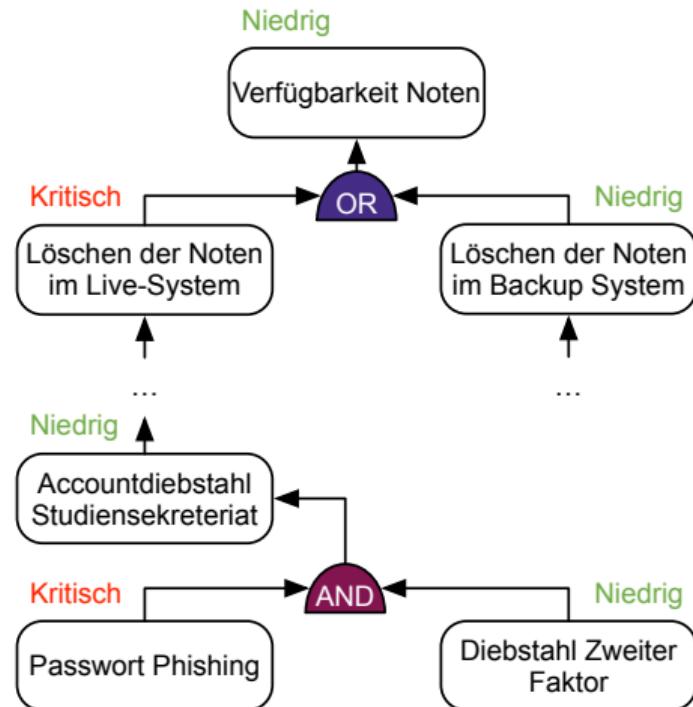
MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Verschiedene Möglichkeiten im Beispiel Verfügbarkeit der Noten
 - Implementierung eines Backup Systems
 - Implementierung einer 2-Faktor Authentifizierung (z.B. Smartphone)



MÖGLICHKEITEN ZUR RISIKOBEHANDLUNG

- Verschiedene Möglichkeiten im Beispiel Verfügbarkeit der Noten
 - Implementierung eines Backup Systems
 - Implementierung einer 2-Faktor Authentifizierung (z.B. Smartphone)
- Entscheidung zu Maßnahmen hängt u.a. von Umsetzbarkeit, Effektivität und Kosten ab



ZUSAMMENFASSUNG

- Probleme bei der Wirtschaftlichkeitsbetrachtung von IT-Sicherheit
- Bedrohungsanalyse mit STRIDE und DREAD
- Bedrohungs- und Risikoanalyse an einem konkreten Beispiel durchführen
- Verfeinerung der Risikoanalyse durch Angriffsbäume
- Möglichkeiten der Risikobehandlung



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Einführung Kryptographie

May 5, 2023

Marc Stöttinger



Don't roll your own Crypto!

Anonymous

WAS VERSTEHEN WIR UNTER KRYPTOLOGIE?

Kryptologie

Wissenschaft der Verfahren zur Geheimhaltung von Nachrichten, aber auch zu deren Berechnung. Kryptologie vereinigt Kryptographie und Kryptanalyse.

→ **Kryptographie:**

- Geheimschriftkunde – offen versendete Nachrichten sollen durch Verschlüsselung bzw. Chiffrierung für Unbefugte nicht lesbar sein.

→ **Kryptanalyse:**

- Meist mathematische und statistische Methoden zur Entzifferung von Geheimtexten, d.h. Informationen unbefugt erlangen.

WOZU BRAUCHEN WIR KRYPTOLOGIE?

- Kryptologie ist als mathematische Disziplin wissenschaftlich fundiert und anerkannt.
- Mathematik liefert – jedenfalls im Prinzip – Rechtfertigung für die „Stärke“ einer Sicherheitsmaßnahme.
- Im Idealfall lässt sich beweisen, dass ein kryptographischer Algorithmus ein gewisses Sicherheitsniveau hat (oder eben nicht).
- Damit kann der Nachweis erbracht werden, dass für eine bestimmte Anwendung der beanspruchte Sicherheitswert tatsächlich erreicht wird.

Achtung! Nachweis für benötigten Sicherheitswert

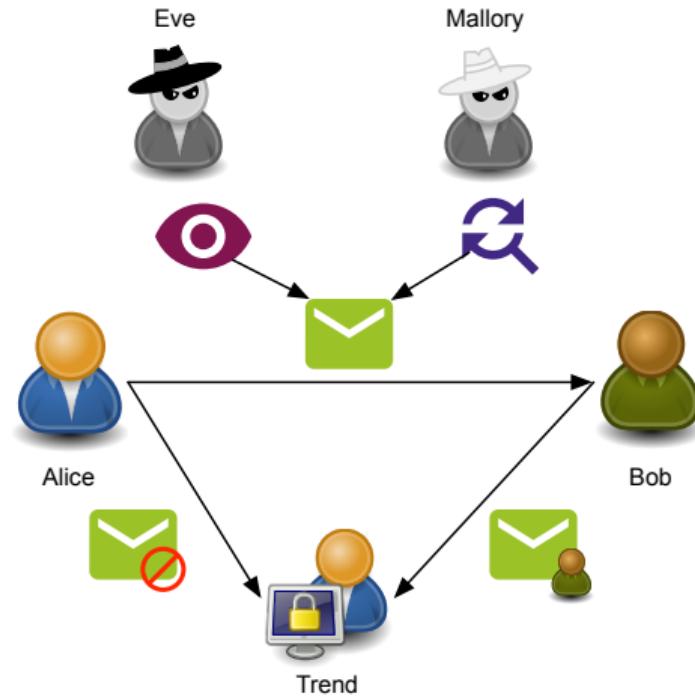
Design und Entwurf von Kryptographischen Algorithmen und Protokollen, benötigte Jahre Erfahrung in Zahlentheorie und Statistik, sowie der Implementierung!

WARUM MACHEN WIR DANN KRYPTOGRAPHIE HIER?!

- Verstehen funktionaler Anforderungen an die Sicherheit von Verschlüsselungsverfahren
- Anhand der Berechnungskomplexität sichere von unsicheren Verfahren unterscheiden können
- Den Verwendungszweck von Betriebsmodi für Blockchiffren verstehen
- Passende Betriebsmodi für einen einfachen Anwendungsfall auswählen können
- Den Unterschied zwischen öffentlichem und privatem Schlüssel verstehen
- In Grundzügen die mathematische Probleme kennen, auf denen asymmetrische Verfahren beruhen
- Die Vor- und Nachteile von symmetrischen- und asymmetrischen Verfahren verstehen
- Für einen gegebenen Kontext bestimmen können, ob ein symmetrisches, asymmetrisches oder hybrides Verschlüsselungsverfahren am geeignetsten ist

KRYPTOGRAPHIE SOAP OPERA

- **Alice** will Nachricht an **Bob** senden
- **Eve** (Eavesdropper) will Nachricht unbefugt lesen
- **Mallory** (Malicious) will Nachricht unbefugt verändern oder sich als Alice ausgeben
- **Trent** (Trusted Entity) ist eine vertrauenswürdige dritte Instanz, die Meinungsverschiedenheiten zwischen Alice und Bob klärt (z.B. ein Gericht)



ANGRIFFSPOTENTIAL NACHRICHTENÜBERTRAGUNG

Sicherheitsziel	Beschreibung	Werkzeug	Kryptographie
Vertraulichkeit	Eve und Mallory sollen die Nachricht nicht lesen können	Verschlüsselung	X
	Bob soll nicht wissen, von wem die Nachricht kommt	Anonymisierung	
	Eve und Mallory sollen die Kommunikation nicht sehen	Steganographie	
Integrität	Änderungen der Nachricht von Mallory sollen erkannt werden	Hashfunktionen, Messages Authentication Codes, Digitale Signaturen	X
Authentizität	Bob will sichergehen, dass die Nachricht von Alice stammt	Message Authentication Codes, Digitale Signaturen	X
Verfügbarkeit	Die Nachricht muss bei Bob ankommen	Redundanz, Content Distribution	
Autorisierung	Andere Nutzende von Alice's oder Bob's Computer dürfen die Nachricht nicht senden oder sehen	Access Control	
Verbindlichkeit	Alice kann die Nachricht im Nachhinein nicht leugnen	Digitale Signaturen	X

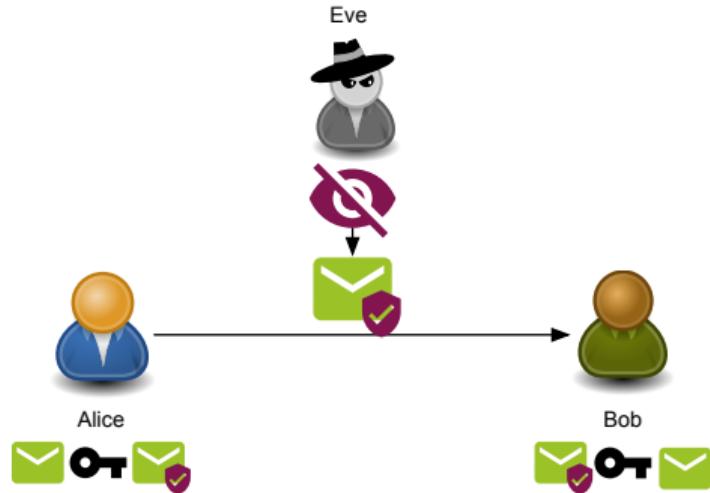
VERTRAULICHKEIT DURCH VERSCHLÜSSELUNG

→ **Bedrohung:**

Eve liest die Nachricht mit

→ **Ziel:**

Personen ohne den entsprechenden Schlüssel können keine Informationen aus verschlüsselter Nachricht gewinnen



DEFINITIONEN

Symbol	Bezeichnung	Erklärung
P	Plaintext/Klartext	Nachricht im Klartext
C	Ciphertext/Chiffretext	Verschlüsselte Nachricht
K_E	Verschlüsselungsschlüssel	Schlüssel der zum Verschlüsseln der Nachricht verwendet wird.
K_D	Entschlüsselungsschlüssel	Schlüssel der zum Entschlüsseln der Nachricht verwendet wird. Muss basierend auf K_E berechnet werden ($K_D = f(K_E)$).
$C = Enc_K(P)$	Verschlüsselungsfunktion	Verschlüsselt den Plaintext P zum Ciphertext C unter Verwendung des Schlüssels K .
$C = Dec_K(P)$	Entschlüsselungsfunktion	Entschlüsselt den Ciphertext C zum Plaintext P unter Verwendung des Schlüssels K . Es gilt: $P = Dec_K(Enc_K(P))$.

MONOALPHABETISCHE SUBSTITUTION - CAESAR CHIFFRE

- Ersetze jeden Buchstaben mit dem Buchstaben K Positionen weiter hinten im Alphabet:

Plaintext	A	B	C	D	...	Z
Ciphertext	E	F	G	H	...	D

- Einfache Vorschrift:
$$C_i = P_i + K \pmod{n}$$
- Für binäre Daten kann eine XOR-Operation statt der Addition genutzt werden.
- Behält allerdings die statistische Verteilung der Buchstaben bei!

Plaintext

CRYPTOGRAPHY, or writing in cipher, the art of writing in such a way as to be incomprehensible except to those who possess the key to the system employed. The unravelling of the writing is called deciphering.

Ciphertext

Oah!c.4aY!5hutMPtUPGRGLEtGLtAGNFCPu
RF~~C~~tyPRtMDtUPGRGLEtGLtQSAFtytUyWtyQ
RMtz~~C~~tGLAMKNP~~C~~FCLQGzJ~~C~~tCVACNRtRM
RFM~~Q~~CtUFMtNMQQCQQtRFCtICWtrMtrFC
QWQR~~C~~Kt~~C~~KNJMWC~~B~~vtcFCtSLPyTCJJGLEtMD
RFCtUPGRGLEtGQtAyJJCBtB~~C~~AGNF~~C~~PGLEv

POLYALPHABETISCHE SUBSTITUTION - VIGENÈRE CIFFRE

- Im Gegensatz zur monoalphabetischen Substitution werden hier viele („poly“) Geheimalphabete zum Ersetzen der Buchstaben genommen:

Plaintext	H	e	l	l	o	...
Schlüssel	3	1	4	3	1	...
Ciphertext	K	f	p	o	p	...

- Einfache Vorschrift:
$$C_i = P_i + K_i \pmod{m} \quad (\text{mod } n)$$
- Wenn $m \ll n$ ergeben sich wieder statistische Strukturen

Plaintext

CRYPTOGRAPHY, or writing in cipher, the art of writing in such a way as to be incomprehensible except to those who possess the key to the system employed. The unravelling of the writing is called deciphering.

Ciphertext

JaWS620ZkNUV5R5bcUp1GwHoUgA7,4661P9
GvydZpG7i1MgyGr1Aupq.8Fr,3MZcU!BioM
dmi!RZ.jMksEsBm.qv!f0M3EAcsGjNw
rulm0MgoM8sBGMmcqiqb0M91W8wBjNp3
FvmaV,cCksyCSm2 iQb0MeuP!yrzFq.eil;
6ltuuvHCv58vp4YR sCb!qswq;froc9XW

PERFEKTE GEHEIMHALTUNG - ONE-TIME PAD

- Substitution wobei P und K gleich lang sind

Plaintext	A	U	F	S	T	A	N	D
Schlüssel	J	A	T	U	C	O	B	I
Ciphertext	J	U	Y	M	V	O	O	L

- Einfache Vorschrift:

$$C_i = P_i + K_i \pmod{n}$$

- Das Verfahren ist allerdings nicht praxistauglich, da der Schlüssel

- genauso lang sein muss wie die Nachricht
- wirklich zufällig generiert werden muss

Plaintext

CRYPTOGRAPHY, or writing in cipher, the art of writing in such a way as to be incomprehensible except to those who possess the key to the system employed. The unravelling of the writing is called deciphering.

Ciphertext

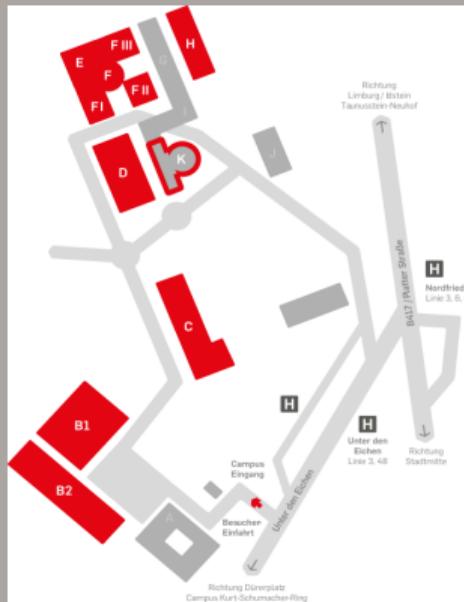
Vga04vNz1C.thq;t5Yk9p,8,0,luvmpXKX
hMI4LahiScEG92mAdqa!C8uzzXT6GZ,uQ2J
9SfEWdvdnI;UDro01e8y1fPfMqXvHL G
QQy.8,13Gm;7sP106 ;L t.OLtX;s3nN
!wSYa8wshgwYQx;HXdnNMosXRV2SooTsIZZ
EG4lXe,9UQyEahM.Q,2KPjLx6S;rJ;5Pcx

DISKUSSION IN KLEINEN GRUPPEN

Tauschen Sie sich mit Ihrem Nachbarn 5 Minuten aus:

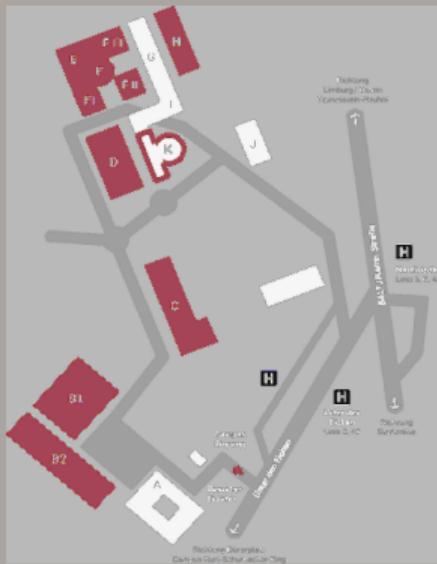
- Diskutieren Sie darüber wieviel von der Campuskarte zu erkennen ist, wenn diese mit den verschiedenen Verschlüsselungsverfahren verschlüsselt wird.

Original



AUSWIRKUNG VON VERSCHLÜSSELUNG

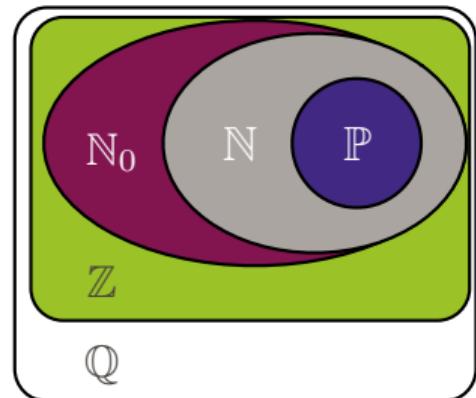
Caesar



Vigenère

MENGEN

- \mathbb{N} : Menge aller positiven Zahlen **ohne** Null: $\{1, 2, 3, \dots\}$
- \mathbb{N}_0 : Menge aller positiven Zahlen **mit** Null: $\{0, 1, 2, 3, \dots\}$
- \mathbb{Z} : Menge aller ganzen Zahlen: $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
- \mathbb{P} : Menge aller Primzahlen: $\{2, 3, 5, 7, 11, 13, \dots\}$
- \mathbb{Q} : Menge der rationalen Zahlen:
$$\left\{ \frac{a}{b} \in \mathbb{Z}, b \neq 0, \text{ggT}(a, b) = 1 \right\}$$
- \mathbb{Z}_m Restklassenring modulo m: $\{1, 2, 3, \dots, m - 1\}$



OPERATOREN

- a plus b : $a + b$
- a mal b : $a \cdot b$
- a teilt b : $a \mid b \Rightarrow (\exists x \in \mathbb{Z}) b = x \cdot a$
- a teilt nicht b : $a \nmid b \Rightarrow (\exists x \in \mathbb{Z}) b \neq x \cdot a$
- Größter gemeinsamer Teiler von a und b : $x = ggT(a, b)$
- Kleinstes gemeinsames Vielfaches von a und b : $x = kgV(a, b)$
- Divisionsrest, wenn man a durch b teilt: $x = a \mod b$

MONOID $\langle M, \circ \rangle$

- $\langle M, \circ \rangle$ ist ein algebraisches System
- M ist eine nicht leere Menge $M = \{a, b, \dots\}$
- \circ ist ein Operator auf Elementen aus M , $\circ = \{+ \text{ oder } \cdot\}$
- Ein Monoid hat folgend Eigenschaften:
 1. Nicht leere Menge: $a, b \in M \Rightarrow a \circ b \in M$
 2. Assoziativ Gesetz: $a, b, c \in M \Rightarrow (a \circ b) \circ c = a \circ (b \circ c)$
 3. Neutrales Element: $a \in M \Rightarrow \exists e, a \circ e = e \circ a = a$
 - Addition: $e = 0$
 - Multiplikation $e = 1$

GRUPPE $\langle G, \circ \rangle$

- G ist eine nicht leere Menge $G = \{a, b, \dots\}$
- \circ ist ein Operator auf Elemente aus G , $\circ = \{+ \text{ oder } \cdot\}$
- Ein Gruppe hat folgend Eigenschaften:
 1. Ist ein Monoid: $\langle G, \circ \rangle$, $\exists e = 0$ bzw. 1
 2. Kommutativ-Gesetz: $a, b \in G \Rightarrow a \circ b = a \circ b$
 3. Inverses Element: $a \in G \Rightarrow \exists a', a \circ a' = e$
 - Addition: $a' = -a$
 - Multiplikation $a' = a^{-1}$
- Eine Algebra $\langle HG, \circ \rangle$ heißt Halbgruppe, wenn sie in Bezug auf die Operation \circ dem Assoziativgesetz genügt.
- Sie wird kommutative Halbgruppe genannt, wenn die Operation \circ zusätzlich kommutativ ist.

RING $\langle R, +, \cdot \rangle$

- R ist eine nicht leere Menge $R = \{a, b, \dots\}$
- Operator auf Elementpaare aus $R, +, \cdot$
- Ein Ring hat folgend Eigenschaften:
 1. Ist eine kommutative Gruppe: $\langle R, + \rangle$, $\exists e = 0$ und $\exists a' = -a$
 2. Distributiv-Gesetz: $a, b, c \in G \Rightarrow a \circ (b + c) = (a \circ b) + (a \circ c)$
 3. ist ein Monoid: $\langle R, + \rangle$, $a \in R : \exists e = 1$ und $\exists a' = -a$
 4. Inverses Element: $a \in G \Rightarrow \exists a', a \circ a' = e$
 - Es existiert zwar ein **inverses Element $a' = -a$** bezüglich der Addition aber **keine multiplikative Inverse**.

KÖRPER $\langle K, +, \cdot \rangle$

- K ist eine nicht leere Menge $K = \{a, b, \dots\}$
- Operator auf Elementpaare aus $K, +, \cdot$
- Ein Körper hat folgend Eigenschaften:
 1. Ist ein kommutativer Ring mit Einselment:
 - $\langle K, +, \cdot \rangle, a \in K$ mit $a \neq 0 \Rightarrow \exists a^{-1} \in K$ mit $a \cdot a^{-1} = 1$
- Jeder Körper $\langle K, +, \cdot \rangle$ und jeder Ring $\langle K, +, \cdot \rangle$ ist **nullteilerfrei**
 - $\forall a, b \in R \setminus 0$ gilt: $a \cdot b \neq 0$
 - $\forall a, b \in K \setminus 0$ gilt: $a \cdot b \neq 0$
- $\mathbb{Z}_m := \{0, 1, \dots, m - 1\}$ ist **genau dann** ein Körper, wenn **m** eine Primzahl ist. \mathbb{Z}_p mit $p \in \mathbb{P}$ ist ein Körper.

ALGEBRA ÜBERSICHT

Struktur		Bez.	Formel (Axiom)		
Algebra ($A, +, \cdot$)		Algebra ($A, +$)			
Körper	Ring	abelsche Gruppe	HG	assoz.	$a + (b + c) = (a + b) + c$
			additive Gruppe	$\exists 0$	$0 + a = a$
				$\exists -a$	$a + (-a) = 0$
				komm.	$a + b = b + a$
				distri.	$a(b + c) = ab + ac$
	Einselem.	abelsche Gruppe	Algebra (A_0, \cdot)		
			HG	assoz.	$a(bc) = (ab)c$
			multipl. Gruppe	$\exists 1$	$1a = a$
				$\exists a^{-1}$	$a a^{-1} = 1$
				komm.	$a b = b a$

MODULARE ARITHMETIK

→ für $a \in \mathbb{Z}$ und $n \in \mathbb{N}$ gibt es eindeutige Quotienten $q \in \mathbb{Z}$ mit Rest r mit:

1. $q = \frac{a}{n}$
2. $r = a \bmod n$

→ Rechenregeln für Modulare Arithmetik:

- Addition: $(a \bmod n) + (b \bmod n) = a + b \bmod n$
- Subtraktion: $(a \bmod n) - (b \bmod n) = a - b \bmod n$
- Multiplikation: $(a \bmod n) \cdot (b \bmod n) = a \cdot b \bmod n$
- Exponentiation: $(g^a \bmod n)^b \bmod n = (g^a)^b \bmod n = g^{a \cdot b} \bmod n$

DEFINITION RESTKLASSE

- Modulo n sind alle Werte $a = i \cdot n + r$ für $i \in \mathbb{Z}$ äquivalent.
- Die Menge $\{i \cdot n + r \mid i \in \mathbb{Z}\}$ wird als Restklasse von r bezeichnet, wobei r ein Repräsentant der Restklasse ist.
- Die Menge aller Restklassen modulo n wird geschrieben als \mathbb{Z}_n .
- Zwei Zahlen $a, b \in \mathbb{Z}$ heißen restgleich, wenn $a \bmod n = b \bmod n \Rightarrow a \equiv b \bmod n$ (a ist **kongruent** zu b modulo n).
- Beispiel: \mathbb{Z}_3 (Zahlen aus \mathbb{Z} modulo 3) besteht aus den folgenden Restklassen:
 - Restklasse für $r = 0 : \{ \dots, -9, -6, -3, \textcolor{red}{0}, 3, 6, 9 \dots \}$
 - Restklasse für $r = 1 : \{ \dots, -8, -5, -2, \textcolor{red}{1}, 4, 5, 10 \dots \}$
 - Restklasse für $r = 2 : \{ \dots, -7, -4, -1, \textcolor{red}{2}, 5, 8, 12 \dots \}$

RING UND RESTKLASSENRING

- Ein Ring $\langle R, + \cdot \rangle$ ist eine algebraische Struktur bei der:
 - $\langle R, \cdot \rangle$ eine **Halbgruppe** bildet
 - $\langle R, + \rangle$ eine **abelsche Gruppe** bildet
 - Distributivgesetze gelten:
 - Linke Distributivität: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ für $\forall a, b, c \in R$
 - Rechte Distributivität: $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ für $\forall a, b, c \in R$
- Ein Ring $\langle \mathbb{Z}_n, + \cdot \rangle$ über einer Restklasse \mathbb{Z}_n wird als **Restklassenring** bezeichnet.

BEISPIEL RESTKLASSENRING $(\mathbb{Z}_6, +, \cdot)$

$\langle \mathbb{Z}_6, + \rangle$ ist eine Gruppe:

+	0	1	2	3	4	5
0	0	0	0	0	0	0
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$\langle \mathbb{Z}_6, \cdot \rangle$ ist eine Halbgruppe:

+	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Zusätzlich zur Halbgruppe:

- $\langle \mathbb{Z}_6, \cdot \rangle$ hat das neutrale Element 1
- $\langle \mathbb{Z}_6, \cdot \rangle$ ist für manche Elemente invertierbar

INVERTIERBARKEIT DER MULTIPLIKATION

- Für $\langle \mathbb{Z}_n, \cdot \rangle$ sind nicht alle Elemente invertierbar
- Aber: Teilerfremde Zahlen $z \in \mathbb{Z}$ zu n sind invertierbar ($\text{ggT}(n, z) = 1$)

Ergebnistabelle $\langle \mathbb{Z}_6, \cdot \rangle$

+	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Für $\langle \mathbb{Z}_6, \cdot \rangle$ sind nur 1 und 5 invertierbar

Ergebnistabelle $\langle \mathbb{Z}_5, \cdot \rangle$

+	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

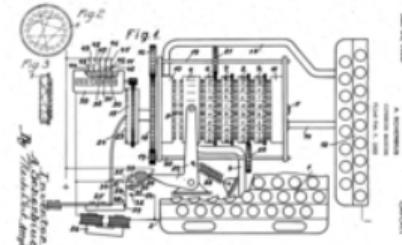
Für $\langle \mathbb{Z}_5, \cdot \rangle$ sind nur 1, 2, 3, 4 invertierbar,
da 5 eine Primzahl ist.

HILL-CHIFFRE

- Entwickelt von Lester S. Hill; 1891-1961, US-amer. Mathematiker, Lehrer und Kryptograph
- Ausgangslage:
 - Restklassenring
 - In einem Körper existieren die **modular Inversen**
- Algorithmus:
 - Verschlüsselung: $C = P \cdot K \text{ mod } p$
 - Entschlüsselung: $C = P \cdot K^{-1} \text{ mod } p$
 - C, P, K sind Matrizen

WAS BEDEUTET SICHERHEIT?

- Enigma wurde im 2. Weltkrieg zur Verschlüsselung genutzt
 - Verschlüsselung basiert auf Polyalphabetische Substitution
 - Analyse war aufgrund komplexer Rotormechanik sehr schwierig
- Um Ciphertexte zu entschlüsseln, nutzten die Alliierten verschiedene Tricks:
 - Teile des Plaintexts waren bekannt (Datum, Absendername,...)
 - Inhalt bestimmter Nachrichten konnte frei gewählt werden (Alliiertes Schiff hat an Position X gehalten)
- Angriffsmodelle werden genutzt, um diese Angriffe zu formalisieren
 - Moderne Verschlüsselungsverfahren müssen bestimmten Angriffsmodellen standhalten



Quelle: https://de.wikipedia.org/wiki/Enigma_Maschine - letzter
Besuch 02.04.23

ANGRIFFSMODELLE

Beispiel: Abfangen von Alices's stud.ip Passwort

- Kontext: Eve ist im selben Raum wie Alice und fängt alle verschlüsselten Pakete ab
- Ziel von Eve: Die Zugangsdaten von Alice

Angriffsmodell	Beschreibung	Beispiel Szenario
Ciphertext-Only	Eve ist nur der Ciphertext bekannt	Nur verschlüsselte Zugangsdaten sind bekannt.
Known-Plaintext	Eve erhält zufällige Plaintext/Ciphertext Paare	Alice loggt sich auf ihrem Konto ein und surft auf bekanntem Teil von stud.ip
Chosen-Plaintext	Eve hat Zugriff auf ein Verschlüsselungssorakel, das beliebige Plaintexte verschlüsselt	Eve sendet eine Nachricht an Alice. Alice loggt sich ein und ruft Eve's Nachricht ab.
Chosen-Ciphertext	Eve hat Zugriff auf ein Entschlüsselungssorakel, das beliebige Ciphertexte entschlüsselt	Eve hat für begrenzte Zeit Zugriff auf Alice's Gerät mit verschlüsselter Sitzung (ohne bestehenden Login) und lässt sich manipulierte verschlüsselte Nachricht entschlüsseln. Alice kommt später wieder und loggt sich auf Webseite ein.

DISKUSSION IN KLEINEN GRUPPEN

Tauschen Sie sich mit Ihrem Nachbarn 5 Minuten aus:

- In welchen Angriffsmodellen ist die monoalphabetische Substitution sicher?
- Gibt es eine Hierarchie unter den Angriffsmodellen?

Angriffsmodelle

Ciphertext-Only

Known-Plaintext

Chosen-Plaintext

Chosen-Ciphertext

KRYPTANALYSE

Kryptanalyse beschäftigt sich mit Methoden und Techniken, um Verschlüsselungen zu brechen. Das Brechen eines Kryptoverfahrens ist in folgende Kategorien eingeteilt:

→ **absolut sicher:**

- wenn nicht genug Information gewonnen werden kann, um hieraus den Klartext oder den Schlüssel zu rekonstruieren.

→ **analytisch sicher:**

- wenn es kein nichttriviales Verfahren gibt, mit dem es systematisch gebrochen werden kann.

→ **komplexitätstheoretisch sicher:**

- wenn es keinen Algorithmus gibt, der das Kryptoverfahren in Polynomialzeit in Abhängigkeit der Schlüssellänge brechen kann.

→ **praktisch sicher:**

- wenn kein Verfahren bekannt ist, welches das Kryptoverfahren mit vertretbarem Ressourcen-, Kosten- und Zeitaufwand brechen kann.

PRAKTISCHES QUANTIFIZIEREN DER SICHERHEIT (1/2)

- Krypto Verfahren sind in der Praxis "ungebrochen", solange **Brute-Force** der effizienteste Angriff ist
 - **Brute-Force:** Testen aller möglichen Schlüsselkombinationen
 - Komplexität der Brute-Force Angriffe steigt exponentiell in der Schlüssellänge
- Es existieren verschiedene Stufen des **Brechens**
 - **Theoretisch Gebrochen:** Ein effizienterer Angriff als Brute-Force wird bekannt
 - **Überholt:** Der Aufwand fällt unter eine Grenze, die mit viel Rechenkapazität erreichbar wäre
 - **Praktisch Gebrochen:** Ein Angriff wurde demonstriert
- Solange ein Verfahren noch nicht als überholt gilt, reden wir von "**Rechnerischer Sicherheit**"

PRAKTISCHES QUANTIFIZIEREN DER SICHERHEIT (2/2)

Rechnerische Sicherheit: Ein Krypto Verfahren ist zwar theoretisch zu brechen, praktisch existieren aber nicht genug Zeit oder Ressourcen

Anzahl der Schlüs-selbits	Anzahl der Schlüs-selkombinationen	Zeitaufwand für Brute-Force in Jahren (Annahme: Pro Kombination eine Operation der gesamten Top500 Supercomputer mit $4,9 \cdot 10^{18}$ Operatio-nen pro Sekunde in Nov 2022)	Beispielverfahren
8	256	0	—
32	$4.294.967.296$	0	—
64	$1,84 \cdot 10^{19}$	0	Simon32/64
80	$1,21 \cdot 10^{24}$	0.008	PRESENT
128	$3,40 \cdot 10^{38}$	$2,20 \cdot 10^{12}$	AES-128
192	$6,28 \cdot 10^{57}$	$4,06 \cdot 10^{31}$	Serpent-192
256	$1,58 \cdot 10^{77}$	$7,49 \cdot 10^{50}$	Chacha20

KERCKHOFFS PRINZIP

- Wie kann garantiert werden, dass ein Verfahren auch wirklich sicher ist?
 - Wurden Angriffe beim Design übersehen? [Tews12]
 - Haben Entwickler Hintertüren in das Verfahren eingebaut? [BD+21]
- **Kerckhoffs Prinzip:** Die Sicherheit des Verfahrens muss auf der Geheimhaltung des Schlüssels beruhen anstatt auf der Geheimhaltung des Verfahrens selbst
 - Klassische Kryptographie ist geprägt vom Wechselspiel zwischen Kryptographie und Krypanalyse (Erkenntnisse ⇒ Entwicklungen).
 - Die Sicherheit eines Kryptosystems darf nicht von dessen Geheimhaltung, sondern nur von der Schlüssellänge abhängen.
- Seien $\mathcal{P}, \mathcal{C}, \mathcal{K}$ die Mengen der Plaintexte, Chiffretexte bzw. Schlüssel und $\mathcal{E} : \mathcal{P} \times \mathcal{K} \rightarrow \mathcal{C}$ ein Verschlüsselungssystem. Ist ein Kryptoanalytiker im Besitz eines Plaintext-Chiffretextpaars $(P, C) \in \mathcal{P} \times \mathcal{C}$, so kann der verwendete Schlüssel K durch vollständige Suche ermittelt werden, da $\mathcal{E}(P, K) = C$ gelten muss.

STANDARDISIERUNG VON KRYPTOALGORITHMEN

- Kryptoverfahren werden via öffentlicher Ausschreibung standardisiert
 - Jede Person darf ein Verfahren einreichen
 - Verfahren müssen Rahmenbedingungen einhalten (z.B., transparentes Design, Schlüssellänge)
 - Die Verfahren werden über mehrere Jahre von Experten analysiert
 - Der Gewinner wird aus der Menge der übrig gebliebenen Verfahren ausgewählt
- Standardisierungsprozesse von der NIST:
 - Advance Encryption Standard (AES) - 2000
 - Kryptographische Hashfunktionen SHA3 - 2015
 - Lightweight Kryptographie - Gewinner bestimmt in 2023
 - Post-Quanten Kryptographie (PQC) - aktiv seit 2017

ZUSAMMENFASSUNG

- Funktionale Anforderungen an die Sicherheit von Verschlüsselungsverfahren verstehen
- Anhand der Berechnungskomplexität sichere von unsicheren Verfahren unterscheiden können
- In Grundzügen die mathematischen Probleme kennen, auf denen asymmetrische Verfahren beruhen



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Verschlüsselung

April 16, 2023

Marc Stöttinger



We need to think about encryption not as this sort of arcane, black art. It's a basic protection.

Edward Snowden

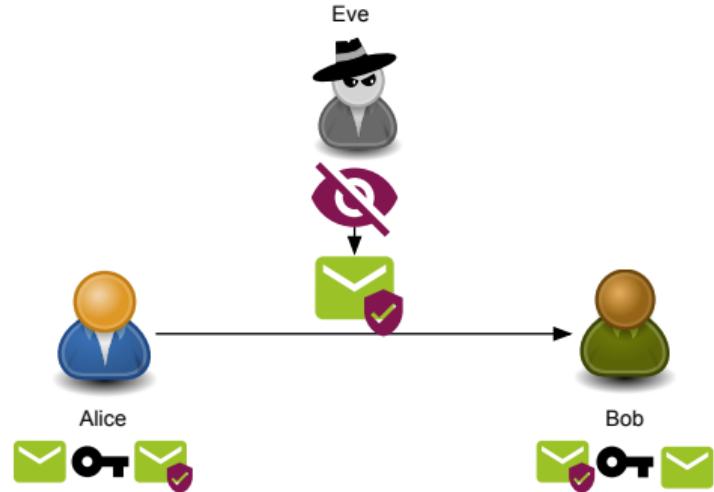
WIEDERHOLUNG: VERTRAULICHKEIT DURCH VERSCHLÜSSELUNG

→ **Bedrohung:**

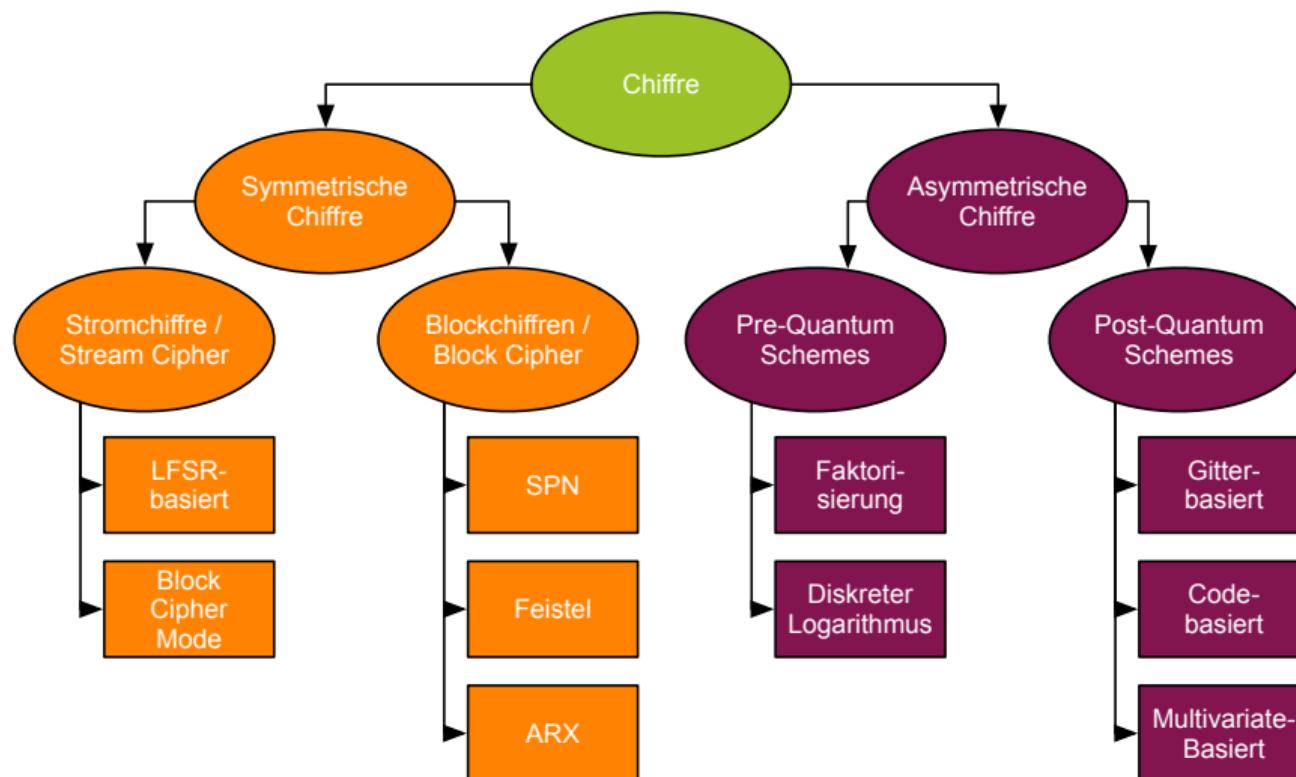
Eve liest die Nachricht mit

→ **Ziel:**

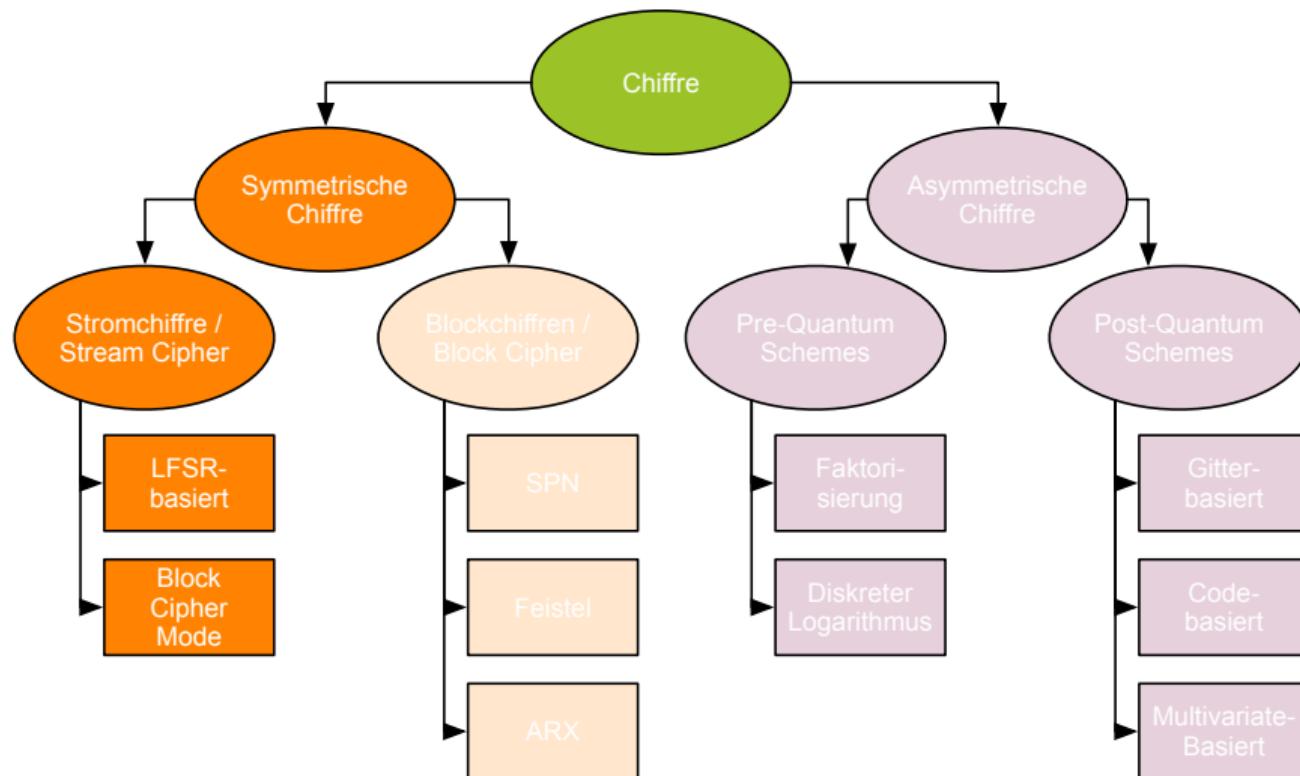
Personen ohne den entsprechenden Schlüssel können keine Informationen aus verschlüsselter Nachricht gewinnen



ÜBERSICHT VON VERSCHLÜSSELUNGEN



STROMCHIFFREN



PERFEKTE GEHEIMHALTUNG - ONE-TIME PAD

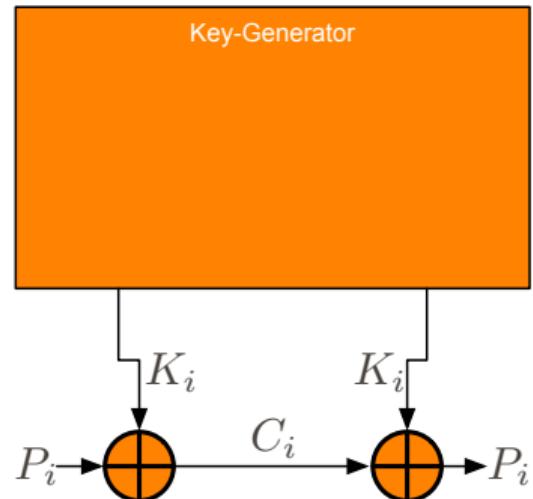
- Substitution wobei P und K gleich lang sind

Plaintext	A	U	F	S	T	A	N	D
Schlüssel	J	A	T	U	C	O	B	I
Ciphertext	J	U	Y	M	V	O	O	L

- Vorschrift für Binärdaten:

$$C_i = P_i \oplus K_i \pmod{n}$$

- Die einzelnen Schlüsselbits K_i können durch einen Key-Generator erzeugt werden.
- K_i wird dann auch als Schlüsselstrom (Key stream) bezeichnet.

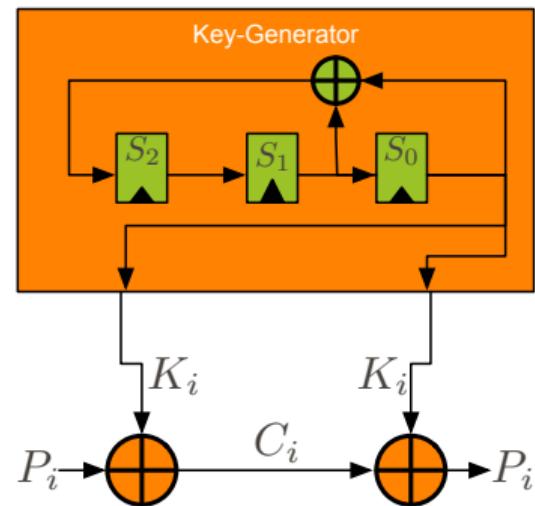


STROMCHIFFRE DESIGN

- Eine einfache Linear Feedback Shift Register (LFSR) Schaltung wird zum Erzeugen von K_i genutzt.

clk	FF_2	FF_1	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

$$\rightarrow s_{i+3} \equiv (s_{i+1} \oplus s_i) \bmod 2$$



PRIMITIVE POLYNOM BASED LFSR

- Generalized form of an LFSR:

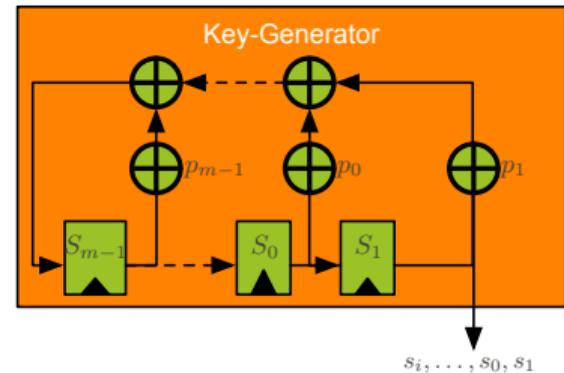
$$s_{i+m} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \bmod 2;$$

$$s_i, p_i \in \{0, 1\}; i = 0, 1, 2, \dots$$

- Primitive Polynomials, a special type of irreducible polynomials, have the form

$$P(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_1x + p_0$$

- Only primitive polynomials generate a maximum sequence of $2^m - 1$.



Maximale Sequenzlänge

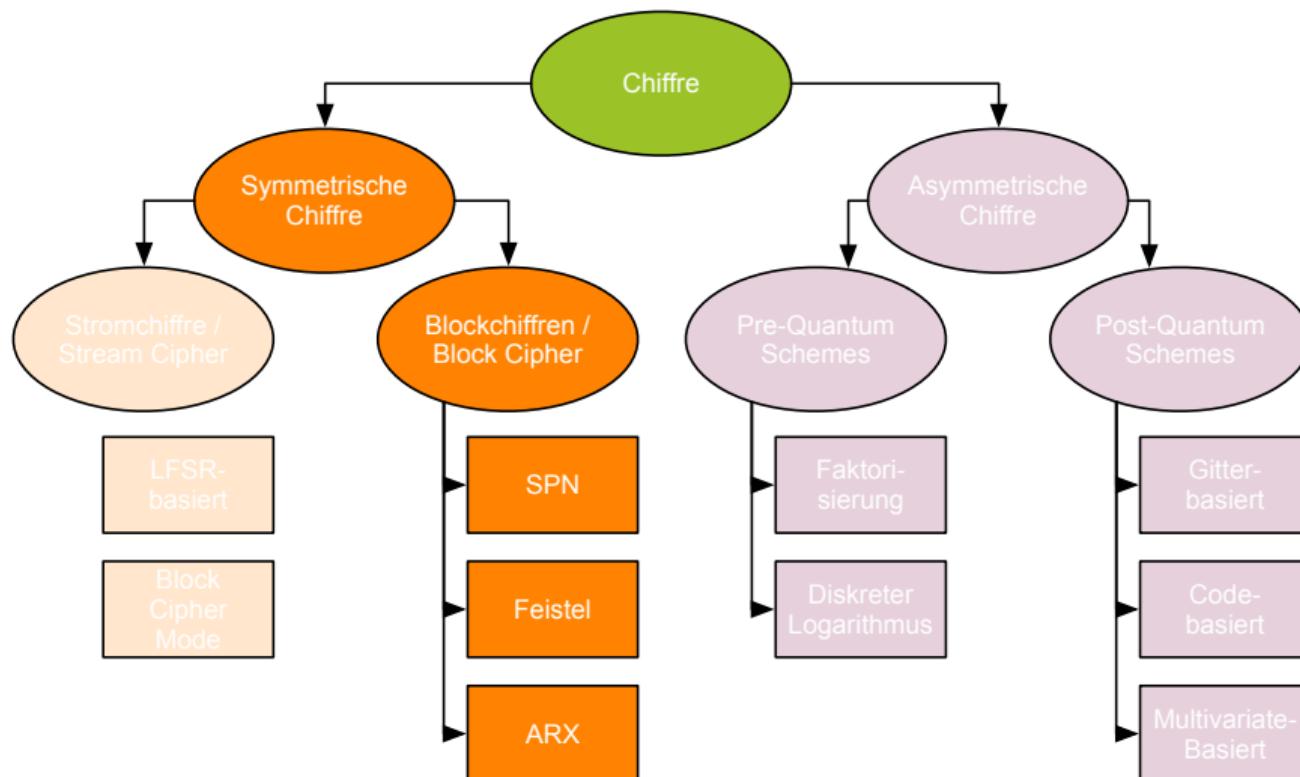
The maximum sequence length, which can be generated by an LFSR of degree m , is $2^m - 1$.

KEY-GENERATOR FÜR STROMCHIFFREN

- Die Schlüsselgeneratoren von modernen Stromchiffren haben meist einen großen internen Zustand.
- Zur Konstruktion der zufälligen Schlüsselsstromsequenz werden meist mehrere LFSR Konstruktionen verwendet .
- Der geheime Schlüssel wird zur Initialisierung des internen Zustands benutzt.

Chiffre	Erstellungsdatum	Schlüssellänge	Interner State	Komplexität bester Angriff
RC4	1987	8-2048 Bits	2064 Bits	2^{13} oder 2^{33}
A5/2	1989	54 Bits	64 Bits	komplett gebrochen
MICKEY	2004	80 Bits	200 Bits	$2^{32.5}$
Trivium	2004	80 Bits	288 Bits	2^{135}
Salsa20	2004	256 Bits	512 Bits	2^{251} (für 8 Runden)

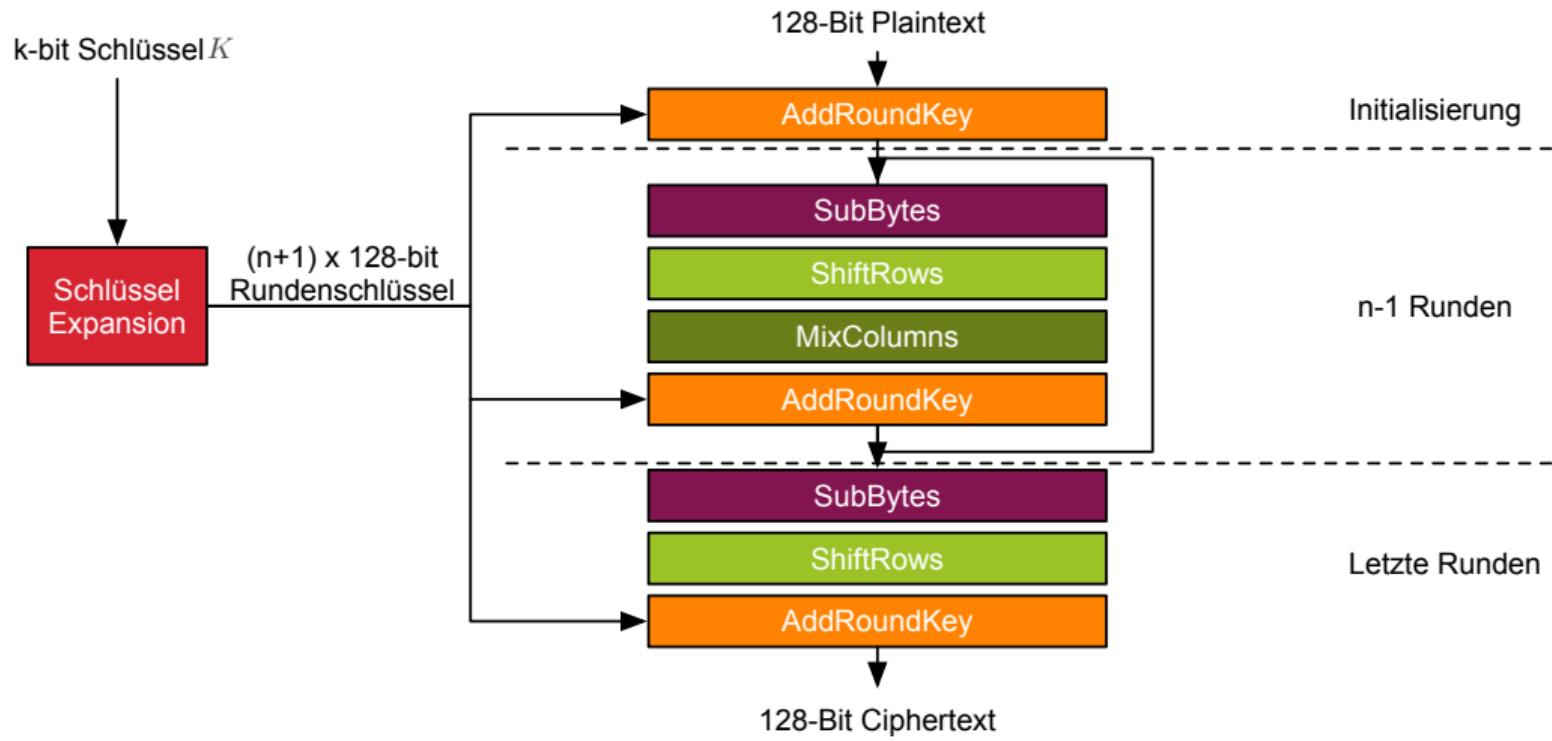
BLOCKCHIFFREN



ADVANCE ENCRYPTION STANDARD (AES)

- In 2000 wurde Rjindael zum Sieger einer Ausschreibung gekürt und als **AES** standardisiert
- AES ist eine **Blockchiffre**, die auf 128-bit Blöcken arbeitet
 - **Blockchiffre**: Der Plaintext wird in Blöcke eingeteilt und blockweise verarbeitet
 - **Stromchiffre**: Zeichen werden einzeln verarbeitet (z.B., monoalphabetische Substitution, One-Time Pad)
- Die Blöcke werden in n Runden durch ein Substitutions-Permutations-Netzwerk (SPN) verschlüsselt
- Es existieren drei AES Varianten mit Schlüssellänge K und Rundenanzahl n :
 - AES-128: $K = 128$ -bit Schlüssel mit $n = 10$ Runden
 - AES-192: $K = 192$ -bit Schlüssel mit $n = 12$ Runden
 - AES-256: $K = 256$ -bit Schlüssel mit $n = 14$ Runden

ADVANCE ENCRYPTION STANDARD (AES)

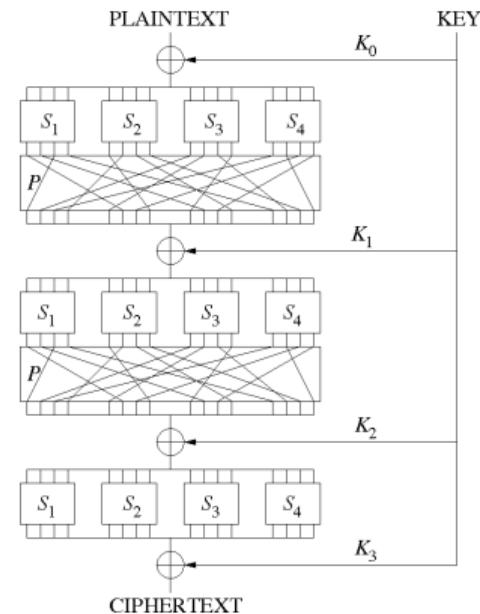


MODERNE METRIKEN FÜR KRYPTOGRAPHISCHE ALGORITHMEN

- Shannonsche Theorie
 - Wichtige **Konstruktionsprinzipien** für die kryptographische Sicherheit sind **Konfusion** und **Diffusion**.
- **Konfusion:**
 - Die Konfusion einer Blockchiffre ist dann groß, wenn die statistische Verteilung der Chiffretexte in Abhängigkeit von der Verteilung der Klartexte für den Angreifer zu groß ist (keine Ausnutzbarkeit).
 - Meistens wird die **S-Box** als nicht-lineares Element in der Blockchiffre für die Konfusion genutzt.
- **Diffusion:**
 - Die Diffusion einer Blockchiffre ist dann groß, wenn jedes einzelne Bit des Klartextes (und des Schlüssels) möglichst viele Bits des Chiffretextes beeinflusst (typisch etwa 50 %).
 - **Permutationen** oder **Schiebeoperationen** werden in Blockchiffren genutzt, um die Diffusion zu realisieren.

SUBSTITUTIONS-PERMUTATIONS-NETZWERK-CHIFFRE (SPN)

- Der Plaintext \mathcal{P} wird in mehrere gleiche große Blöcke aufgeteilt $P_1, P_2, \dots, P_n \in \mathcal{P}$
- Die Verschlüsselungsvorschrift besteht aus einer mehrfach wiederholten Rundenfunktion $f_R(\cdot)$ mit individuellem Rundenschlüssel K_i
- Die Rundenfunktion besteht aus einer nichtlinearen Sbox und einer Permutation.
- Für die Entschlüsselung wird die Umkehrfunktion $f_R^{-1}(\cdot)$ zu $f_R(\cdot)$ benötigt.

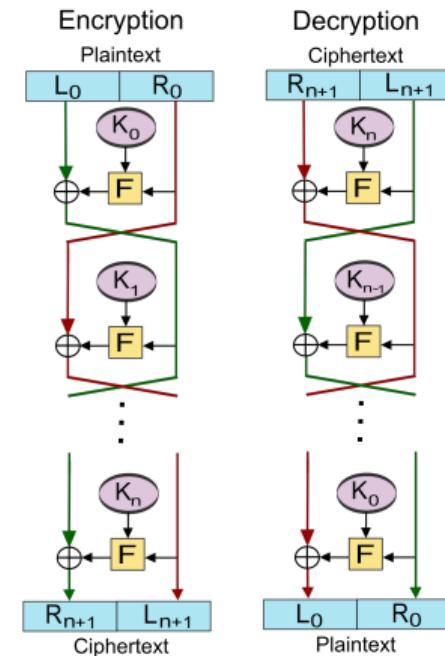


Quelle:

<https://de.wikipedia.org/wiki/Substitutions-Permutations-Netzwerk>

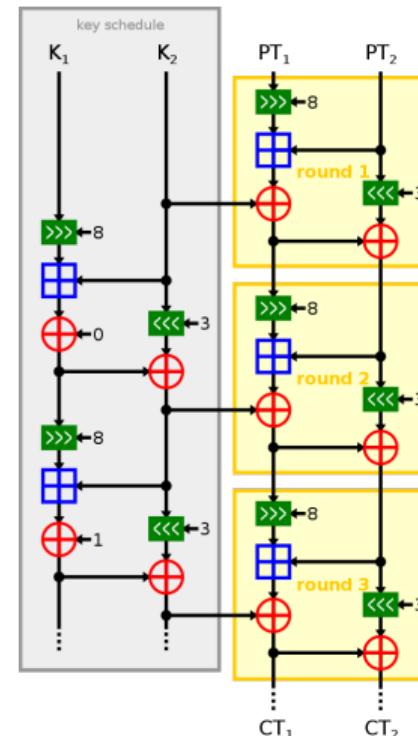
FEISTEL-CHIFFRE (LUBY-RACKOFF BLOCKCHIFFREN)

- Basiert auch auf der Mehrfachausführung von Rundenfunktionen mit Rundenschlüsseln.
- Plaintext wird in zwei Blöcke (L und R) aufgeteilt, die nach jeder Runde vertauscht werden.
- Verschlüsselung und Entschlüsselung kann mit den gleichen Rundenfunktionen $f_R(\cdot)$ ausgeführt werden
- Das Design von $f_R(\cdot)$ ist schwieriger als bei SPN-Chiffren



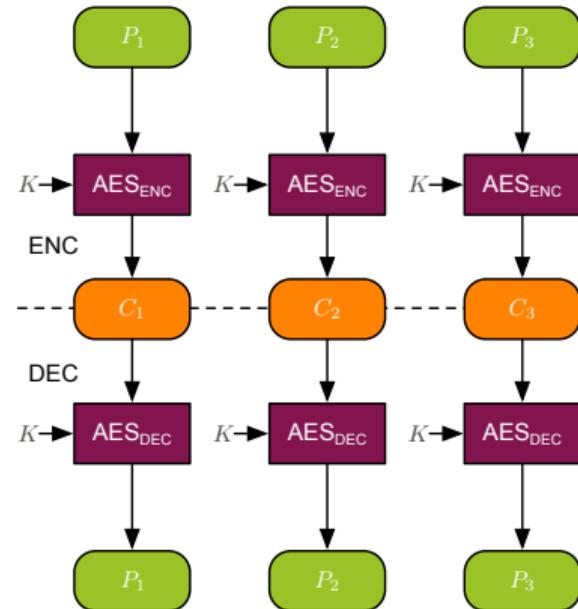
ADD-ROTATE-XOR-CHIFFRE

- ARX-Chiffren benutzen als Basisoperationen nur Addition, Rotation und XOR
- Dadurch sind diese sehr kompakt implementierbar und effizient für Standardprozessoren
- Nicht bester Trade-off bei der Umsetzung in Hardware
- Die Resistenz gegen kryptanalytische Angriffe noch nicht umfänglich, da es recht junge Verfahren sind



ELECTRONIC CODE BOOK (ECB)

- AES verarbeitet die 128-bit Blöcke $P_1, P_2, P_3 \in \mathcal{P}$ des Plaintextes unabhängig von einander.
- Auf die gleiche Eingabe erfolgt eine gleiche Ausgabe, ähnlich wie monoalphabetischen Chiffren.
- Spezielle Betriebsmodi sind notwendig!



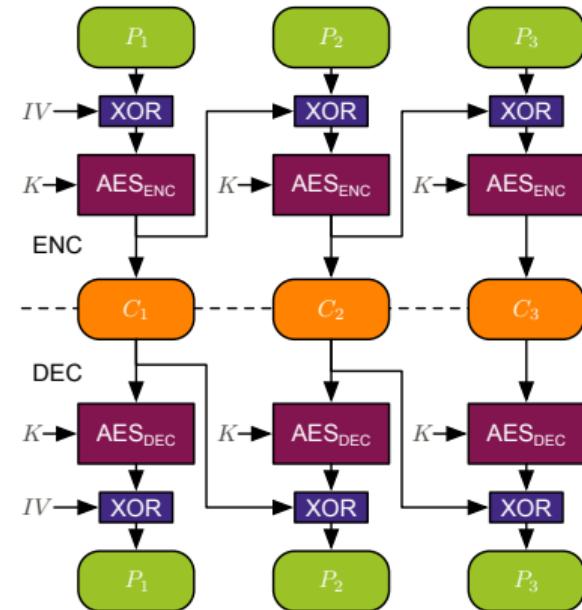
BETRIEBSMODUS VON BLOCKCHIFFREN

Blockchiffren können in verschiedenen Modi betrieben werden

Name	Bezeichnung	Einsatzgebiet
ECB	Electronic Code Book	Einsatz in Ausnahmefällen oder wenn nur ein Block verschlüsselt werden muss
CBC	Cipher Block Chaining	Verschlüsselung bei Datenübertragung
CFB	Cipher Feedback Mode	Verschlüsselung entspricht einer selbstsynchronisierenden Stromchiffre
OFB	Output Feedback Mode	Verschlüsselung mit Fehlerresistenz
CTR	Counter Mode	Verschlüsselung mit Fehlerresistenz; macht aus Blockchiffre eine Stromchiffre
XTS	Ciphertext Stealing	Festplattenverschlüsselung; Besonders gesichert gegen Angriffe auf Implementierung
GMAC/C-MAC	Galois/Cipher Message Authentication Mode	Authentifikation von Daten (Abschnitt „Message Authentication Codes“)
GCM	Galois-Counter Mode	Verschlüsselung und Authentifikation von Daten (Abschnitt „Message Authentication Codes“)

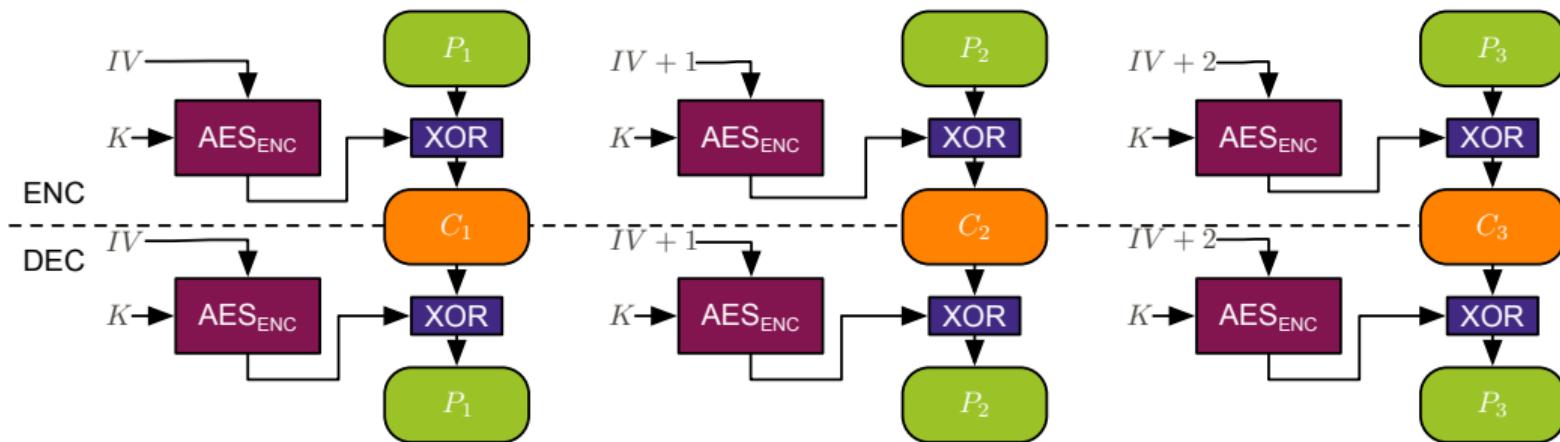
CIPHER BLOCK CHAINING (CBC)

- Ciphertext des vorherigen Blocks fließt in nächsten Block mit ein (via XOR)
- Zufälliger Initialisierungsvector IV , um gleiche Plaintexte $P_1 = P_2$ zu unterschiedlichen Ciphertexten $C_1 \neq C_2$ zu verschlüsseln
- Nachteil ist, dass der Mode nicht parallelisiert ist und Übertragungsfehler propagiert werden

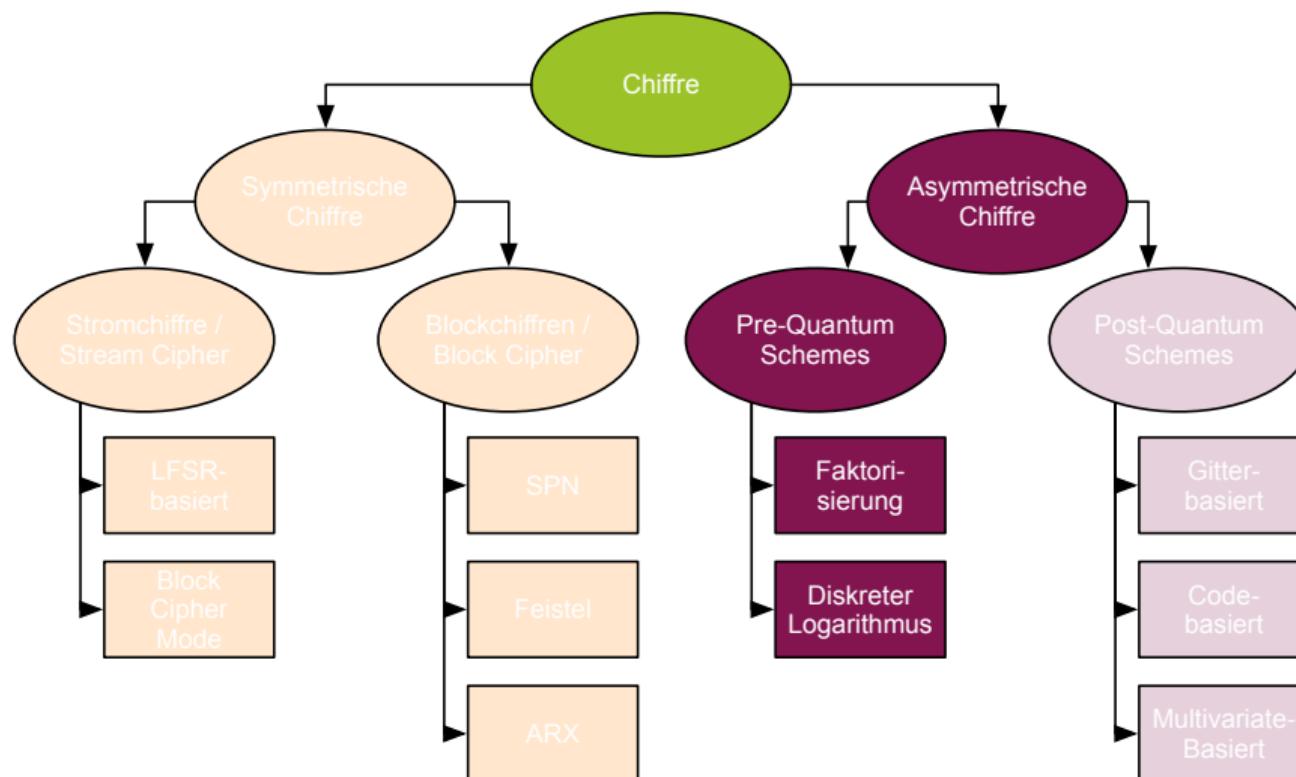


COUNTER MODE (CTR)

- Zufälliger IV wird verschlüsselt und mit Plaintext ver-XORed
 - Hochgradig parallelisierbar und AES kann vorberechnet werden (Stromchiffre)
 - Übertragungsfehler wirken sich nur auf lokalen Block aus
 - Nur die Verschlüsselungsvorschrift wird benötigt für Ver- und Entschlüsselung



ASYMMETRISCHE VERSCHLÜSSELUNGSVERFAHREN



SYMMETRISCHE VS. ASYMMETRISCHE VERSCHLÜSSELUNGSVERFAHREN

- Bei AES benötigen beide Parteien den gleichen, geheimen Schlüssel K
 - AES fällt daher in die Kategorie der **Symmetrischen** oder **Private-Key** Verschlüsselungsverfahren
- Meist existiert aber kein geheimer, ausgetauschter Schlüssel
 - Ad-hoc Kommunikation mit unbekannten Parteien im Internet
 - Jedes Paar Parteien benötigt eigenen Schlüssel ($\frac{m(m-1)}{2}$ bei m Parteien)
- Lösung: **Asymmetrische** oder **Public-Key** Verschlüsselungsverfahren



ASYMMETRISCHE VERSCHLÜSSELUNG GRUNDPRINZIP

1. Empfänger generiert ein Schlüsselpaar K_E, K_D .
 - K_E : **öffentlicher** Schlüssel, der von allen Parteien zum Verschlüsseln genutzt werden kann.
 - K_D : **geheimer** Schlüssel, mit dem Ciphertexte entschlüsselt werden können.
 - K_E und K_D stehen in einer Relation $K_E = f(K_D)$ und $K_D = f^{-1}(K_E)$.
2. Sender nutzt K_E , um Plaintext P mit $C = Enc_{K_E}(P)$ zu verschlüsseln.
3. Nur Empfänger kann C mit $P = Dec_{K_D}(C)$ zu entschlüsseln.
4. Asymmetrische Verfahren basieren auf mathematisch schweren Problemen, um sicherzustellen, dass nicht von K_E auf K_D geschlossen werden kann.

ASYMMETRISCHE VERSCHLÜSSELUNG RIVEST-SHAMIR-ADLEMAN (RSA)

- RSA wurde 1977 entwickelt von R. **R**ivest, A. **S**hamir und L. **A**dleman.
- RSA kann zur asymmetrischen Ver-/Entschlüsselung genutzt werden.
- Die Sicherheit von RSA basiert auf:
 - Dem RSA Problem (e -te Wurzel modulo N)
 - Der Schwierigkeit der Primfaktorzerlegung für große Zahlen
- RSA Ver-/Entschlüsselung mit n -bit Modulus N hat Komplexität $\mathcal{O}(n^3)$
 - Multiplikation zweier n -bit Werte hat $\mathcal{O}(n^2)$
 - Exponentiation mit n -bit Exponent hat $\mathcal{O}(n^3)$
- Schlüsselgenerierung ist sehr rechenintensiv
 - Finden und Verifizieren von Primzahlen

ASYMMETRISCHE VERSCHLÜSSELUNG RIVEST-SHAMIR-ADLEMAN (RSA)

Alice

Bob

SchlüsselgenerierungWähle zufällige Primzahlen p und q Berechne $N = p \cdot q$ Wähle e zufällig mit $\text{ggT}(\phi(N), e) = 1$ Berechne d als: $e \cdot d \mod \phi(N) = 1$ Setze $K_E = (N, e)$ und $K_D = d$

$$K_E = (N, e)$$

VerschlüsselungBerechne $C = P^e \mod N$ **Entschlüsselung**

$$C$$

Berechne $P = C^d \mod N$

EINSCHUB ZUR EULERSCHE PHI-FUNKTION

- $\phi(m)$ gibt die Anzahl derjenigen natürlichen Zahlen $n < m$ an, die teilerfremd zu m sind;
 $m, n \in \mathbb{N}, \phi m = |\{0 \leq n \leq m \mid \text{ggT}(n, m) = 1\}|$
 - Beispiel: $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$, $\phi(5) = 4$, da nur $\text{ggT}(0, 5) \neq 1$ ist für $\forall n \in \mathbb{Z}_5$
- Spezialfälle:
 - $p \in \mathbb{P} \Rightarrow \phi(p) = (p - 1)$
 - $k \in \mathbb{N} \Rightarrow \phi(p^k) = p^{k-1} \cdot (p - 1)$
 - $p, q \in \mathbb{P}$ und $p \neq q \Rightarrow \phi(p \cdot q) = \phi(q) \cdot \phi(p) = (p - 1) \cdot (q - 1)$
- Weitere nützliche Eigenschaften:
 - Wenn $\text{ggT}(a, n) = 1$ ist, dann gilt: $a^{\phi(n)} \pmod{n} = 1$
 - Ist $n = p \in \mathbb{P}$, so ergibt sich der Satz von Fermat: $a^{p-1} \pmod{p} = 1; (a \neq 0)$
 - Somit kann man das modular Inverse berechnen: $a^{-1} \pmod{p} = a^{p-2} \pmod{p}; (a \neq 0)$

WESHALB IST RSA SICHER?

- Öffentlich ist: $K_E = (N, e), C$
- Geheim sind: $K_D = d, p, q, P$
- Berechnung des Plaintextes $C = P^e \pmod{N}$
 - Invertierung: $P = \sqrt[e]{C} \pmod{N} \rightarrow$ Problem der e -ten Wurzel \pmod{N} .
- Alternative: Berechnung des privaten Schlüssels $K_D = d$
 - Bedingung $e \cdot d \pmod{\phi(N)} = 1$
 - Berechne $\phi(N) = (p - 1) \cdot (q - 1) \Rightarrow$ Problem der Primfaktorzerlegung

RSA – SICHERHEIT

- RSA ist als asymmetrisches Verfahren bereits im Chosen-Plaintext Modell
 - Angreifer kann beliebige Plaintexte mit öffentlichem Schlüssel K_E verschlüsseln
- Kurze Plaintexte können via Brute-Force gebrochen werden
 - Telefonnr. (≈ 32 bit): Verschlüsseln aller Nummern mit K_E und Vergleich mit Ciphertext
- Exponent e für Verschlüsselung wird kurz gewählt, um Berechnung zu beschleunigen
 - $e \in \{3, 65537\}$
- Textbuch RSA benötigt weitere Paddingverfahren, um Brute-Force Angriffe auszuschließen
 - **RSA-OAEP Padding**: Nachricht wird um Zufallszahl und Prüfsumme erweitert

IMPLEMENTIERUNG ASYMMETRISCHE VERSCHLÜSSELUNG

- Asymmetrische Verfahren nur sehr schwer sicher zu implementieren [B99]
 - Primzahlen in RSA dürfen weltweit nicht doppelt vorkommen [ND+12]
 - Bestimmte Primzahlen müssen vermieden werden [C96]
 - Bestimmte Werte für d und e müssen vermieden werden
 - Fehler im Paddingverfahren können zur Kompromittierung des Schlüssels führen [B98]
- Etliche Tricks können asymmetrische Verfahren beschleunigen
 - Chinesischer Restsatz
 - Wahl einer Basis aus einer Restklassengruppe mit kleinerer Ordnung
- Implementieren Sie asymmetrische Verfahren **nicht selbst**, sondern nutzen Sie bestehende Bibliotheken!

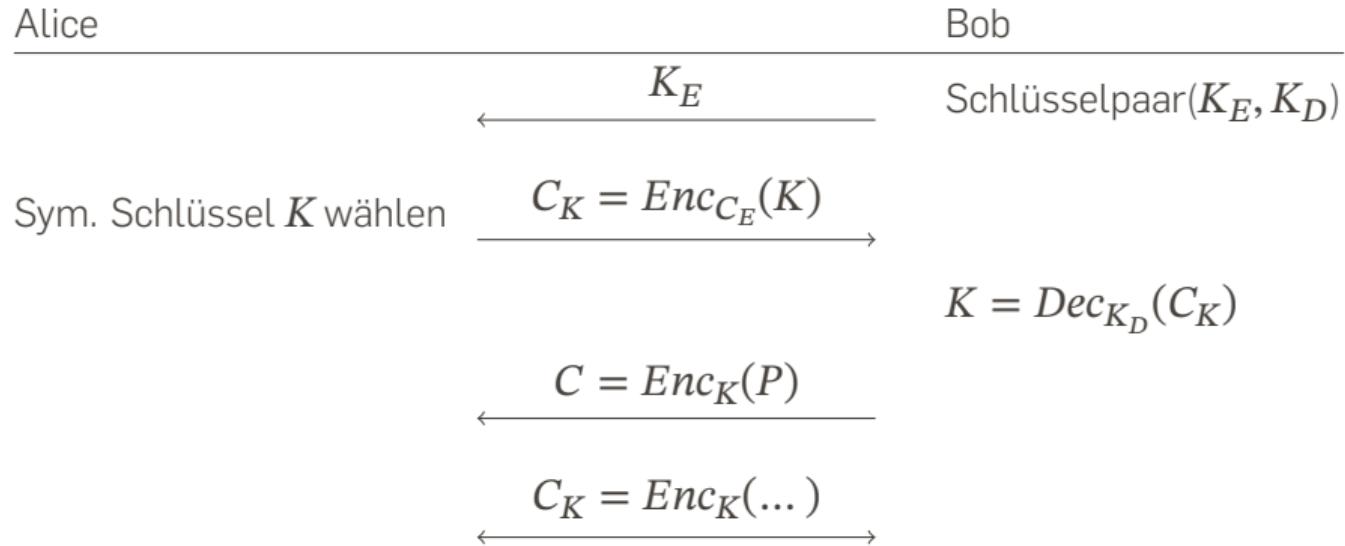
HYBRIDE VERSCHLÜSSELUNG (1/2)

Aspekt	Symmetrische Verschlüsselung	Asymmetrische Verschlüsselung
Vorteile	Sehr schnell (~Gigabyte/Sekunde)	Es muss kein geheimer Schlüssel ausgetauscht sein
Nachteile	Geheimer Schlüssel muss ausgetauscht sein	Langsam (~Hunderte Kilobyte/Sekunde)

→ Hybride Verschlüsselung kombiniert die Vorteile beider Verfahren:

1. Asymmetrische Verfahren, um einen symmetrischen Schlüssel auszuhandeln
2. Symmetrische Verfahren, um die Daten zu übertragen

HYBRIDE VERSCHLÜSSELUNG (2/2)



DIFFIE-HELLMAN VERFAHREN (DH)

- Asymmetrisches Verfahren zur Schlüsselvereinbarung, entwickelt in 1976
- Basiert auf dem diskreten Logarithmusproblem in primen Restklassenringen
- Voraussetzung: Alice und Bob kennen öffentliche Primzahl p und Basis g
 - Mögliche Primzahlen und Basen sind in Standards definiert DHP
- DH kann nicht für Verschlüsselung genutzt werden, sondern nur für Schlüsselvereinbarung
 - DH benötigt weiteres Verschlüsselungsverfahren (z.B. symmetrisches Verfahren)

EINSCHUB ZYKLISCHE GRUPPEN

- Eine Gruppe (G, \circ) hat eine endliche Anzahl von Elementen. Die Anzahl der Elemente gibt die Ordnung (Kardinalität) der Gruppe G mit $|G|$ an.
 - Beispiele:
 - $\mathbb{Z}_9 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\} \Rightarrow |\mathbb{Z}_9| = 9$
 - $\mathbb{Z}_9^* = \{1, 2, 4, 5, 7, 8\} \Rightarrow |\mathbb{Z}_9^*| = \phi(9) = 6$
- Die Ordnung $ord(a)$ eines Elements $a \in < G, \circ >$ ist die kleinste positive ganze Zahl k mit $a^k = a \circ a \circ a \dots \circ a = 1$.
- Eine Gruppe G ist zyklisch, wenn die Gruppe G ein Element α mit $ord(\alpha) = |G|$ enthält. α ist ein Generator oder primitives Element von G .
 - Beispiel: Das Element $\alpha^i = a = 2$ ist ein Generator für \mathbb{Z}_{11}^*

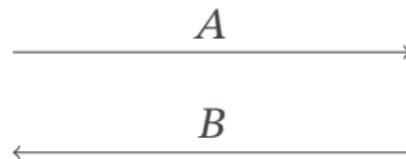
$$\begin{array}{llll}
 a^1 \equiv 2 \pmod{11} & a^2 \equiv 4 \pmod{11} & a^3 \equiv 8 \pmod{11} & a^4 \equiv 5 \pmod{11}, \\
 a^5 \equiv 10 \pmod{11} & a^6 \equiv 9 \pmod{11} & a^7 \equiv 7 \pmod{11} & a^8 \equiv 3 \pmod{11}, \\
 a^9 \equiv 6 \pmod{11} & a^{10} \equiv 1 \pmod{11} & &
 \end{array}$$

DIFFIE-HELLMAN PROTOKOLL

Alice

Wählt a Berechne $A \equiv g^a \pmod{p}$

Bob

Wählt b Berechne $B \equiv g^b \pmod{p}$ Berechne $K \equiv$ $B^a \pmod{p} \equiv g^{b \cdot a} \pmod{p}$ Berechne $K \equiv$ $A^b \pmod{p} \equiv g^{a \cdot b} \pmod{p}$

K kann für sym. Verschlüsselung genutzt werden

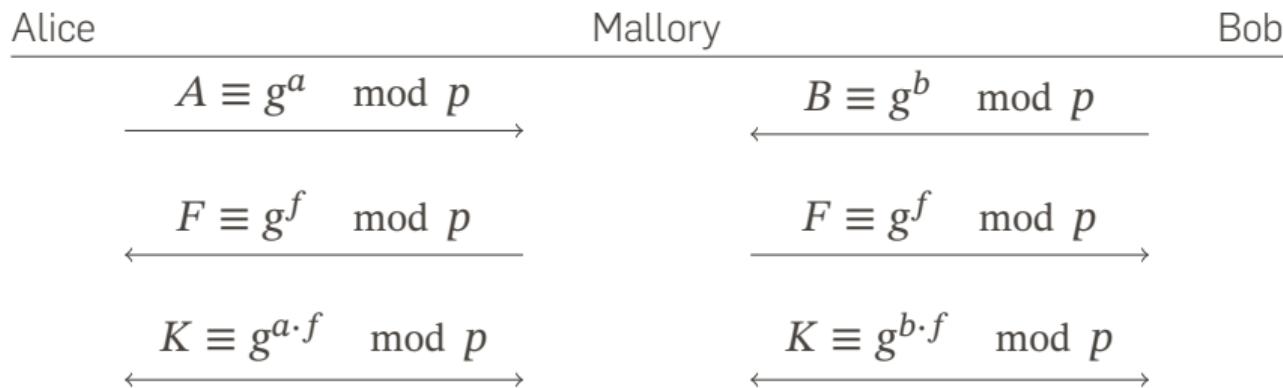


WIESO IST DH SICHER?

- Eve möchte den Schlüssel K berechnen
 - Öffentlich: $g, p, A \equiv g^a \pmod{p}, B \equiv g^b \pmod{p}$
 - Geheim: a, b und $K = p^{a \cdot b}$
- Um a (oder b) zu finden, muss Eve den diskreten Logarithmus berechnen:
 - $a \log_g A \pmod{p}$ oder
 - $b \log_g B \pmod{p}$
- Aber: Bester bekannter Algorithmus zur Berechnung des diskreten Logarithmus hat Komplexität $\mathcal{O}\left(2^{\frac{n}{2}}\right)$ für n -bit p (vereinfacht!).

DH SCHLÜSSEL BEHALTEN ODER LÖSCHEN?

- Originales DH Protokoll: a und b werden für jeden Austausch neu generiert
 - **Vorteil:** Falls a oder b einer Sitzung veröffentlicht werden, ist nur die aktuelle Sitzung korrumptiert (sog. **Forward Secrecy**) ⇒ Standard in vielen Protokollen
 - **Nachteil:** Mallory kann Schlüsselaustausch abfangen, da Alice und Bob sich nicht anhand von A und V authentifizieren können (sog. **Man-in-the-Angriff**)



ZUSAMMENFASSUNG

- Verschlüsselungsalgorithmus AES
- Verwendungszweck von Betriebsmodi für Blockchiffren
- Passende Betriebsmodi für einen einfachen Anwendungsfall auswählen
- Unterschied zwischen öffentlichem und privatem Schlüssel
- Verschlüsselungsalgorithmus RSA
- Vor- und Nachteile von symmetrischen- und asymmetrischen Verfahren
- Aufbau und Vorteile von hybriden Verschlüsselungsverfahren
- DH Verfahren sowie Vor- und Nachteile des Behaltens der öffentlichen Schlüssel



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Integrität

May 26, 2023

Marc Stöttinger



Without integrity, encryption is meaningless.

Bruce Schneier

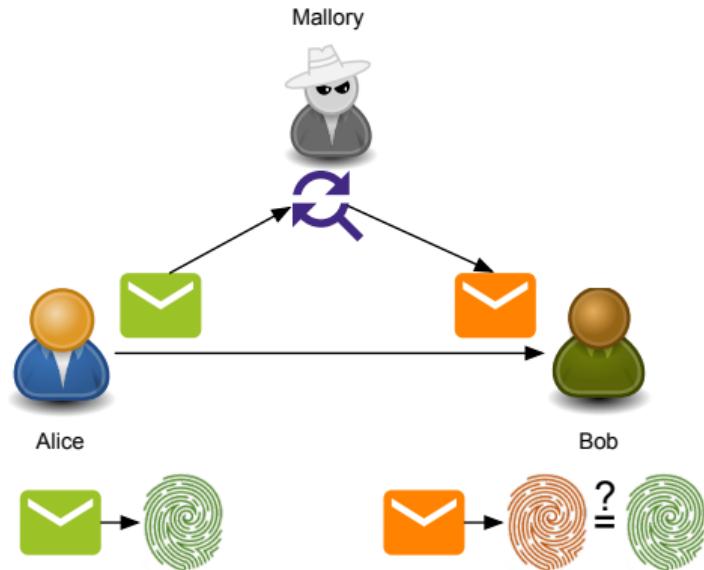
MOTIVATION INTEGRITÄT

→ **Bedrohung:**

Mallory verändert die Nachricht

→ **Ziel:**

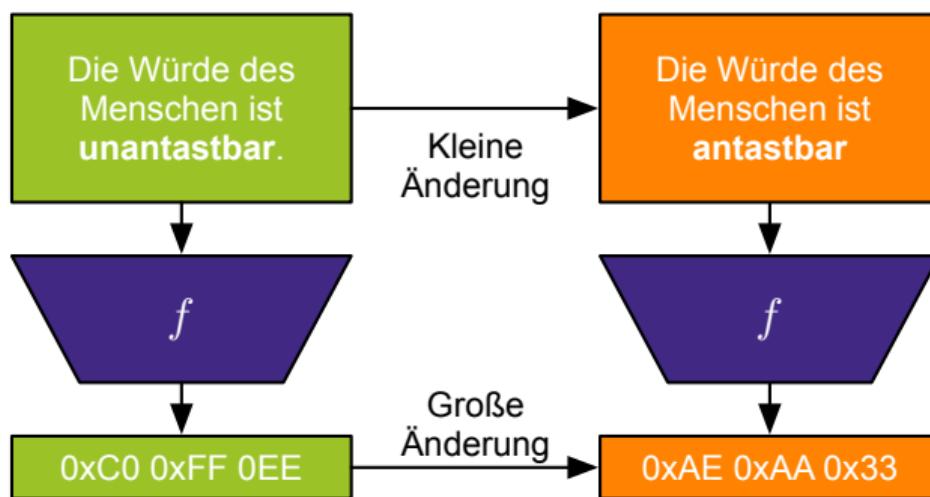
Eindeutiger Fingerabdruck mit dem
unerlaubte Änderungen an der
Nachricht erkannt werden können



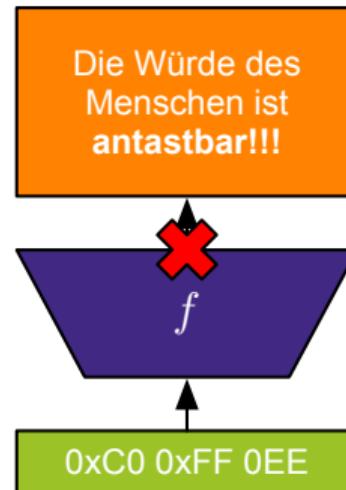
INTEGRITÄT DURCH EINWEGFUNKTIONEN

Eine Einwegfunktion f bildet einen **beliebig langen** Wert auf einen **nicht-invertierbaren** Wert fixer Länge ab

Änderungen in Eingaben von Einwegfunktionen



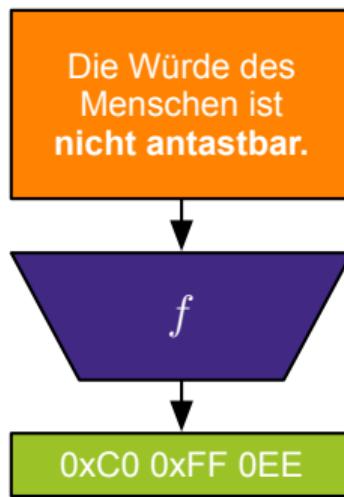
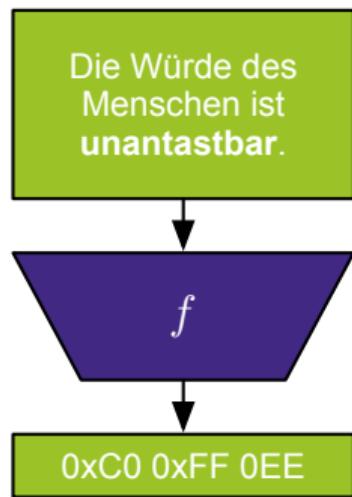
Einwegeigenschaft



KOLLISIONSRESISTENZ VON EINWEGFUNKTIONEN

Kollisionen nicht vermeidbar, da die Länge der Nachricht reduziert wird

→ Möglicher Angriff: Ausprobieren von Nachrichten bis eine Kollision gefunden wird



Hashfunktionen
Hashfunktionen erweitern die Einwegfunktionen um Kollisionsresistenz

EIGENSCHAFTEN VON HASHFUNKTIONEN H

Eine Hashfunktion H bildet eine **beliebig lange Nachricht m** auf einen **Hashwert (Digest) fixer Länge $H(m)$** und besitzt die folgenden Eigenschaften:

1. Einwegeigenschaft: (Preimage resistance)

- Die Funktion $H(m)$ muss effizient berechenbar sein
- Es darf nicht möglich sein, die Funktion H zu invertieren, d.h. vom Hashwert auf ein Urbild m zu schließen

2. Schwache Kollisionsresistenz: (second pre-image resistance)

- Es darf nicht möglich sein, zu m ein anderes m' zu finden mit $m \neq m'$ und $H(m) = H(m')$

3. Starke Kollisionsresistenz: (collision resistance)

- Es darf nicht möglich sein, zwei beliebige m und m' zu finden mit $m \neq m'$ und $H(m) = H(m')$

SCHWACHE VS. STARKE KOLLISIONSRESISTENZ GEBURTSTAGSPARADOX (1/2)

→ **Beispiel:** Geburtstag am gleichen Tag im Jahr

→ **Schwache Kollisionsresistenz:**

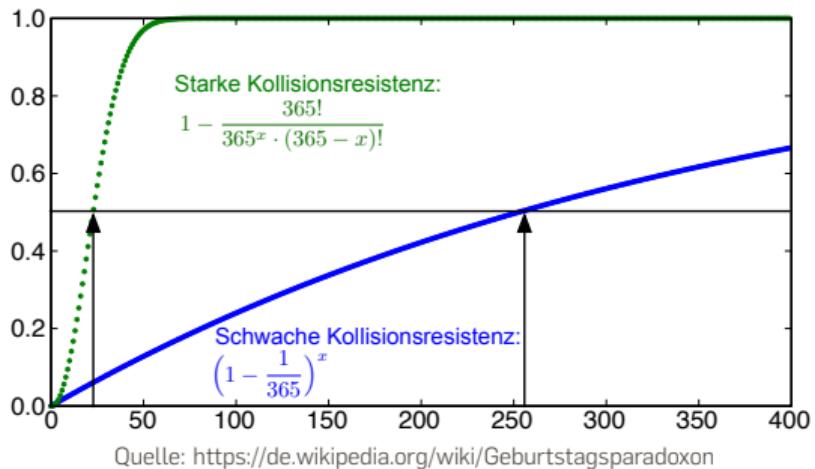
- Wie viele Personen müssen im Raum sein, damit mit $\geq 50\%$ Wahrscheinlichkeit eine Person am gleichen Tag **wie Sie** Geburtstag hat?

→ **Starke Kollisionsresistenz:**

- Wie viele Personen müssen im Raum sein, damit mit $\geq 50\%$ Wahrscheinlichkeit **zwei beliebige** Personen im Raum am gleichen Tag Geburtstag haben?

SCHWACHE VS. STARKE KOLLISIONSRESISTENZ GEBURTSTAGSPARADOX (2/2)

- Schwache Kollisionsresistenz
(Bestimmter Tag): 253
- Starke Kollisionsresistenz
(Beliebiger Tag): 26



WAHL DER HASHWERTLÄNGE

- Wie lang muss ein Hashwert sein, um starke Kollisionsresistenz zu besitzen?
- **Wurzel** als obere Schranke der starken Kollisionsresistenz
 - Bei Elementen mit x -bit Länge sind bei einer Menge von $\sqrt{2^x} = 2^{\frac{x}{2}}$ zufällig gewählten Werten mit $\leq 50\%$ Wahrscheinlichkeit zwei Werte gleich
 - **Beispiel Geburtstagsparadox:** 365 Tage im Jahr, $\sqrt{365} = 19,10 \leq 26$
- Um **x -bit Berechnungssicherheit** zu erhalten, muss die **Ausgabelänge** einer **Hashfunktion $2x$ -bit** sein:
 - 128-bit Sicherheit: 256-bit Ausgabelänge Hashfunktion
 - 256-bit Sicherheit: 512-bit Ausgabelänge Hashfunktion

ÜBERSICHT VON HASH ALGORITHMEN

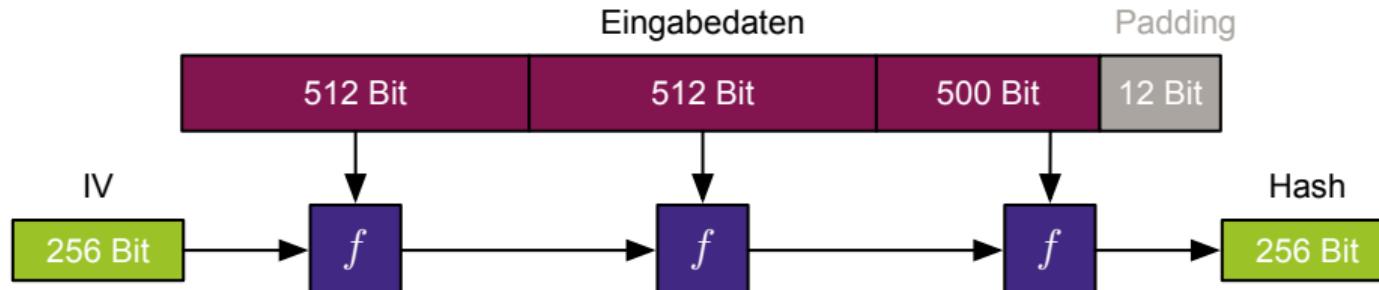
- Basierend auf den bisher diskutierten Anforderungen von Hash-Funktionen gibt es zwei generelle Typen von Hashfunktionen:
 - Dedizierte Hashfunktionen
 - Hashfunktionen basierend auf Blockchiffren
- Um von einer beliebigen Eingabelänge auf eine fixe Ausgabe zu kommen, haben sich folgend Konstruktionsprinzipien etabliert:
 - Merkle-Damgård (Kollisionaresitenz: Hälften der Ausgabelänge)
 - Sponge (Kollisionaresitenz: Minimum (Hälften der Ausgabelänge, Hälften der Kapazität))
- Die Konstruktion einer Hashfunktion beruht auf der speziellen Kompressionsfunktion und einem der beiden Konstruktionsprinzipien

HASHFUNKTIONEN UND SICHERHEIT

Hashfunktion	Digest Länge [Bits]	Sicherheit
MD5	128	Schwache Kollisionsresistenz theoretisch gebrochen (2^{123} [SA09]) Starke Kollisionsresistenz praktisch gebrochen (~35 Minuten Berechnung)
RIPEMD	128/160/256/320	RIPEMD-128 Starke Kollisionsresistenz praktisch gebrochen RIPEMD-160/256/320 Sicher (Angriffe gegen Versionen mit reduzierten Runden)
SHA1	160	Schwache Kollisionsresistenz theoretisch gebrochen ($2^{159,3}$ [KK12]) Starke Kollisionsresistenz praktisch gebrochen (110 GPU Jahre Berechnung)
SHA2	224/256/384/512	Angriffe gegen Versionen mit reduzierten Runden
SHA3	224/256/384/512	Angriffe gegen Versionen mit reduzierten Runden

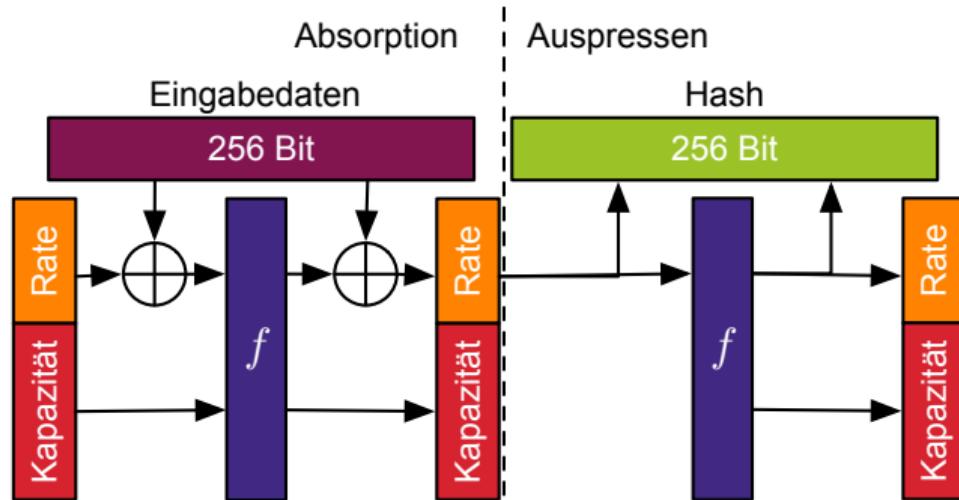
STANDARD HASH ALGORITHM 2 (SHA2)

- Standardisiert vom US NIST im Jahr 2002 als Nachfolger des SHA-1
- Kommt in den Varianten SHA2-224, SHA2-256, SHA2-384 und SHA2-512 und basiert auf einer Merkle-Damgård Konstruktion
 - Zahl am Ende ist die Bitlänge des Hashwertes
 - SHA224/256 bzw. SHA384/512 nutzen die gleiche Funktion mit unterschiedlicher Ausgabelänge
- SHA2-256 nutzt eine Kompressionsfunktion f , die 512-Bit Eingabedatenblöcke mit einem 256-Bit internen Zwischenstand verarbeitet:



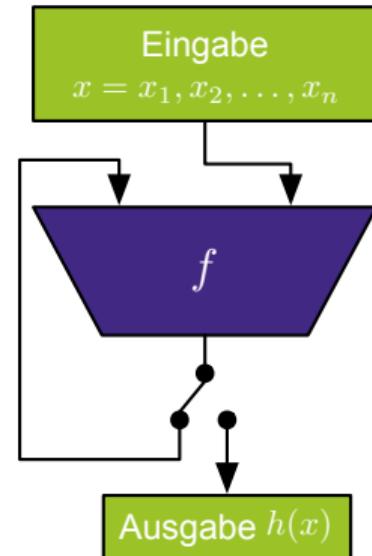
STANDARD HASH ALGORITHM 3 (SHA3)

- Standardisiert vom US NIST im Jahr 2015 als Nachfolger des SHA2 und basiert auf einer Spong-Konstruktion
- Varianten SHA3-224, SHA3-256, SHA3-384, SHA3-512 und SHAKE128, SHAKE256
- mit der Variante SHAKE können beliebige Länge Digest generiert werden



HASHFUNKTIONEN BASIEREND AUF BLOCKCHIFFREN BETRIEBSMODUS

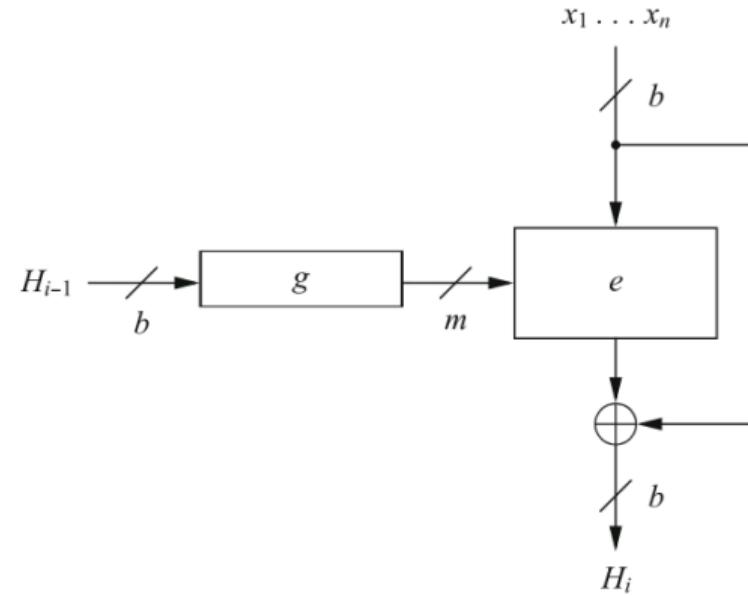
- Hashfunktionen, welche Blockchiffren als Permutationsfunktion benutzen, verwenden eine Merkle-Damgård-Konstruktion
- Die Kollisionsresistenz entspricht in den meisten Fällen der Hälfte der Blockgröße
- Im Fall von AES-128/192/256 ist die Kollisionsresistenz der Konstruktion immer 64-Bit wegen der Blockgröße.



MATYAS-MEYER-OSEAS HASH

- Die Rückkopplung des vorherigen Hashwerts H_{i-1} geschieht über den Schlüsseleingang, die Läge muss über die Abbildungsfunktion g angepasst werden
- Funktionsvorschrift:

$$H_i = \text{Enc}_{g(H_{i-1})}(x_i) \oplus x_i$$
- b entspricht der Blockgröße der Chiffre und m der Schlüssellänge

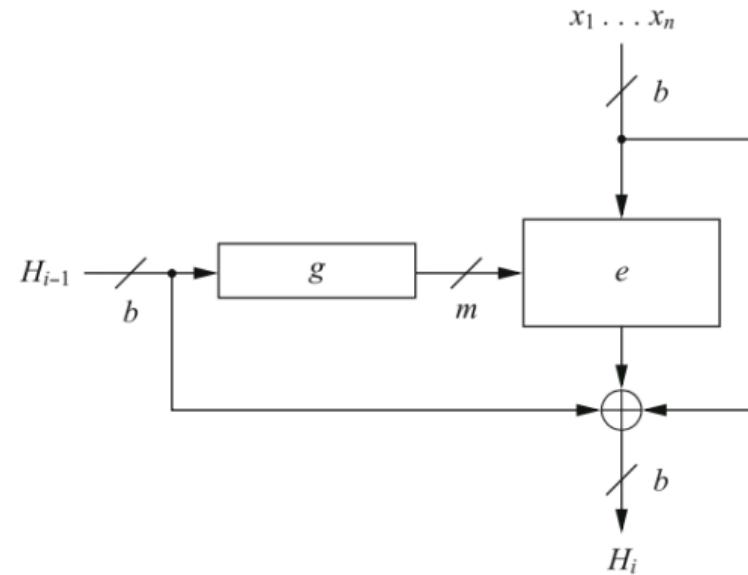


Quelle: Christoph Paar, Jan Pelz: Kryptografie verständlich, 2016, Springer

MIYAGUCHI-PRENEEL HASH

- Die Miyaguchi-Preneel Hash-Konstruktion verknüpft noch die Message mit Digest als Ergänzung zur Matyas-Meyer-Oseas Hash-Konstruktion
- Funktionsvorschrift:

$$H_i = \text{Enc}_{g(H_{i-1})}(x_i) \oplus x_i \oplus H_{i-1}$$
- b entspricht der Blockgröße der Chiffre und m der Schlüssellänge



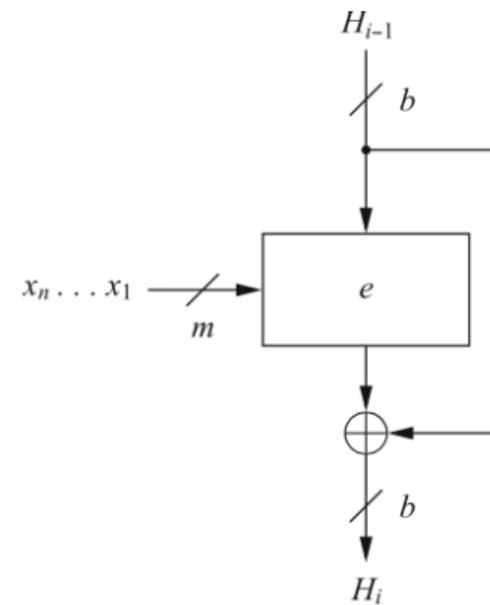
Quelle: Christoph Paar, Jan Pelz: Kryptografie verständlich, 2016, Springer

DAVIS-MEYER HASH

- Die Davis-Meyer Hash-Konstruktion nutzt den Schlüsseleingang der Chiffre für die Texteingabe
 - Rückkopplung des vorherigen Digest über Plaintexteingang der Chiffre
 - Keine Abbildungsfunktion g wird benötigt.

- Funktionsvorschrift:

$$H_i = \text{Enc}_{x_i}(H_{i-1}) \oplus H_{i-1}$$



Quelle: Christoph Paar, Jan Pelz: Kryptografie verständlich, 2016, Springer

ZUSAMMENFASSUNG

- Einsatzzwecke und Eigenschaften von kryptographischen Hashfunktionen
- Unterschied zwischen schwacher und starker Kollisionsresistenz
- Existierende kryptographische Hashfunktionen kennen (dedizierte und auf Blockchiffren basierende Hashfunktionen)
- Beurteilung ob eine Funktion die Eigenschaften der kryptographischer Hashfunktionen erfüllt



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Zufallszahlen und -generatoren

May 26, 2023

Marc Stöttinger



Random numbers should not be generated with a method chosen at random.

Donald Knuth

ZUFALLSZAHLENGENERIERUNG

- Kryptographische Verfahren benötigen sichere Zufallszahlen für
 - Wahl symmetrischer Schlüssel
 - Initialisierungsvektoren (IVs) für Betriebsmodi von Blockchiffren
 - Primzahl- und Parametergenerierung bei RSA, DSA, DH, ...
- Aber was sind „sichere“ Zufallszahlen?
 - Welche Eigenschaften müssen „sichere“ Zufallszahlen haben?
 - Wie sieht ein „sicherer“ Zufallszahlengenerator aus?

DISKUSSION IN KLEINEN GRUPPEN

Welche der folgenden Zufallszahlenfolgen sind „sicher“?

- 42 42 42 42 42 42 42 42 42 42
- 1 6 11 16 21 26 31 36 41 46 51
- 3141 5926 5358 9793 2384

Eigenschaften von kryptographischen Zufallszahlen

Welche Eigenschaften machen „sichere“ Zufallszahlen aus?

BEISPIELE FÜR UNSICHERE ZUFALLSZAHLENGENERATOREN

Welche der folgenden Zufallszahlenfolgen sind „sicher“?

- 42 42 42 42 42 42 42 42 42 42 → Kein Zufall
- 1 6 11 16 21 26 31 36 41 46 51 → Nicht gleichverteilt
- 3141 5926 5358 9793 2384 → Vorhersagbar! Stellen von Pi sind bekannt

Eigenschaften von kryptographischen Zufallszahlen

Welche Eigenschaften machen „sichere“ Zufallszahlen aus?
Gleichverteilung, Nichtvorhersagbarkeit

WAS FÜR EIGENSCHAFTEN HABEN ZUFALLSZAHLEN?

Ideale Zufallszahlen sind **nichtvorhersehbar**, **unabhängig** und **gleichverteilt** (ideale Zufälligkeit).

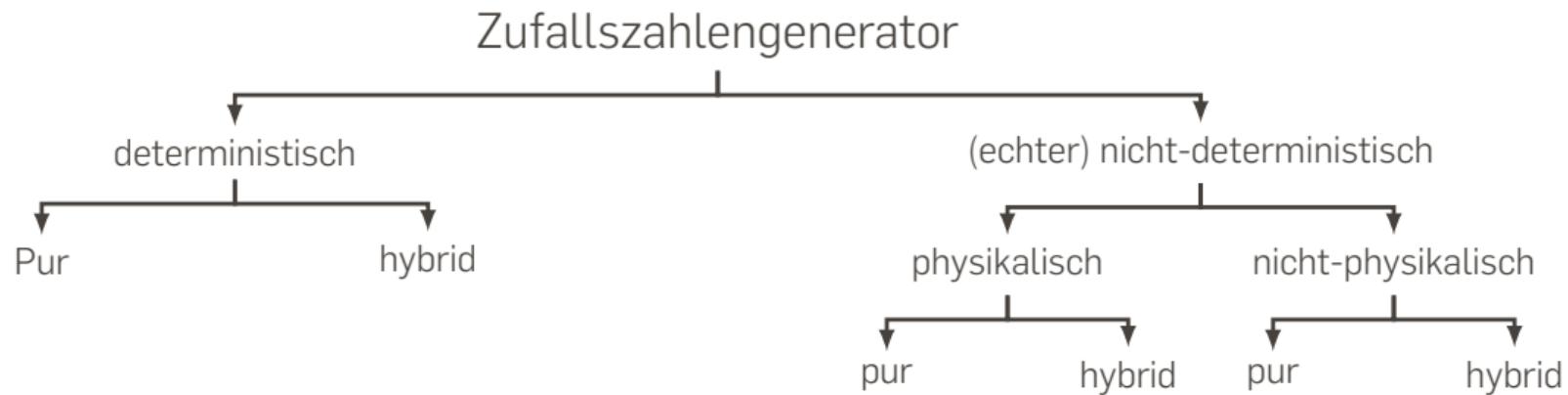
- Nichtvorhersehbarkeit: Die Zufallszahl ist vor der Generierung mit einem gewissen Grad der Ungewissheit nicht vorhersehbar. Der Grad der Ungewissheit kann durch die Entropie quantifiziert werden.
- Unabhängigkeit: Die Erzeugung früherer Zufallszahlen darf keinen nachvollziehbaren Einfluss auf die aktuelle Erzeugung einer Zufallszahl haben.
- Gleichverteilung: Jeder zulässige Wert einer Zufallszahl hat die gleiche Chance, einzutreten.

Entropie

Eine gute Entropiequelle ist essentiell für einen Zufallszahlengenerator.

KRYPTOGRAPHISCHER ZUFALLSZAHLGENERATOREN

- Ein RNG besteht aus einem **nicht-deterministischen** Teil (Entropiequelle) und einem **deterministischen Teil**, der aus diesen Daten die Ausgangssequenz des RNG (Zufallszahlen) erzeugt.
- Der nicht-deterministische Teil des RNG nutzt eine **physikalische Entropiequelle** oder **nicht-physikalische Entropiequelle** zur Erzeugung einer Zufallszahlenfolge.



ECHTE ZUFALLSZAHLENGENERATOREN (PTRNG)

- Der Kern eines jeden **Physical True Random Number Generator** (PTRNG) ist die **Entropiequelle** → **raw** Zufallszahlen.
 - Durch Ausnutzung eines analogen Signals erzeugt ein **Digitalisierungsmechanismus** eine Folge von **digitalen "rohen" Daten**
 - Ein **Nachbearbeitungsalgorithmus** wandelt die Rohdaten in **interne Zufallszahlen** um.
- **Zeitdiskrete** physikalische Entropiequelle ist z.B.,
 - Radioaktiver atomarer Zerfall
- **Analog** Physikalische Entropiequellen sind z.B.,
 - Thermische Widerstandsentropie: Die Spannung zwischen Widerständen schwankt zufällig aufgrund der Vibration von Atomen.
 - Diodendurchbruch-Entropie: Der Sperrstrom durch Dioden variiert zufällig aufgrund des Tunnelns von Elektronen.

BEISPIEL FÜR ECHTE ZUFALLSZAHLENGENERATOREN (TRNG)

- Basieren auf **nichtvorhersehbaren, physikalischen** Prozessen
 - von bewegten Klumpen in Lavalampen
 - Sperrstrom durch Dioden schwankt zufällig aufgrund von Tunneln von Elektronen
- Statistische Verteilung des Rauschens ist **häufig suboptimal**
 - Lösung: TRNG aggregiert Messungen, um gute Verteilung zu erhalten
 - Ausgaberate des TRNGs entsprechend niedrig (~KB/sec)



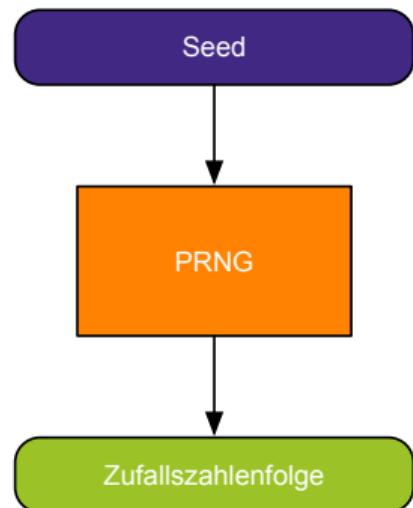
Quelle: <https://www.cloudflare.com/de-de/learning/ssl/lava-lamp-encryption/>

ECHTE ZUFALLSZAHLENGENERATOREN (NPTRNG)

- Ein **nichtphysikalischer echter Zufallszahlengenerator** (NPTRNG) verwendet **externe Signale** als Entropiequelle.
- Das Konzept der Zufälligkeit ist der **Mangel an Information** über Prozesse und ihre Ergebnisse.
- Externe Entropiequellen sind z.B.,
 - **Prozesse** wie Platten-I/O-Operationen und Interrupts (z.B. Linux RNG **/dev/random**).
 - **Systemdaten** als Tickzähler seit Systemstart, Prozess- und Thread-IDs, und aktuelle Ortszeit (z.B. in MS Windows CE Enhanced Cryptographic Provider).
 - **Menschliche Interaktion** als Mausbewegung und Tastenanschläge (z.B. PGP-Schlüsselerzeugung).
- Eine große Menge von Daten aus verschiedenen Quellen und Nachbearbeitung (z.B. durch eine Hash-Funktion) sind erforderlich.

PSEUDO ZUFALLSZAHLENGENERATOREN (PRNG)

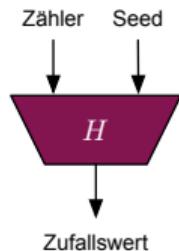
- Algorithmus, der aus einem Startwert (Seed) eine Zufallszahlenfolge erzeugt
 - Deterministisch: Gleicher Seed → Gleiche Zufallszahlenfolge
- Sicherheit eines PRNG hängt ab von
 - Zufälligkeit und Nichtvorhersagbarkeit des Seeds
 - Verwendetem Algorithmus
- Brute-Force auf Seed ist möglich, da PRNG deterministisch
 - Mindestens 2^{128} Möglichkeiten für Seed (für Rechnerische Sicherheit)
- Empfohlen Standard für Anforderungen an den Seed [SP800-90ARev1] und [SP800-90C3pd]



EIGENSCHAFTEN EINER PRNG FUNKTION

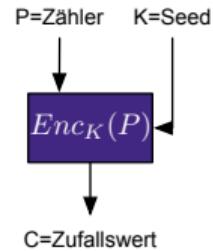
→ Sicherheitseigenschaften einer PRNG

1. Vom Zufallswert darf nicht auf den Seed geschlossen werden
→ Ansonsten: Berechnung des Seeds und Bruch der Nichtvorhersagbarkeit
2. Der Zufallswert muss gut verteilt sein
→ Ansonsten: Ausnutzung einer schlechten Verteilung



Sicherheitseigenschaften:

1. Einwegfunktion
2. Starke Kollisionsresistenz



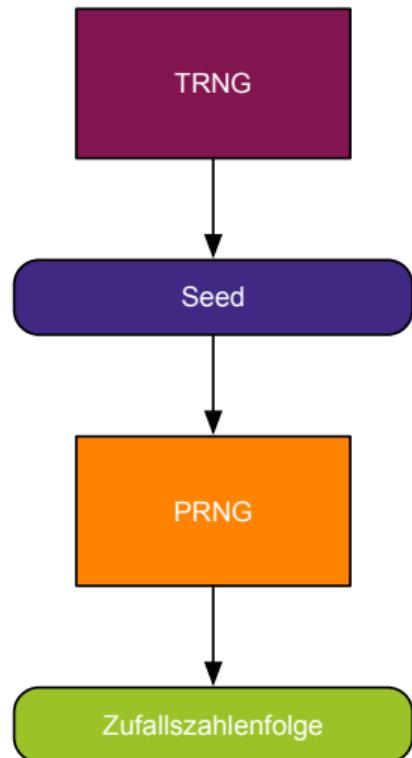
Sicherheitseigenschaften:

1. Seed nicht erreichbar aus C und P
2. eindeutige Zuordnung durch $C = Enc_K(P)$

- Empfohlener Standard für Konstruktion von PRNGs [SP800-90ARev1]

HYPRIDE ZUFALLSZAHLENGENERATOREN

- Kurzer Seed wird aus TRNG generiert
 - Umgeht Problem der langsamen TRNG Geschwindigkeit
- Seed dient als Eingabe zum PRNG und wird dort beliebig erweitert
- Kombiniert die Vorteile beider Verfahren
 - Sicherheit basiert auf echtem Zufall
 - Effizienz von PRNG wird genutzt
- Empfohler Standard für Konstruktion [AIS].



PROBLEME BEI ZUFALLSZAHLENGENERATOREN

- **Grundproblem:** Nichtvorhersagbarkeit und Gleichverteilung nur schwer prüfbar
- Häufige Sicherheitsprobleme bei Zufallszahlengeneratoren
 - **Konstanter Wert** oder **Uhrzeit** als Seed für PRNG (z.B: MiFare Chips [MIF])
 - **TRNG** hat **schlechte Verteilung** (z.B.: gemeinsame Primzahl in RSA)
 - **PRNG Eigenkonstruktionen** mit schlechtem Design (z.B.: Windows PRNG [DGP07])
 - **Implementierungsfehler** in PRNG (z.B.: Android Java PRNG [MMS13])
- Möglichkeiten zur Überprüfung
 - Statistische Tests für Gleichverteilung (DieHarder Testsuite, TestU01, ...)
 - BSI Standardisierungen für TRNGs/PRNGs (AIS 20 und 31 Standards [AIS])

ZUSAMMENFASSUNG

- Eigenschaften kryptographischer Zufallszahlengeneratoren
- Eigenschaften und Unterschiede von TRNGs, PRNGs und hybriden RNGs
- Beurteilen ob gewählte Funktionen die Eigenschaften von kryptographischer Zufallszahlengeneratoren erfüllt



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Authentizität und Verbindlichkeit

June 3, 2023

Marc Stöttinger



Authenticity is the bedrock of cryptography, for without it, even the strongest encryption is worthless.

Bruce Schneier

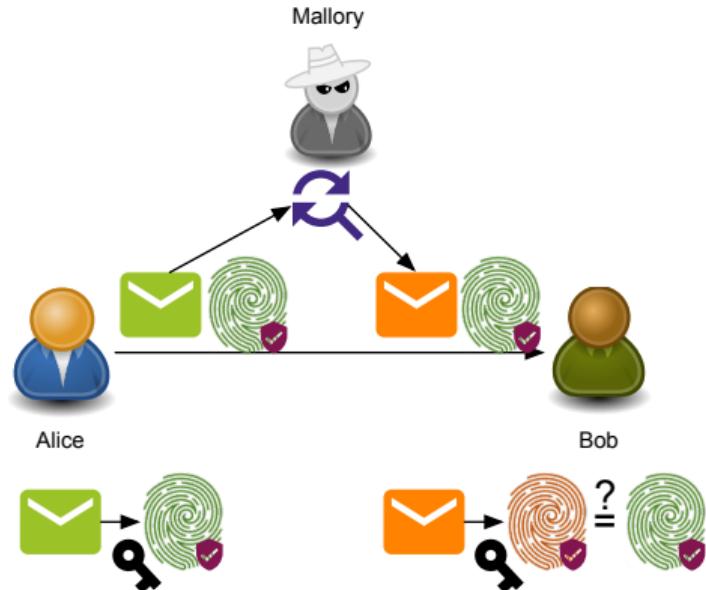
INTEGRITÄT UND AUTHENTIZITÄT DURCH MESSAGE AUTHENTICATION CODES

→ **Bedrohung:**

- Integrität: Mallory ändert die Nachricht
- Authentizität: Mallory fälscht eine Nachricht und gibt sich als Alice aus

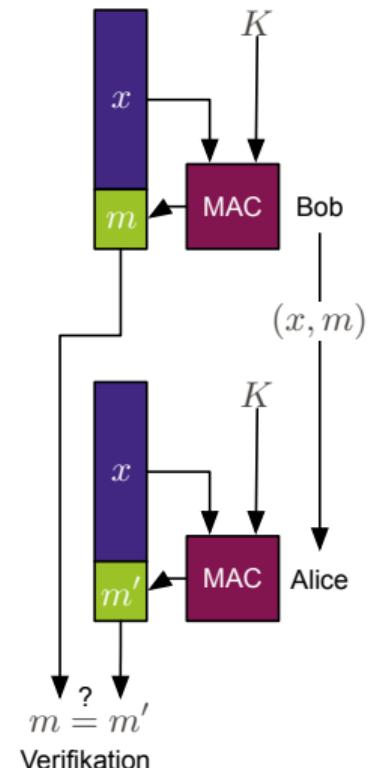
→ **Ziele:**

- Integrität: Bob kann prüfen, ob die Nachricht verändert wurde
- Authentizität: Bob kann prüfen, ob die Nachricht von Alice stammt



WIESO REICHT INTEGRITÄT NICHT AUS?

- Integrität garantiert nur, dass Veränderungen erkannt werden, z.B Fehler bei der Übertragung.
 - Mit einer Prüfsumme kann jeder den Ciphertext auf Fehler überprüfen.
- Es ist keine Aussage über eine Verfälschung möglich
 - Die Prüfsumme kann jeder selber berechnen
 - Bei einem verschlüsselten Ciphertext ist nicht bekannt, woher der Ciphertext stammt oder ob dieser authentisch ist.
- Es wird ein Mechanismus benötigt, dass die Prüfsumme nur mit Kenntnis eines Geheimnisses berechnet werden kann.

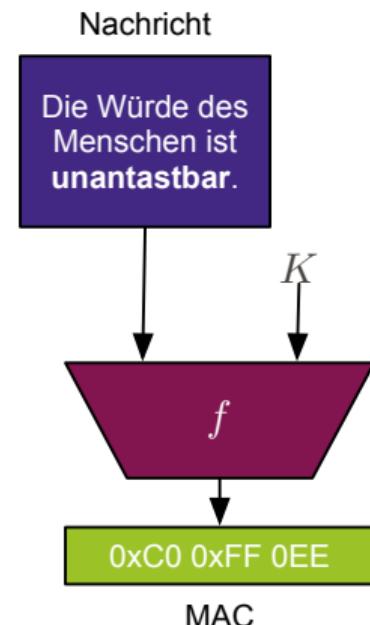


MESSAGE AUTHENTICATION CODES (MACS)

→ **Message Authentication Code (MAC)** nutzen

Hashfunktionen und symmetrische Verschlüsselungsverfahren zusammen mit symmetrischen Schlüsseln, um Integrität und Authentizität zu gewährleisten

- Schnell, da symm. Verschlüsselung und Hashfunktionen genutzt werden
 - Unverbindlich, da der Schlüssel beiden Parteien bekannt ist
-
- Benötigte Eigenschaften eines MAC Verfahrens:
- **Einwegeigenschaft**: Der Schlüssel darf nicht aus MAC und Nachricht bestimbar sein
 - **Schwache Kollisionsresistenz**: Keine andere Nachricht mit gleichem Schlüssel und gleicher MAC darf effizient berechenbar sein



BEKANNTEN MAC-ALGORITHMEN

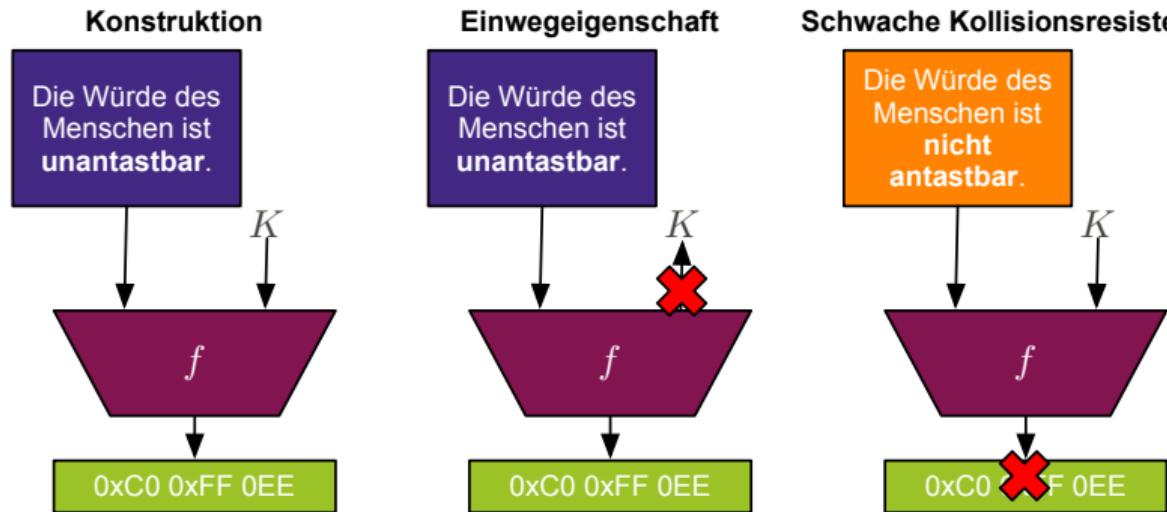
- Es existieren spezielle MAC-Konstruktionen, um ein MAC-Verfahren aus einer Hashfunktion oder einem symmetrischen Verschlüsselungsverfahren zu konstruieren

Verfahren	Schlüssellänge	Kommentar
Hash-based MAC (HMAC)	Blockgröße der Hashfunktion (z.B. 256-bit bei SHA2-256)	Gute Sicherheit, da starke Kollisionsresistenz, allerdings vergleichsweise langsam
Blockchiffre Modi GMAC und CMAC	Schlüssellänge der Blockchiffre (z.B. 128 bei AES-128)	In bestimmten Fällen geringe Kollisionsresistenz
Blockchiffre Modus GCM	Schlüssellänge der Blockchiffre (z.B. 128 bei AES-128)	In bestimmten Fällen geringe Kollisionsresistenz, bietet aber zusätzlich Verschlüsselung an

MESSAGE AUTHENTICATION CODES (MACS) SICHERHEIT HASHFUNKTIONEN

→ MACs via Hashfunktionen sind sicher, da:

- Einweg: Schlüssel kann nicht aus MAC und Nachricht berechnet werden
- Schwache Koll.: Andere Nachricht mit gleichem MAC und Schlüssel schwer findbar
- Je nach verwendeter Hashfunktion ist eine Maskierung der Digest notwendig

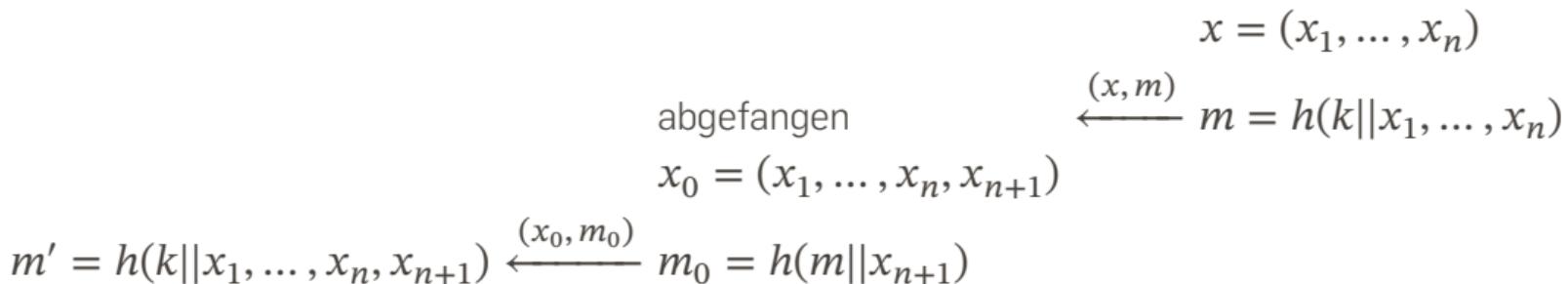


ANGRIFF GEGEN HMACS MIT VORANGESTELLTEM GEHEIMNIS

Alice

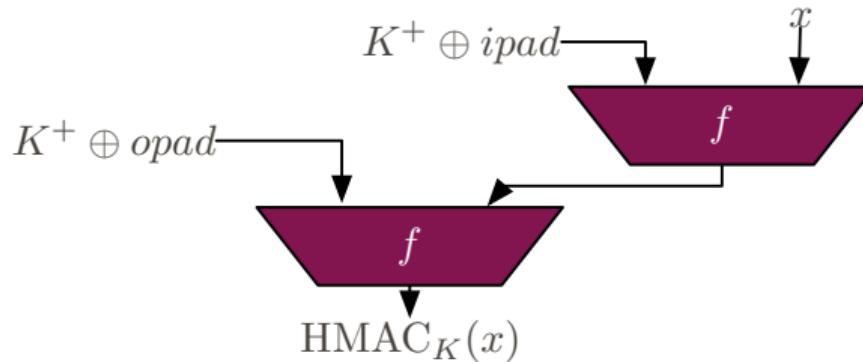
Mallory

Bob

**Valide MAC, da $m' = m_0$**

HASH-BASED MAC

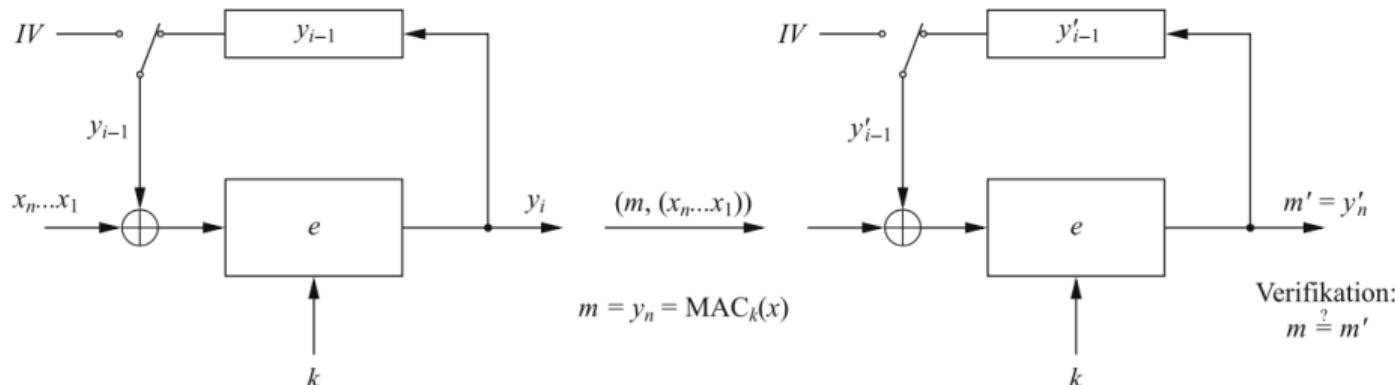
- Die [HMAC] Konstruktion sollte immer eine Padding-Mask nutzen, um die Hashfunktion $f(x)$ in einen HMAC zu wandeln: $HMAC_K(x) = h[(K^+ \oplus opad)|h[(K^+ \oplus ipad)|x]]$
- Konstanten $ipad$ und $opad$ werden als Padding-Masken genutzt.
- K wird auf die Blockgröße des Hash aufgefüllt und K^+ bezeichnet



MESSAGE AUTHENTICATION CODES (MACS) SICHERHEIT SYM. VERSCHLÜSSELUNG

→ MACs via symmetrischer Verschlüsselungsverfahren sind sicher, da:

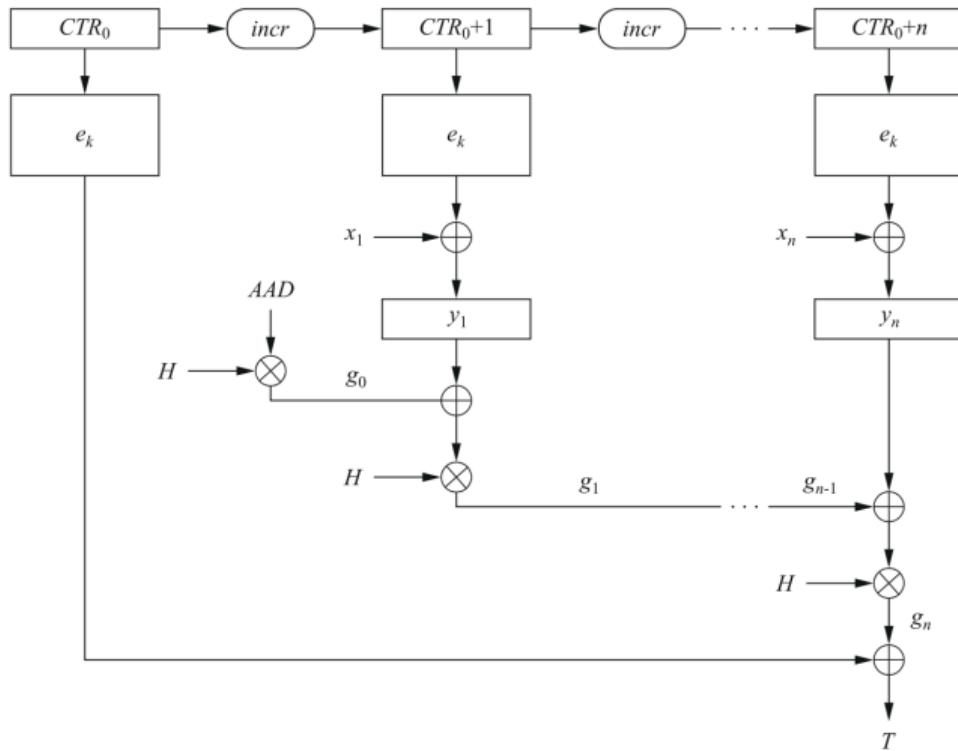
- **Einweg**: Schlüssel kann nicht aus Plaintext $P = x_1, \dots, x_n$ und Ciphertext $C = m$ berechnet werden
- **Schwache Koll.**: Jeder Plaintext P wird in einen zufälligen und eindeutigen Ciphertext C unter fixem Schlüssel k verschlüsselt.



Quelle: Christoph Paar, Jan Pelz: Kryptografie verständlich, 2016, Springer

BLOCKCHIFFRE BETRIEBSMODI GALOIS COUNTER MODE (GCM)

- Konstruktion um Blockchiffre in MAC-Verfahren umzuwandeln
- Multiplikationskonstante $H = ENC_k(0)$
- ADD zusätzliche Authentisierungsdaten
- T ist der Authentisierungstoken, der als MAC genutzt werden kann.



Quelle: Christoph Paar, Jan Pelz: Kryptografie verständlich, 2016, Springer Security

DISKUSSION IN KLEINEN GRUPPEN

Sind MACS uneingeschränkt vertrauenswürdig?

Einen Gruppe von Freunden hat einen symmetrischen Schlüssel miteinander getauscht, um sich gegenseitig via AES-GCM sichere und vertrauenswürdige Nachrichten zu schicken. Was für ein Problem könnte in diesem Setting entstehen?

Verbindlichkeit

Überlegen Sie sich einen Fall, in dem das oben identifizierte Problem zum Tragen kommen kann.

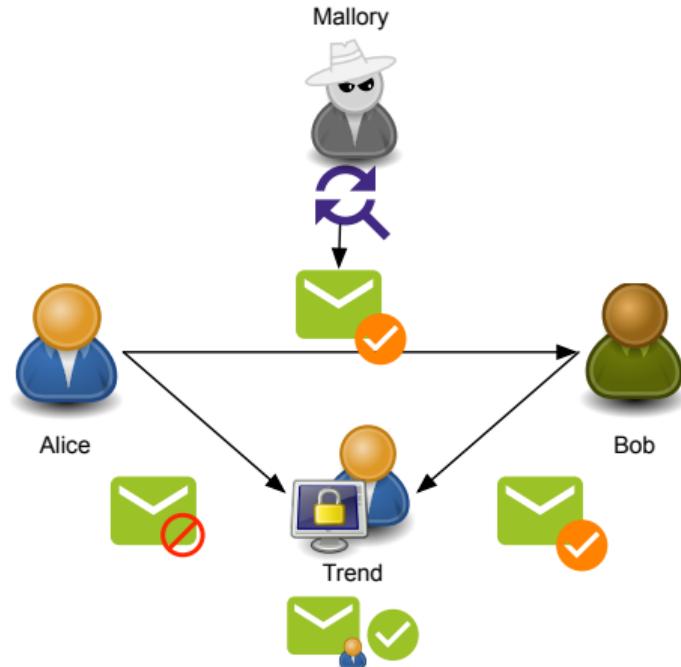
INTEGRITÄT, AUTHENTIZITÄT UND VERBINDLICHKEIT

→ **Bedrohung:**

- Integrität: Mallory ändert die Nachricht
- Authentizität: Mallory fälscht eine Nachricht und gibt sich als Alice aus
- **Nicht-Abstreitbarkeit:** Alice bestreitet eine Nachricht an Bob gesendet zu haben

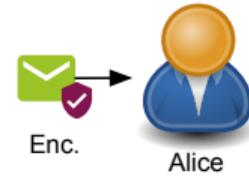
→ **Ziele:**

- Integrität: Bob kann prüfen, ob die Nachricht verändert wurde
- Authentizität: Bob kann prüfen, ob die Nachricht von Alice stammt
- **Nicht-Abstreitbarkeit:** Bob kann gegenüber einer vertrauenswürdigen, dritten Instanz nachweisen, dass eine Nachricht von Alice stammt



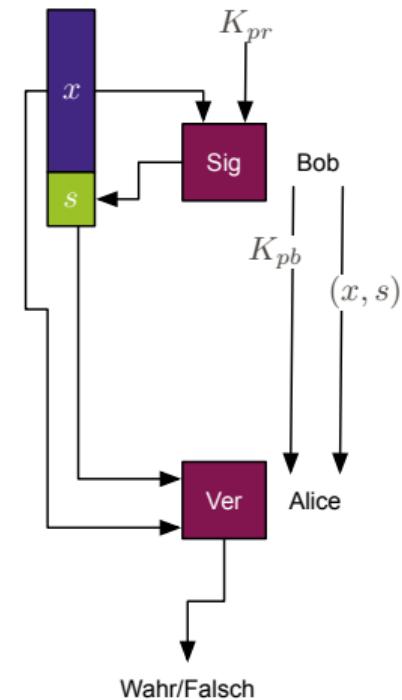
DIGITALE SIGNATUREN

- Digitales Pendant zur handgeschriebenen Unterschrift
 - Zu einer öffentlichen Nachricht m soll es eine digitale Signatur sig geben
- Anforderungen an ein digitales Signaturverfahren:
 1. Jeder muss die Signatur von sig zu m verifizieren können
 2. Nur Alice darf eine gültige Signatur sig zur Nachricht m erzeugen
- Vergleich mit Anforderungen bei Verschlüsselung mit asymmetrischer Kryptographie:
 1. Jeder darf eine Nachricht an Alice verschlüsseln können
 2. Nur Alice darf den Ciphertext entschlüsseln können



GENERELLER ABLAUF

- Bob erstellt als erstes ein Private-Public-Schlüsselpaar (K_{pr} und K_{pb})
 - K_{pr} wird benötigt, um die Signatur zu erzeugen
 - K_{pb} wird an Alice geschickt, damit sie die Signatur verifizieren kann
- Signieren der Nachricht m :
 - Über einen Einweg- oder Falltür-Funktion ($F_{trap}(\cdot)$) wird s erzeugt:
 $F_{trap}(K_{Pr}, m) \rightarrow S$
 - **Die Falltür Funktion ist im allg. keine Verschlüsselung!**
 $F_{trap}(K_{Pr}, m) \rightarrow s \neq ENC_{K_{pr}, m} \rightarrow s$
- Verifikation der Signatur s :
 - Mit einer Verifikationsfunktion wird mit Hilfe von K_{pb} und m geprüft, ob s valide ist



ALICE SENDET VIA RSA SIGNIERTE NACHRICHT AN BOB

Alice

Bob

 $K_{pb} = (N, e)$ und $K_{pr} = d$ Kanal1: Sende $K_{pb} = (N, e)$


Signiere $s = m^d \mod N$ Kanal2: (m, s)

Berechne $m' = s^e \mod N$
Verfizier ob $m = m'$

Signaturprüfung funktioniert, da $s^e \mod N = m^{ed} \mod N = m$

- Jede Partei darf $K_{pb} = (N, e)$ kennen und Signaturen prüfen
- Nur Alice kennt $K_{pr} = (s)$ und kann somit Nachrichten signieren

RECAP: BOB SENDET VIA RSA VERSCHLÜSSELTE NACHRICHT AN ALICE

Alice

$$K_{pb} = (N, e) \text{ und } K_{pr} = d$$

Bob

Kanal1: Sende $K_{pb} = (N, e)$

$$\text{Entschlüssele } P = C^d \mod N \xleftarrow{\text{Kanal2: } (C)}$$

$$\text{Verschlüssel } C = P^e \mod N$$

Verfizier ob $m = m'$

Entschlüsselung funktioniert genauso wie bei der Signatur nur

- Sonderfall für Schulbuch RSA Entschlüsselung entspricht der Signierfunktion!
- Es ist sehr gefährlich, einfach Schulbuch RSA zu benutzen!

EXISTENZIELLE FÄLSCHUNG

Alice

Mallory

Bob

$$\xleftarrow{(N, e)}$$

$$\xleftarrow{(N, e)} K_{pr} = (d), K_{pb} = N, e$$

1. Wähle Signatur: $s \in \mathbb{Z}_N$
2. Berechne die Nachricht:

$$\xleftarrow{(x, s)} m \equiv s^e \pmod{N}$$

Verifikation:

$$m' \equiv s^e \pmod{N} = m$$

Signatur ist valide!

- Probabilistische Signaturverfahren verhindern diesen Angriff.
- Das **RSA-EMSA-PSS** Schema für Signaturen sollte im Fall von RSA genutzt werden

DIGITALE SIGNATURVERFAHREN IN DER PRAXIS

- Weitere Verfahren zur Signaturberechnung existieren:
 - Digital Signature Algorithm (DSA)
 - Elliptic Curve DSA (ECDSA)
 - Elgamal Signatur
 - Merkle Signatur
- Für Verbindlichkeit müssen weitere Informationen an den öffentlichen Schlüssel gebunden werden (⇒**Zertifikate** und **PKI** im Kapitel Protokolle)

DIGITAL SIGNATURE ALGORITHM (DSA) ENTSTEHUNG UND VERWENDUNG

- Digital Signature Algorithm (DSA) wurde 1994 standardisiert [DSA]
 - Von der Benutzung von DSA wird mittlerweile abgeraten!
- Die Sicherheit von DSA beruht auf dem diskreten Logarithmen Problem

DSA Algorithmus	Input	Output	Durchgeführt von
Parametergenerierung	-	Parameter (p, q, g)	Vertrauenswürdige Partei
Schlüsselgenerierung	(p, q, g)	Schlüssel $K_{pb} = y, K_{pr} = x$	Alice (Sender*in)
Signieren	$(p, q, g), x, m$	Signatur (r, s) zu M	Alice (Sender*in)
Signieren	$(p, q, g), x, m, (r, s)$	Wurde (r, s) für M von Besitzer*in von K_{pb} erzeugt?	Bob (Empfänger*in)

DIGITAL SIGNATURE ALGORITHM (DSA) SCHLÜSSELGENERIERUNG

Alice

Bob

Parametergenerierung (öffentlich)

Wähle eine Primzahl p Wähle eine Primzahl q die $p - 1$ teiltBerechne $g \equiv h^{(p-1)/q} \pmod{p}$

Schlüsselgenerierung

 $\xleftarrow{(p, q, g)}$ für ein zufälliges h $\xrightarrow{(p, q, g)}$ Wähle x mit $1 \leq x \leq q$ Berechne $y \equiv g^x \pmod{p}$ Setze $K_{pb} = y$ und $K_{pr} = x$ $\xrightarrow{(K_{pb}) = y}$

DIGITAL SIGNATURE ALGORITHM (DSA) SIGNIEREN UND VERIFIZIEREN

Alice

Bob

Wähle k mit $1 \leq k \leq q$

Berechne: $r \equiv (g^k \mod p) \mod q \neq 0$

Berechne: $s \equiv (k^{-1} \cdot (m + r \cdot x)) \mod p \neq 0$

$\xrightarrow{m, (r, s)}$

Berechne $w \equiv s^{-1} \mod q$

Berechne $u_1 \equiv m \cdot w \mod q$

Berechne $u_2 \equiv r \cdot w \mod q$

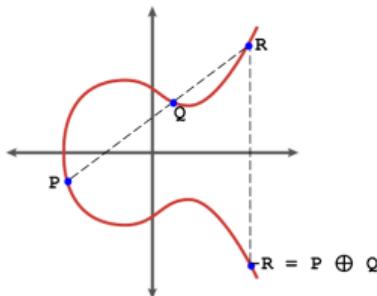
Berechne $v \equiv (g^{u_1} \cdot y^{u_2} \mod p) \mod q$

Falls $v = r \rightarrow$ valide

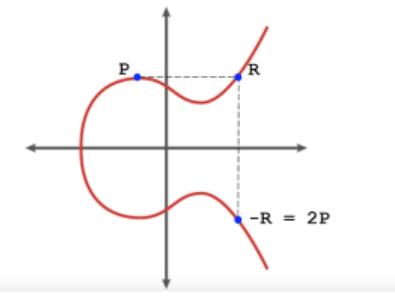
ASYMMETRISCHE VERSCHLÜSSELUNG ELLIPTISCHE KURVEN (1/2)

- Eine Alternative zu primen Restklassenringen sind **elliptische Kurven (ECC)**
 - **Elliptische Kurve:** Menge an Punkten die eine Gleichung erfüllen, z.B.: $y^2 = x^3 + ax + b$
- Seien A, P zwei Punkte auf einer Kurve mit $a \cdot P = A$
 - **Einfach:** Aus P und a den Punkt A zu berechnen ($A = a \cdot P$)
 - **Schwer:** Aus P und A den Wert a zu berechnen ($a = P/A$)

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



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



Quelle: <https://blog.intothesymmetry.com/2019/07/on-isogenies-verifiable-delay-functions.html>

ASYMMETRISCHE VERSCHLÜSSELUNG ELLIPTISCHE KURVEN (2/2)

- Elliptische Kurven können in Verfahren genutzt werden, die auf dem diskreten Logarithmusproblem basieren
- Vorteil von elliptischen Kurven ist, dass die Kurven kleinere Bit-Werte besitzen

Bitlänge Schlüssel	sym.	Bitlänge Primzahl	Bitlänge ECC	Ratio Bitlänge Primzahl / ECC
80		1024	160	6,4
128		3072	256	12
256		15360	512	30

ECDSA SCHLÜSSELGENERIERUNG

Alice

Bob

Schlüsselgenerierung

Wähle einen Zufallswert d mit $0 < d < q$

Berechne $B = dA$

Setze $K_{pb} = (p, a, b, q, A, B)$

und $K_{pr} = d$

$$\xleftarrow{E(p, a, b, q, A)}$$

$$\xrightarrow{E(p, a, b, q, A)}$$

$$\xrightarrow{(K_{pb}) = (p, a, b, q, A, B)}$$

ECDSA SIGNIEREN UND VERIFIZIEREN

Alice

Bob

Wähle emphermal k_E mit $0 \leq k_E \leq q$

Berechne: $R = k_E A$

Setze: $r = x_R$

Berechne: $s \equiv (h(x) + d \cdot r)k_E^{-1} \pmod{q}$ $\xrightarrow{(x, r, s)}$

Berechne $w \equiv s^{-1} \pmod{q}$

Berechne $u_1 \equiv w \cdot h(x)w \pmod{q}$

Berechne $u_2 = r \cdot w \pmod{q}$

Berechne $P = u_1 A + u_2 B$

Falls $x_P \equiv q \rightarrow$ valdie

ANGRIFFE AUF SIGNATURVERFAHREN

- Die Sicherheit von ECDSA hängt stark vom Zufallswert K ab:
 - Falls k bekannt wird, kann $K_{pr} = x$ berechnet werden
 - Falls k wiederverwendet wird, kann $K_{pr} = x$ berechnet werden [PS3]
- Bei ECDSA müssen bestimmte Signaturen abgefangen werden (z.B. $s = 0$ [Orac])
- RSA Verschlüsselung und Signaturen niemals mit dem gleichen Schlüsselpaar!
 - Verschlüsselte Nachricht könnte entschlüsselt werden via Anfrage zur Signatur
 - Unbeabsichtigte Signatur könnte erzeugt werden via Anfrage zur Entschlüsselung
- RSA benötigt Paddingverfahren (z.B. RSA-PSS) zur sicheren Signaturerzeugung

ASYMMETRISCHE SIGNATURVERFAHREN

- Direktes Signieren und Verifizieren von großen Nachrichten ist sehr ineffizient
 - Signieren: $s = k^{-1} \cdot (M + r \cdot r) \bmod q$
 - Verifizieren: $u_1 = M \cdot w \bmod q$
- Analog zur hybrider Verschlüsselung: Große Nachricht mit Hilfsfunktion in einen kleinen, eindeutigen Fingerabdruck umwandeln, der dann signiert wird
- Anforderung an Hilfsfunktion und Fingerabdruck
 - Jede Person sollte die Hilfsfunktion berechnen können
 - Es sollte nicht möglich sein, vom Fingerabdruck auf eine Nachricht zurückzurechnen

ZUSAMMENFASSUNG

- MACs basierend auf Hashfunktionen und symmetrischer Verschlüsselung
- Besprechung von verschiedenen MAC Verfahren
- Unterschiede zwischen MACs und digitalen Signaturen
- Korrekter Einsatz von MACs oder digitalen Signaturen beurteilen
- Sicherheitsgarantien der Digitalen Signaturen
- Beziehung zwischen dem RSA Verschlüsselungs- und Signaturverfahren
- Existierende Digitale Signaturverfahren
- Elliptischen Kurven gegenüber primen Restklassenringen



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Protokolle für Authentizität

June 7, 2023

Marc Stöttinger

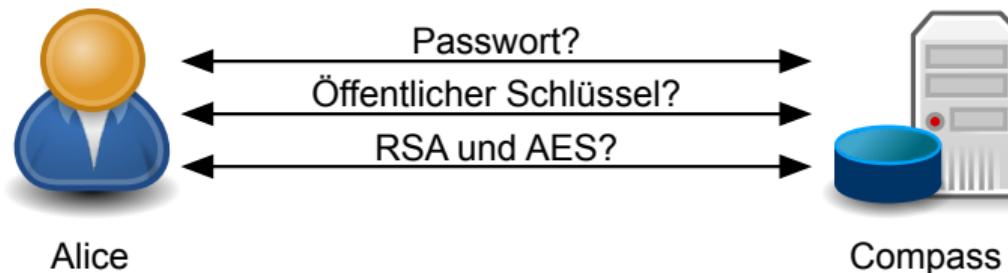


Wo das Wissen aufhört, beginnt der Glaube.

Augustinus Aurelius

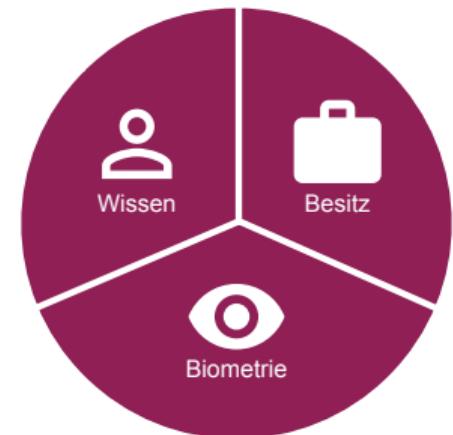
MOTIVATION

- **Bisher:** Kryptographische Verfahren ohne Anwendungskontext
 - Wie werden sie in der Praxis eingesetzt und was gibt es zu beachten?
- **Anwendungsbeispiel:** Alice möchte sich bei Compass einloggen
 - Wie kann sich Alice authentifizieren?
 - Wie kann sich der Server authentifizieren?



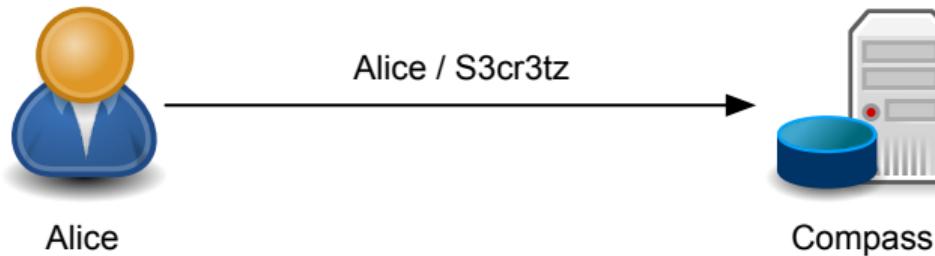
AUTHENTIFIKATIONSMERKMALE

Merkmal	Sicherheit basiert auf	Beispiel
Wissen	Nur Nutzer bekannt	Passwort, PIN, Sicherheitsfrage
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	Fingerabdruck, Gesicht, Iris, Retina, Gang, Tastaturanschläge
Besitz	Im Besitz von Nutzer und Entwendung wird bemerkt	Chipkarte (z.B. SIM-Karte), Smartphone, USB Token, TAN Generatoren



PASSWORT

- Passwörter sind das häufigste Authentifizierungsverfahren im Internet
- **Nur Alice kennt** einen geheimen String, der nicht erraten werden kann. Probleme:
 1. **Nicht erraten**: Wie sieht ein gutes, nicht-erratbares Passwort aus?
 2. **Nur Alice kennt**: Das Passwort wird eingegeben, übertragen und auf dem Server gespeichert!
 3. **Sicher zu benutzen**: Das Passwort kann eventuell im Klartext übertragen werden!



User	Password
Alice	S3cr3tz
Bob	1233456
Eve	*Lauschi*
...	...

ERRATBARKEIT PASSWORT

- Menschen wählen Passwörter nicht sicher, sondern leicht merkbar
- Passwort Policies, um Menschen zu sicheren Passwörtern zu **erziehen**
- Jahrelanges Tauziehen zwischen Nutzer und Policies führte nicht zu sichereren Passwörtern

Beispiel Passwort	Angriffsstrategie	Passwort Policy Update
asdf	Zufällige Buchstabenkombination testen	Mindestens 8 Zeichen
passwort	Wörter aus Wörterbuch testen	Großbuchstaben müssen enthalten sein
PassWort	Buchstabenkombinationen klein und groß	Ziffern hinzufügen
PassWOrt21	Jahreszahlen anhängen und Gängige Substitutionen (e → 3)	Sonderzeichen hinzufügen
P\$ssWOrt21	Gängige Substitutionen (4 → \$)	Passwortupdates nach 90 Tagen
P\$ssWOrt22	Counter am Ende hochzählen	...

PASSWORT-TIPPS FÜR NUTZER

- **Kurz und komplex:** 8+ Zeichen aus Groß/Kleinschreibung, Ziffern und Sonderzeichen
 - Beispiel: Ein Passwort, das aus einem Satz abgeleitet wurde! → 1P,dae\$aw!W2
- **Lang und weniger komplex:** 20+ Zeichen aus Groß/Kleinschreibung
 - Beispiel: FischbrötchenKaufErinnerungMaerkteFrisch
- Nutzen Sie **einzigartige Passwörter** für wichtige Dienste
 - Z.B., eMail, Online Banking, Unternehmens-IT
 - Passwortmanager, um Passwörter zu generieren und zu speichern
- Regelmäßig **Passwort-Leaks** prüfen (z.B. [HIBP])
 - Für kontrollierte Domains ist eine Registrierung möglich
 - Passwörter können via HIBP API geprüft werden

GEGENMASSNAHMEN ZU BRUTE-FORCE ANGRIFFE

- Brute-Force Angriffe können Passwörter zwar raten, müssen diese aber auch prüfen
 1. Probe-hafter Login auf Webseite
 2. Abgreifen geheimer Daten auf dem Server
- Detektierende und reaktive Gegenmaßnahme **Probe-hafter Login**
 - Wartezeit zwischen Anmeldeversuchen als exponentiell wachsend konfigurieren
 - Blockieren einer IP nach X Versuchen
 - Sperrung eines Accounts für X Minuten
 - Logging der Versuche zur Erkennung eines Angriffs
- Preventive Gegenmaßnahmen zum **Abgreifen geheimer Daten** auf dem Server
 - Speichern in einer nicht-auslesbaren Form via Hashing

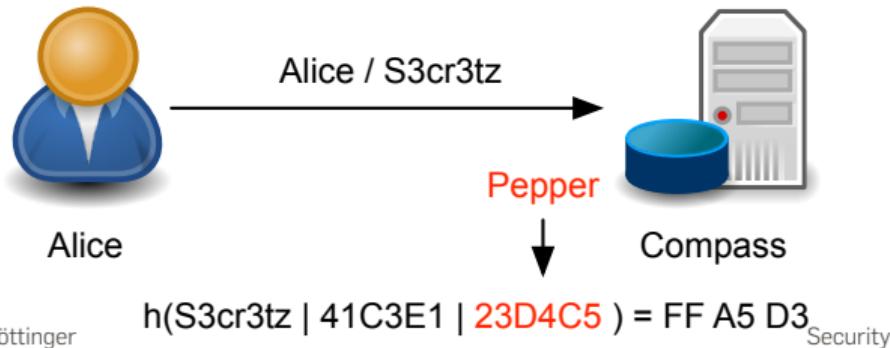
SICHERES SPEICHERN VON PASSWÖRTERN - SALT UND PEPPER HASHING

→ Passwörter werden mit zusätzlichen und zufälligen Daten gehashed

1. **Salt**: Individueller Zufallswert, der mit Passwort gespeichert wird
2. **Pepper**: Zufälliger und geheimer Wert, der für alle Passwörter konstant ist
3. Berechnung als $h(\text{Passwort} \parallel \text{Salt} \parallel \text{Pepper})$

→ Sicherheitsgewinn:

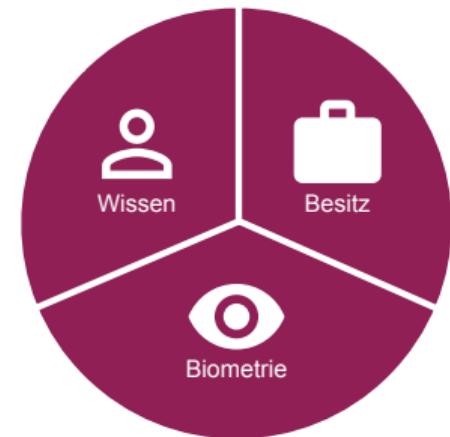
- **Salt**: Rainbow Tables verhindert, da für jedes Passwort ein eigener Salt gewählt wird
- **Pepper**: Brute-force erschwert, da Pepper geraten werden muss (solange Pepper unbekannt)



User	PW Hash	PW Salt
Alice	FF A5 D3	41C3F1
Bob	ED 12 4F	12D32E
Eve	1F 3E 38	2945F2
...

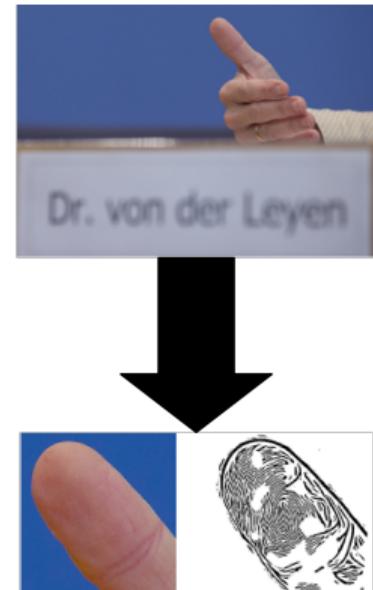
AUTHENTIFIKATIONSMERKMALE

Merkmal	Sicherheit basiert auf	Beispiel
Wissen	Nur Nutzer bekannt	Passwort, PIN, Sicherheitsfrage
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	Fingerabdruck, Gesicht, Iris, Retina, Gang, Tastaturanschläge
Besitz	Im Besitz von Nutzer und Entwendung wird bemerkt	Chipkarte (z.B. SIM-Karte), Smartphone, USB Token, TAN Generatoren



BIOMETRISCHE IDENTIFIKATION

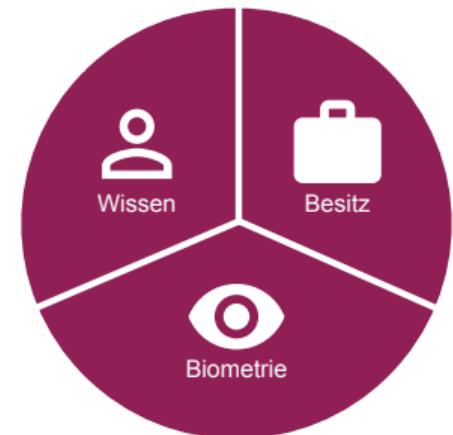
- Biometrische Merkmale sind zwar schwer von Menschen zu kopieren, aber
 - Öffentlich einsehbar und nicht sonderlich geschützt
 - Maschinell nachstellbar, wenn das Modell bekannt ist
 - Unsicher wenn Daten einmal veröffentlicht wurden
- Nutzbarkeit ist sehr gut
 - Biometrische Merkmale immer dabei
 - Intuitiv, da weite mediale Verbreitung
- Erkennung häufig fehlerhaft
 - Externe Einflüsse wirken störend (Licht, Kälte)
 - Merkmale ändern sich über die Zeit (Gesicht, Deutlichkeit des Fingerabdrucks)



Quelle: CCC2014

AUTHENTIFIKATIONSMERKMALE

Merkmal	Sicherheit basiert auf	Beispiel
Wissen	Nur Nutzer bekannt	Passwort, PIN, Sicherheitsfrage
Biometrie	Physiologisch oder verhaltenstypisch einzigartige und unkopierbare Merkmale einer Person	Fingerabdruck, Gesicht, Iris, Retina, Gang, Tastaturanschläge
Besitz	Im Besitz von Nutzer und Entwendung wird bemerkt	Chipkarte (z.B. SIM-Karte), Smartphone, USB Token, TAN Generatoren



AUTHENTIFIKATION VIA BESITZ

- Wissen und Biometrie können kopiert werden ohne, dass Nutzer es bemerkt
- **Lösung:** Physisches Objekt, das Nutzer ständig bei sich tragen kann und dessen Diebstahl bemerkt wird
 - Das Objekt sollte nicht einfach kopierbar sein
 - Geheimnisse auf dem Objekt dürfen nicht auslesbar sein
- Zur Authentifikation darf der Schlüssel den Chip nicht verlassen
 - Spezielle Protokolle benötigt



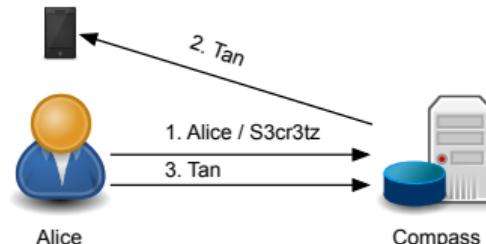
VOR- UND NACHTEILE DER AUTHENTIFIZIERUNGSTECHNIKEN

Merkmal	Vorteile	Nachteile
Wissen	<ul style="list-style-type: none"> → Einfach zu implementieren → theoretisch sicher → keine zusätzliche Technik 	<ul style="list-style-type: none"> → Sicherheit abhängig vom gewählten Passwort → schwer zu merken bei vielen Zugängen → Kopieren nicht bemerkbar
Biometrie	<ul style="list-style-type: none"> → Kein Transport oder Merken notwendig → Eindeutig pro Mensch 	<ul style="list-style-type: none"> → Physikalischer Scanner benötigt → Externe Faktoren können zu Fehlern führen → Kopieren manchmal nicht bemerkbar → Merkmal nicht wechselbar → Ein Leak reicht, um Sicherheit des Merkmals zu korrumpern → Kann auch gegen Nutzer verwendet werden
Besitz	<ul style="list-style-type: none"> → Kein Merken notwendig → Entwendung ist bemerkbar → Standardmäßig hohe Sicherheit 	<ul style="list-style-type: none"> → Ggf. physikalische Schnittstelle benötigt (Smart Card Reader) → Aufwändig in der Umsetzung → Muss von Nutzer mittransportiert werden

ZWEI- UND MEHRAKTOURAUTHENTIFIZIERUNG

- Mechanismen aus Wissen, Besitz und Biometrie können kombiniert werden, um mehr Sicherheit zu erreichen
 - **Zweifaktor Authentifizierung**: Kombination von zwei Mechanismen aus verschiedenen Kategorien
 - **Multifaktor Authentifizierung**: Kombination von mehr als zwei Mechanismen aus verschiedenen Kategorien

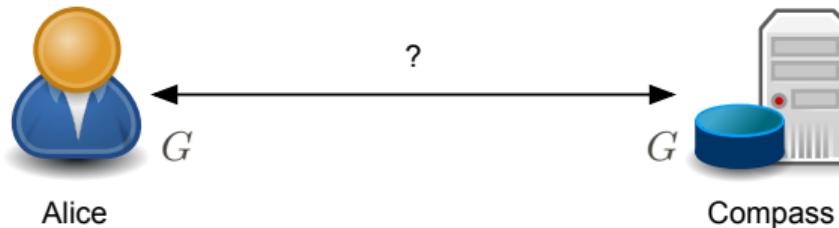
- **Beispiel: Online Überweisung**
 - Wissen (Passwort)
 - Besitz (SIM Karte / Smartphone)



DISKUSSION IN KLEINEN GRUPPEN

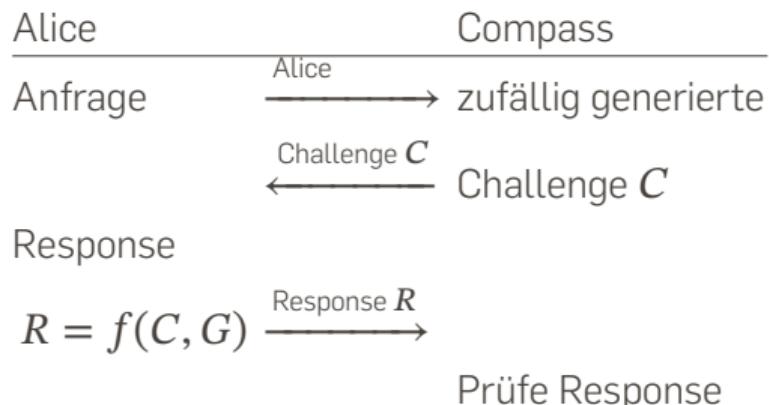
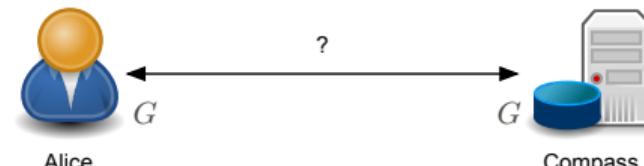
Authentifizieren ohne Senden von Geheimnissen

Alice's Smart Card und der Server kennen ein gemeinsames Geheimnis G (Passwort, Schlüssel, ...). Wie kann sich Alice's Smart Card gegenüber dem Server authentifizieren, ohne das Geheimnis G zu senden (Nehmen Sie an, dass der Angreifer alle Nachrichten lesen und wiedereinspielen kann.)?



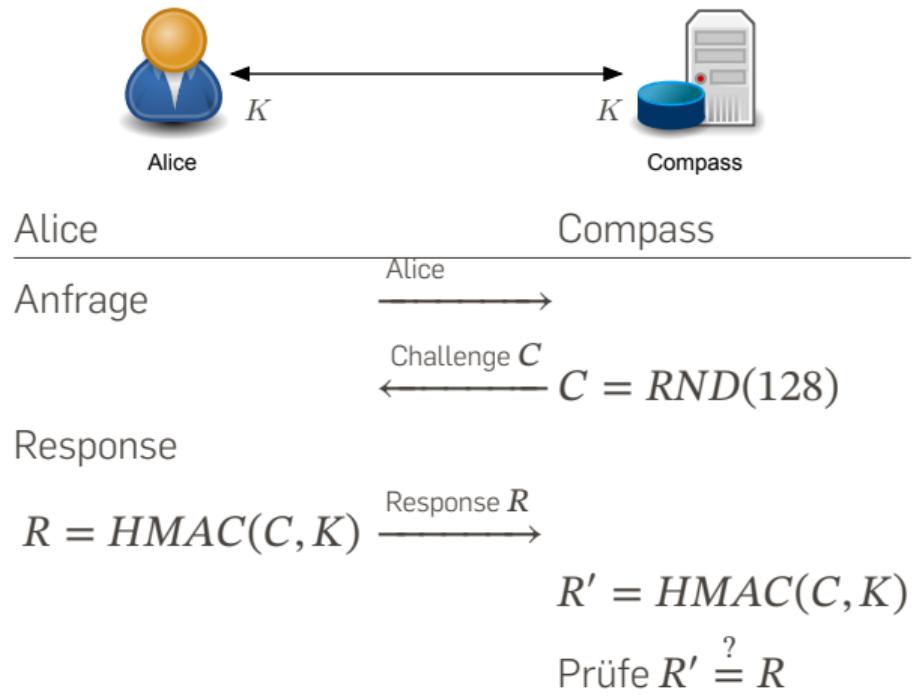
CHALLENGE-RESPONSE PROTOKOLLE

- **Initial:** Der Server und Alice's Smart Card kennen beide ein Geheimnis G
- Funktion f kann durch verschiedene kryptographische Verfahren instanziert werden (z.B.: (A)Symmetrische Verschlüsselung, MAC Funktion, ...)
- Nötige Eigenschaften von f :
 - Einwegeigenschaft
 - Schwache Kollisionsresistenz



CHALLENGE-RESPONSE PROTOKOLLE MIT HMAC [RFC2617]

- Geheimnis besteht aus symmetrischem Schlüssel K
- Funktion f ist HMAC-SHA1 (**HMAC**)
- Eigenschaften für f direkt gegeben durch MAC Verfahren
- Als Challenge wird eine 128-Bit Zufallszahl generiert
- Zur Verifikation: Server führt die gleiche Berechnung durch



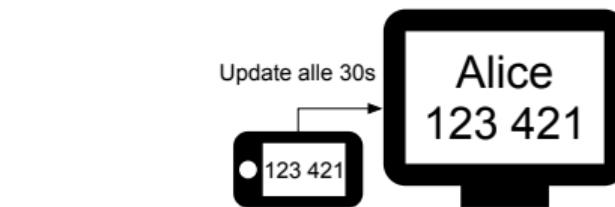
AUTHENTIFIKATION VIA BESITZ EINMALPASSWORT (OTP)

→ **Einmalpasswort (OTP)**: Automatisiert generierter und einmalig nutzbarer Verifikationscode

- HMAC-based OTP (HOTP): HMAC-SHA1 auf Zähler und Schlüssel [RFC4226]
- Time-based OTP (TOTP): HOTP mit Zeit statt Zähler [RFC6238]



1. Backend generiert Schlüssel K für Nutzer
2. Gerät liest und speichert Schlüssel K

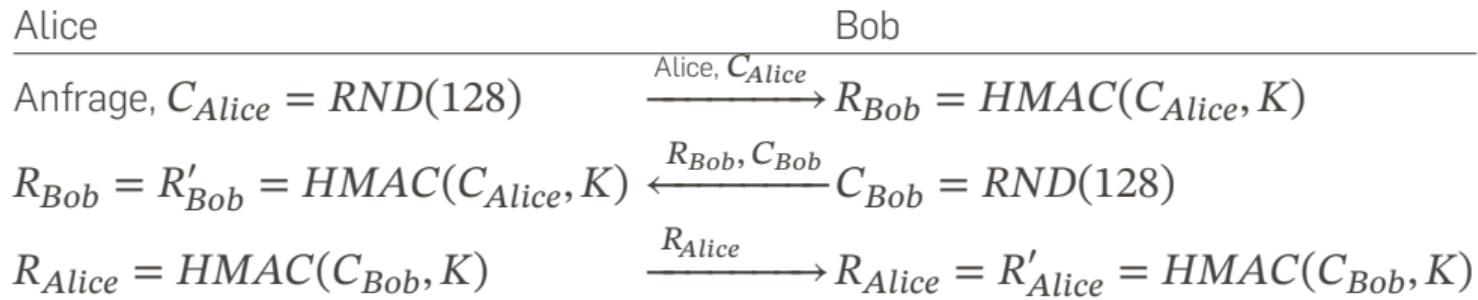


1. Gerät berechnet alle 30s $HMAC(K, Zeit)$
2. Backend verifiziert MAC und prüft Zeittoleranz

MUTUAL CHALLENGE-RESPONSE PROTOKOLLE



OPTIMIERTES MUTUAL CHALLENGE-RESPONSE PROTOKOLL?



KEINE EIGENEN MUTUAL CHALLENGE-RESPONSE PROTOKOLLE ENTWICKLEN

Mallory

Anfrage, $C_{Mallory} = RND(128)$ $R_{Bob} \stackrel{?}{=} R'_{Bob} = HMAC(C_{Alice}, ?)$ $R_{Alice} = HMAC(C_{Bob}, ?)$ R_{Bob} von Session 2

Bob

 $\xrightarrow{\text{Alice, } C_{Mallory}} R_{Bob} = HMAC(C_{Mallory}, K)$ $\xleftarrow{R_{Bob}, C_{Bob}} C_{Bob} = RND(128)$ $\xrightarrow{R_{Alice}} R_{Alice} \neq R'_{Alice} = HMAC(C_{Bob}, K)$ $\xrightarrow{R_{Bob}} R_{Bob} = R'_{Alice} = HMAC(C_{Bob}, K)$

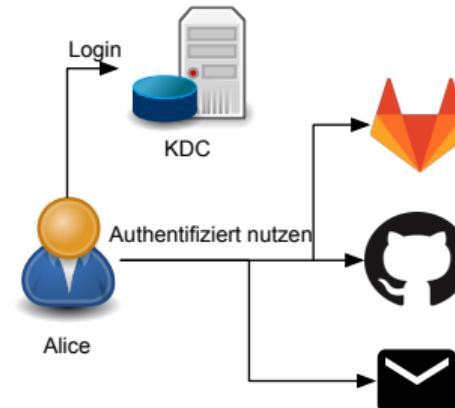
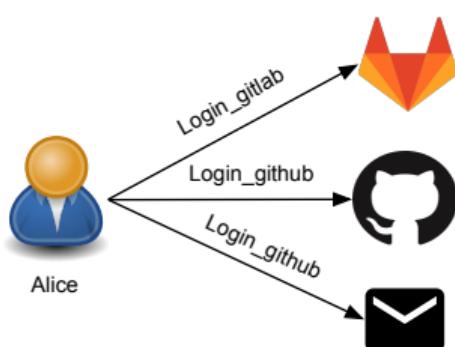
Session 2:

Anfrage

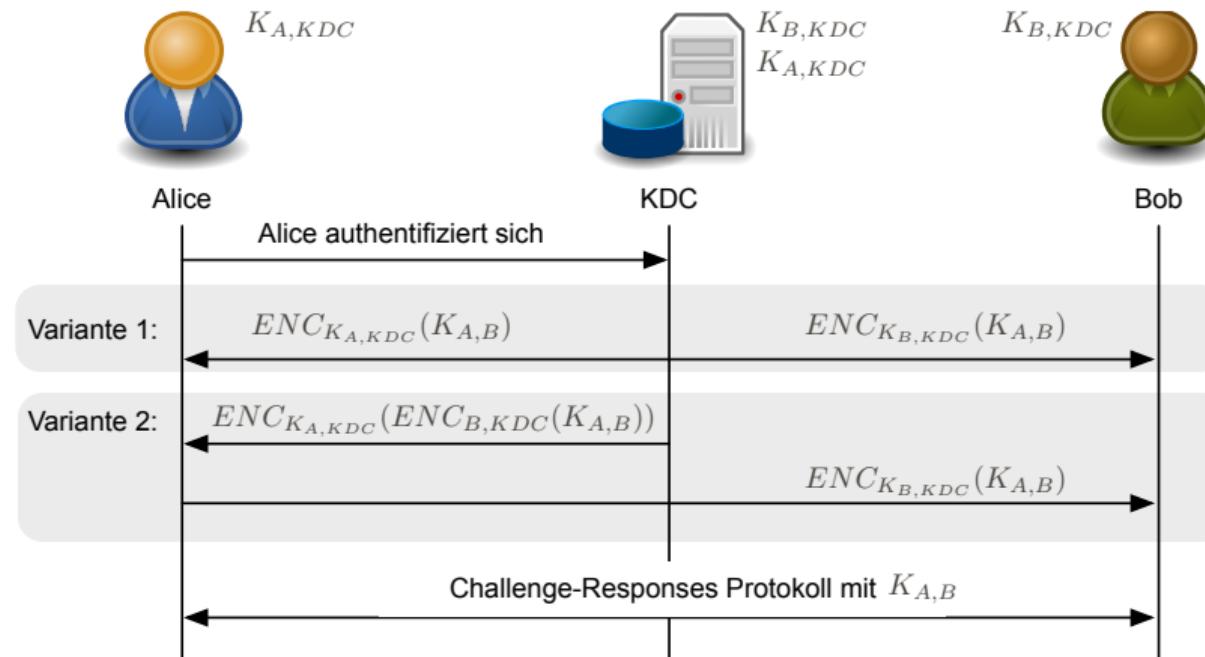
 R_{Bob} $\xrightarrow{\text{Alice, } C_{Bob}} R_{Bob} = HMAC(C_{Bob}, K)$ $\xleftarrow{R_{Bob}, C_{Bob}} C_{Bob} = RND(128)$

SINGLE SIGN-ON (SSO) SYSTEME

- Viele Systeme verlangen eine individuelle Authentifizierung
- Lösung: Ein Schlüsselverwaltungsservers (KDC) ermöglicht einen Single Sign-On (SSO) Service, die zentrale Authentifizierung ermöglichen
 - Unternehmensnetzwerke: Kerberos
 - Privat: OAuth/OpenID Systeme (Google, Facebook, ...)



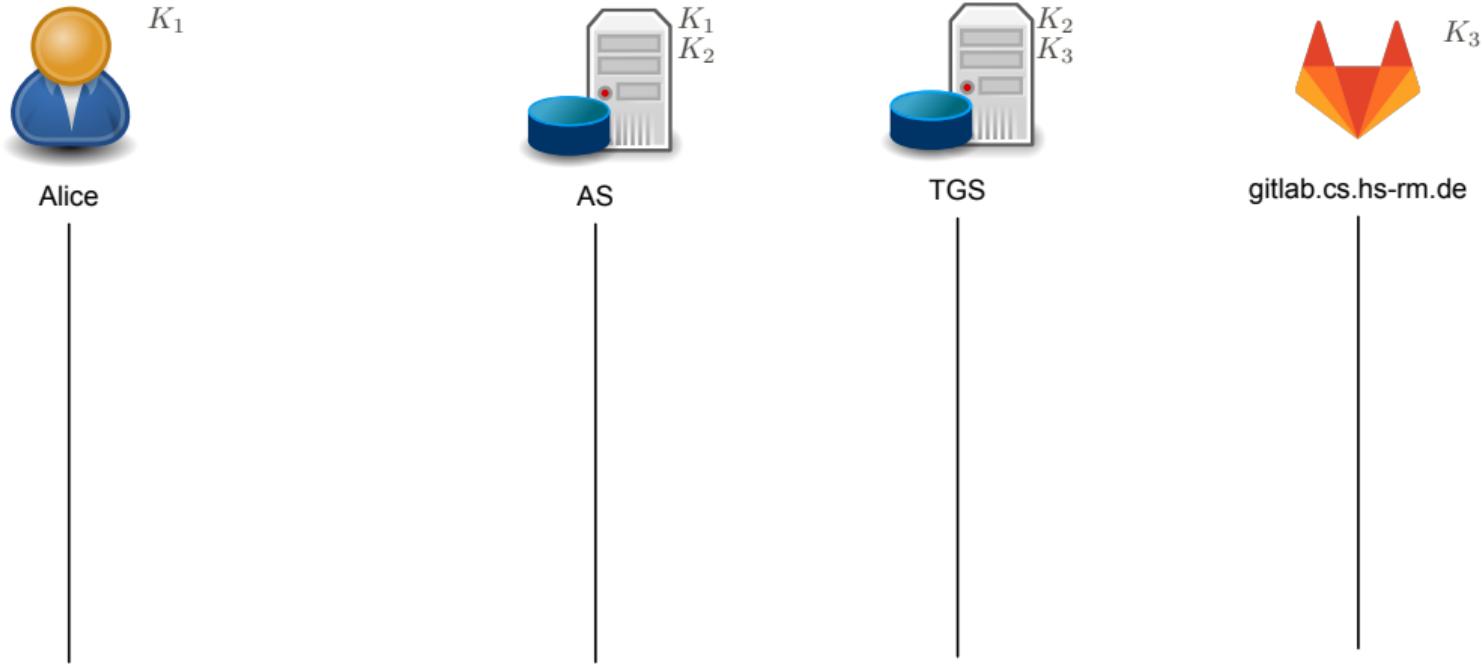
AUFBAU VON KEY DISTRIBUTION CENTERS FÜR AUTHENTIFIZIERUNG



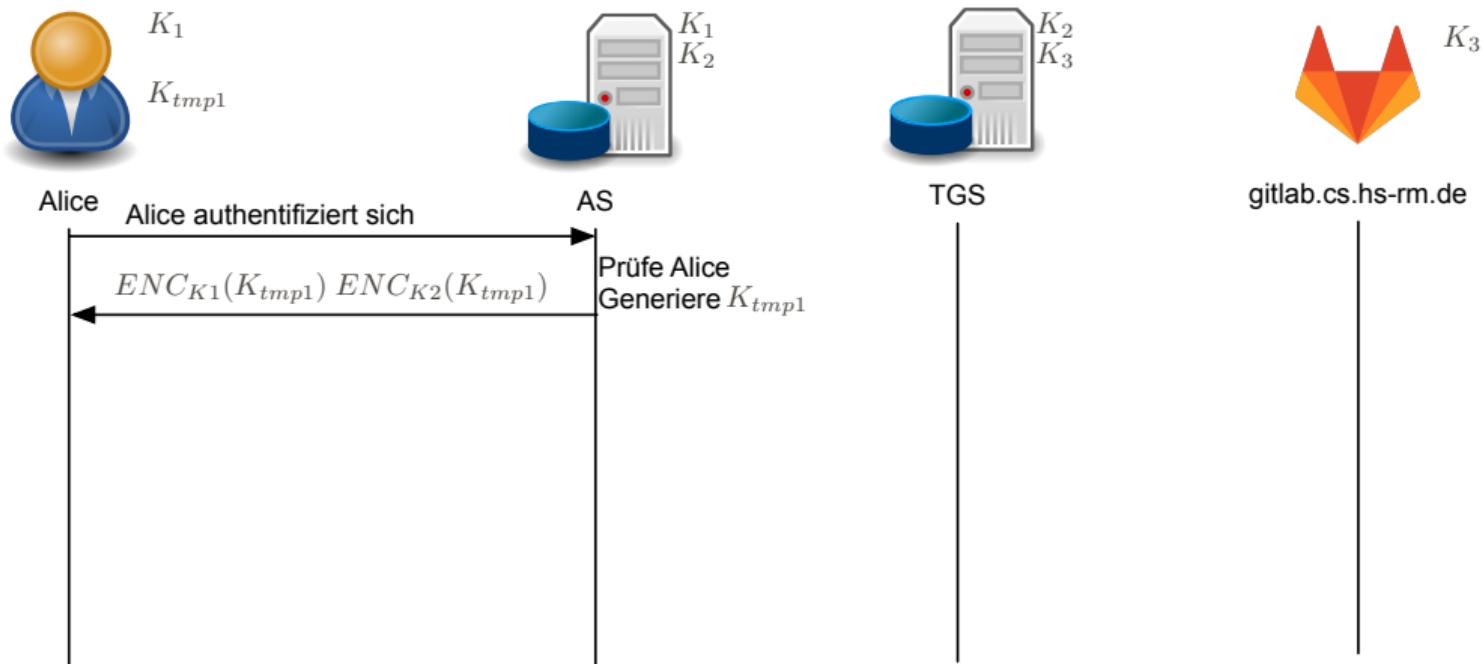
KERBEROS SINGLE SIGN-ON

- Kerberos wurde am MIT entwickelt und ist der de-facto Standard für SSO in Unternehmen
- Nutzer authentifiziert sich nur gegenüber Domain Controller und bekommt von diesem Tickets zur Nutzung von Services ausgestellt
- Der Kerberos Server übernimmt die Rolle des KDC
 - **Authentifizierungsserver (AS)**: Prüft Identität und stellt ein Ticket für eine Sitzung über eine gewisse Dauer aus
 - **Ticketgenehmigungsserver (TGS)**: Prüft, ob Nutzer-Sitzung aktiv ist und stellt Tickets zur Nutzung der Services aus

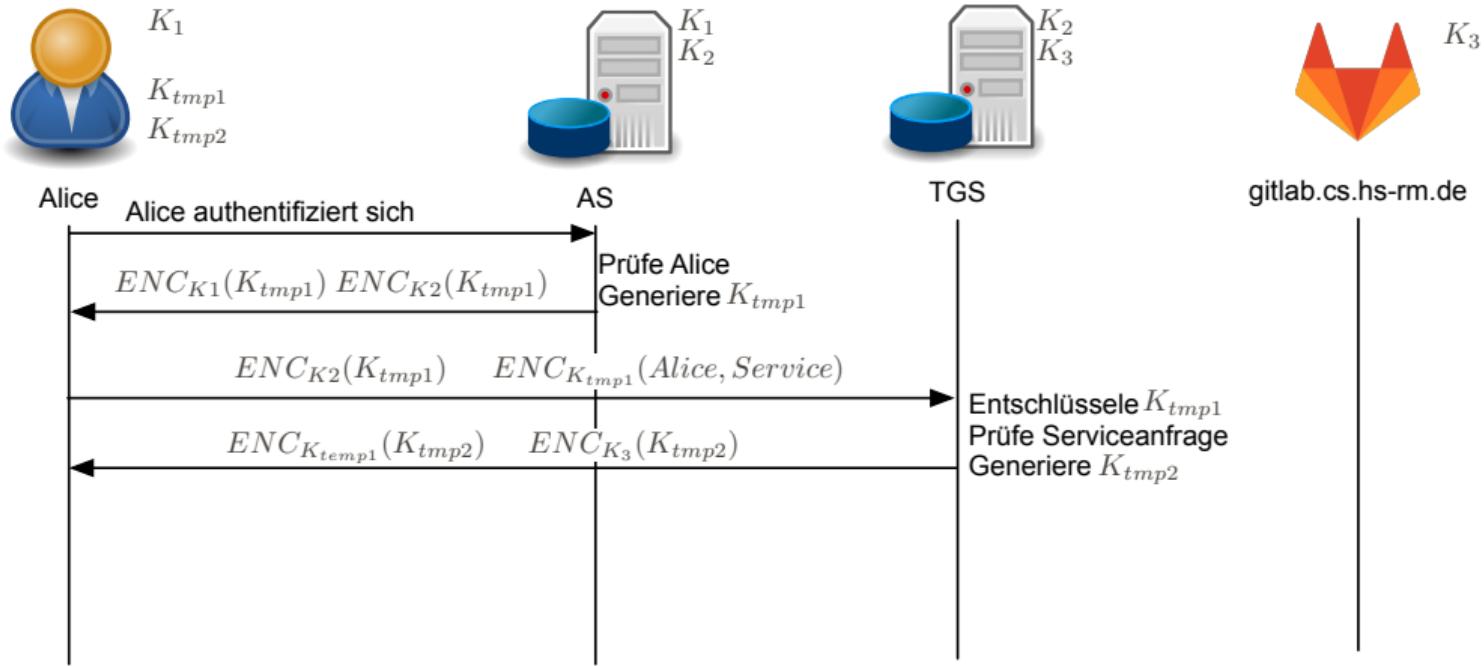
ABLAUF KERBEROS



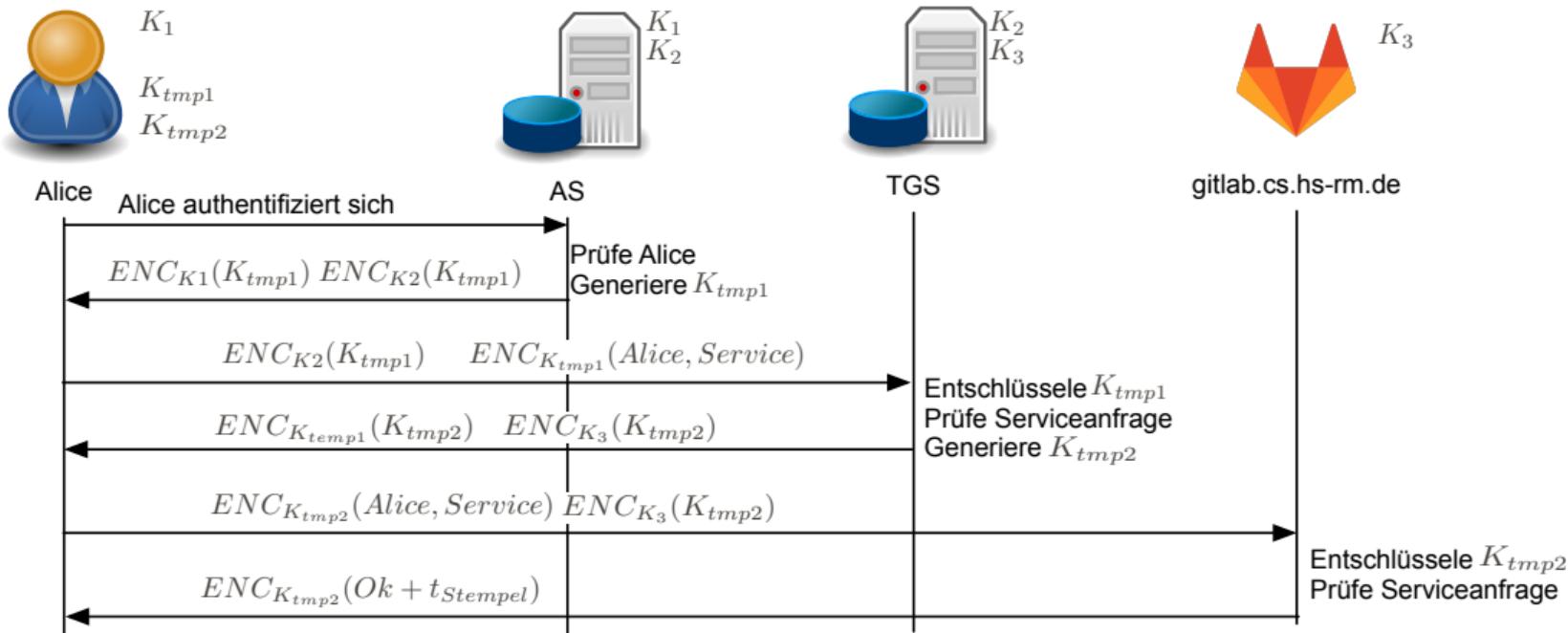
ABLAUF KERBEROS



ABLAUF KERBEROS

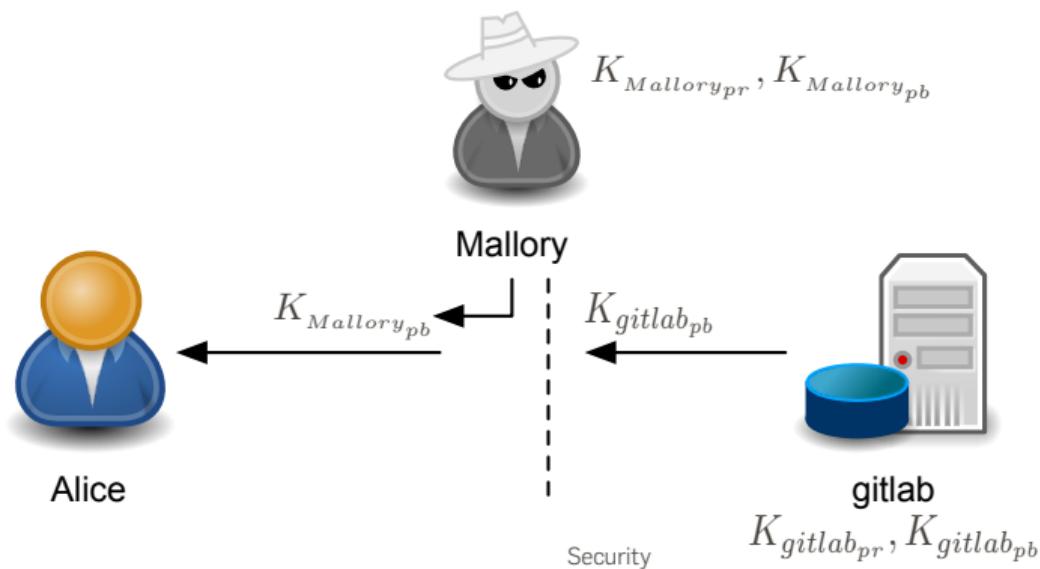


ABLAUF KERBEROS



AUTHENTIFIKATION OHNE AUSGETAUSCHTE GEHEIMNISSE

- **Problem:** Webseite wurde noch nie besucht, kein ausgetauschtes Geheimnis verfügbar
- Bedrohung: Mallory tauscht öffentlichen Schlüssel bei initialer Übertragung aus, um Kommunikation abzufangen



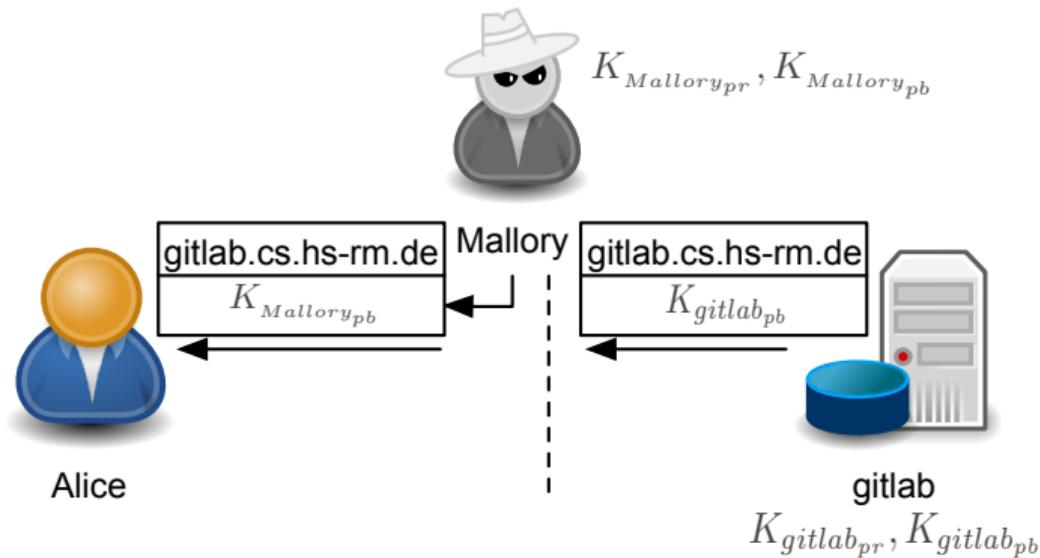
ZERTIFIKATE(1/3)

- Zertifikate verknüpfen den öffentlichen Schlüssel des Webservers mit Attributen, die nachgeprüft werden können (z.B. DNS-Name oder IP)
- Zertifikate enthalten u.a. die folgenden Informationen
 - Gültigkeitsdauer (Beginn / Ende)
 - Verwendungszweck des Schlüssels (Signieren / Verschlüsseln)
 - Verwendete kryptographischen Algorithmen
- Zertifikat mit öffentlichem Schlüssel wird übertragen

Subject Name	
Common Name	gitlab.cs.hs-rm.de
Issuer Name	
Country or Region	US
Organisation	Let's Encrypt
Common Name	R3
Serial Number	03 7C BE 99 A1 D0 4B 85 E0 36 24 F3 5D
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.11
Parameters	None
Not Valid Before	Tuesday, 21. February 2023 at 08:21:00 C
Not Valid After	Monday, 22. May 2023 at 09:20:59 Centra
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	None
Public Key	256 bytes: BB 3E B9 B2 5E 15 22 D2 ...
Exponent	65537
Key Size	2.048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes: 34 DB 0E C9 49 10 1D 50 ...

ZERTIFIKATE(2/3)

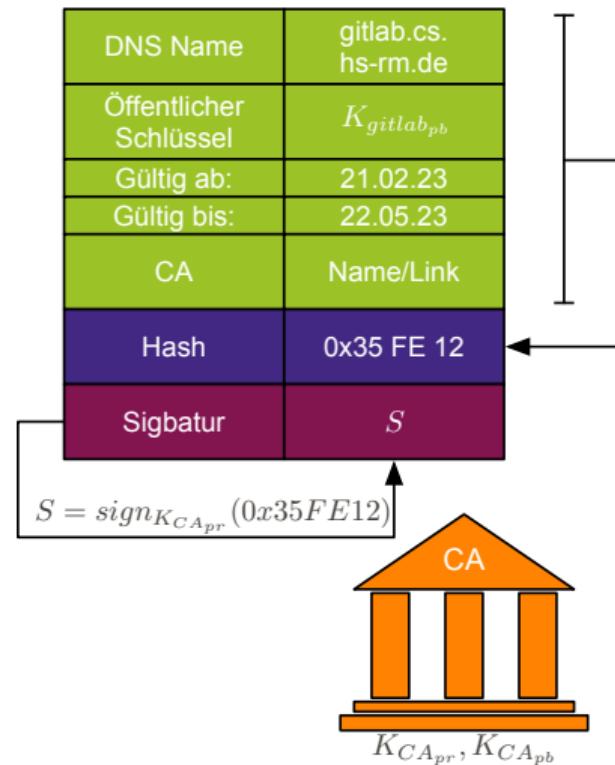
→ **Problem:** Authentizität des Zertifikates ist nicht sichergestellt.



ZERTIFIKATE(3/3)

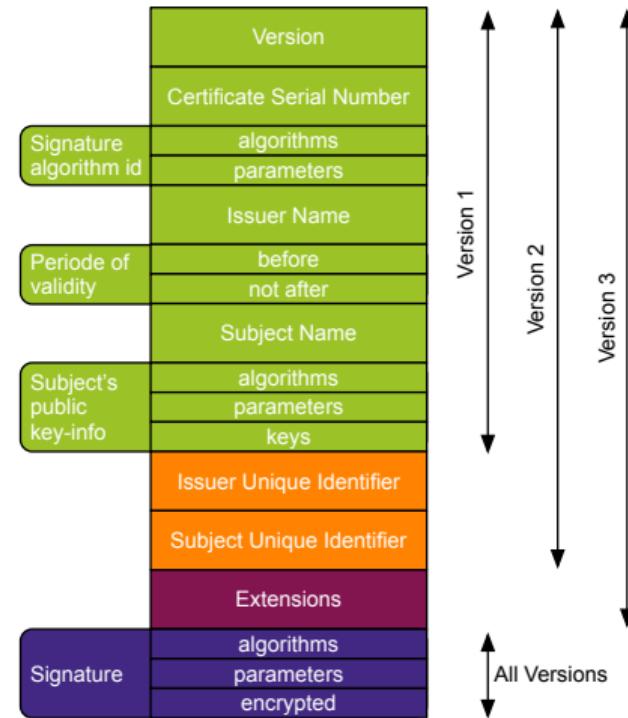
- Zertifizierungsstelle (CA) erstellt digitale Signatur eines Zertifikates
 1. Identität der CA wird in Zertifikat aufgenommen
 2. Hashwert über Zertifikatsinhalt wird gebildet
 3. CA mit Schlüsselpaar (K_{pr}, K_{pb}) signiert Hashwert des Zertifikates
 4. Signatur wird an Zertifikat angehangen

- Nutzer kann Zertifikat mit Schlüssel $K_{CA_{pb}}$ der CA prüfen



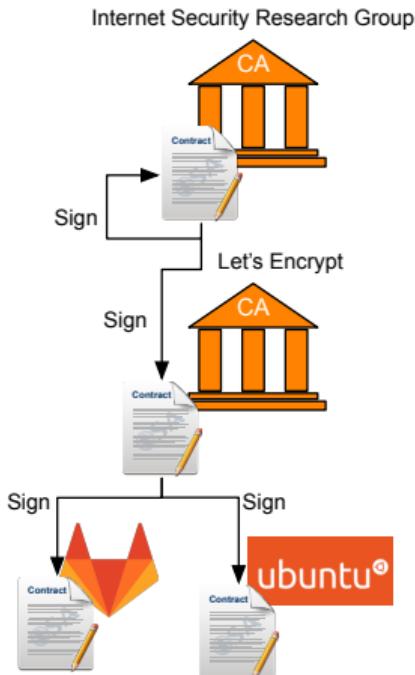
X.509 ZERTIFIKAT

- Version 1:
 - X.509-Version
 - Seriennummer
 - Algorithmus der Unterschrift
 - Name des Ausstellers
 - Gültigkeit
 - Zertifikatsinhaber
 - Öffentlicher Schlüssel des Zertifikatsinhabers
- Ergänzungen in Version 2:
 - Austeller-ID
 - Zertifikatsinhaber-ID
- Ergänzungen in Version 3:
 - Erweiterungen
- Signatur



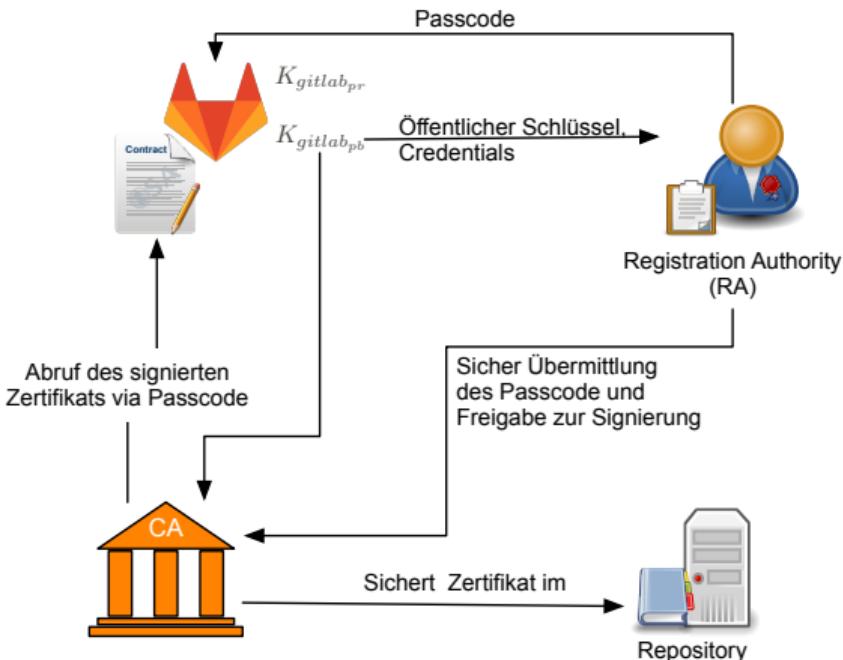
PUBLIC KEY INFRASTRUKTUR (PKI)

- Eine **Public Key Infrastruktur (PKI)** wird benötigt, um
 - Zertifikate zu erstellen und zu signieren
 - Informationen über Zertifikate bereitzustellen
 - Unsichere Zertifikate zu entfernen
- Teilnehmende in einer PKI
 - **Wurzel-CA**: Oberste Zertifizierungsstelle
 - **CA**: Stellt Zertifikate aus
 - **Webseiteninhaber**: Beantragen Zertifikate unter Nachweis der Identität (z.B. durch gitlab einer zufällig erzeugten Datei unter einer gegebenen URL)
- Zertifikat der Wurzel-CA liegt im Betriebssystem



ZERTIFIKAT ERZEUGEN

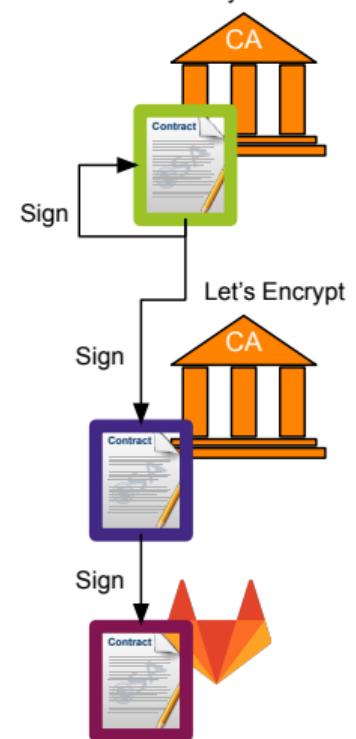
1. Generieren von $K_{gitlab_{pr}}$ und $K_{gitlab_{pb}}$
2. Übermitteln der Credentials und $K_{gitlab_{pb}}$ an (RA)
3. RA übermittel Passcode zum Abruf des Zertifikats
4. Übermittlung des $K_{gitlab_{pb}}$ an die CA
5. RA übermittelt Passcode an CA, wenn Credentials valide sind
6. CA stellt signiertes Zertifikat bereit, welches über Passcode abrufbar ist
7. CA übermittel Zertifikat an ein Repository zur Archivierung



ZERTIFIKAT PRÜFEN

1. gitlab Zertifikat unbekannt
2. Prüfe Gültigkeit und Identität
3. Prüfe Signature
4. CA Zertifikat (Let's Encrypt) unbekannt
5. Prüfe Gültigkeit und Identität
6. Prüfe Signature
7. Root CA Zertifikat bekannt und verifiziert
8. Signatur CA Zertifikat OK
9. CA Zertifikat Authentisch
10. gitlab Zertifikat Signatur OK
11. gitlab Zertifikat ist authentisch

Internet Security Research Group



ZERTIFIKAT WIDERRUFEN

- Grund für Certificate Revocation:
 - K_{pr} kompromittiert
 - Gerät , Benutzer oder Identität sperren
- CA erstellt einen Certificate Revocation List (CRL)
 - Seriennummer der widerrufen Zertifikate
 - CRL werden auf Webservern oder in Active Directories hinterlegt
 - Gültigkeit von Tagen, somit keine kurzfristige Sperrung
- Alternative Online Certificate Status Protocol (OCSP [RFC 6960])
 - Validierungsdienst zur Gültigkeit von Zertifikaten

ZERTIFIKAT PINNING

- Angabe welches Zertifikat für welche Webseite anerkannt wird
 - Öffentlicher Schlüssel-Hash verweist auf ein Zertifikat der Website selbst oder auf ein Stamm-/Zwischenzertifikat einer CA, die der Website bekannt ist
- Erschwert MiTM-Angriffe und Austausch von Zertifikaten
 - Direkte Verbindung zwischen einem Zertifikat und einem Hostnamen
 - Bei Erstzugriff wird der Schlüssel-Hash im Browser gespeichert
 - Hashwert kann Zertifikatsschlüssel oder von der ausstellenden CA sein
- Erhöhter Managementaufwand durch Zertifikat Pinning
 - Es werden mindestens zwei valide Hashwerte benötigt, damit ein neuer Hashwert nachgeladen werden kann

ZUSAMMENFASSUNG

- Merkmale zur Authentifizierung sowie Kombinationen dieser (2FA)
- Vor- und Nachteile der verschiedenen Merkmale
- Challenge-Response Protokoll zur Authentifizierung
- Konstruktion von Challenge-Response Protokollen basierend auf kryptographischen Primitiven konstruieren
- Prinzip von Single-Sign-On Protokollen und Kerberos
- Verwendungszweck von Zertifikaten und einer PKI
- Generations- und Verifikationsprozess von Zertifikaten innerhalb einer PKI



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Protokolle für sichere Kommunikation

June 16, 2023

Marc Stöttinger

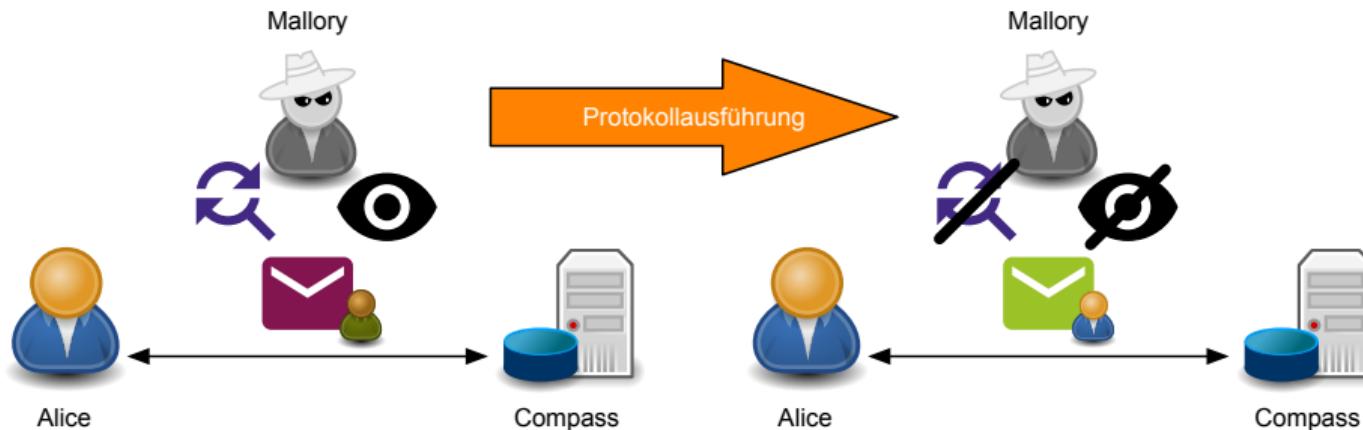


Secure communication protocols serve as the fortified gateways that protect the sanctity of our digital interactions.

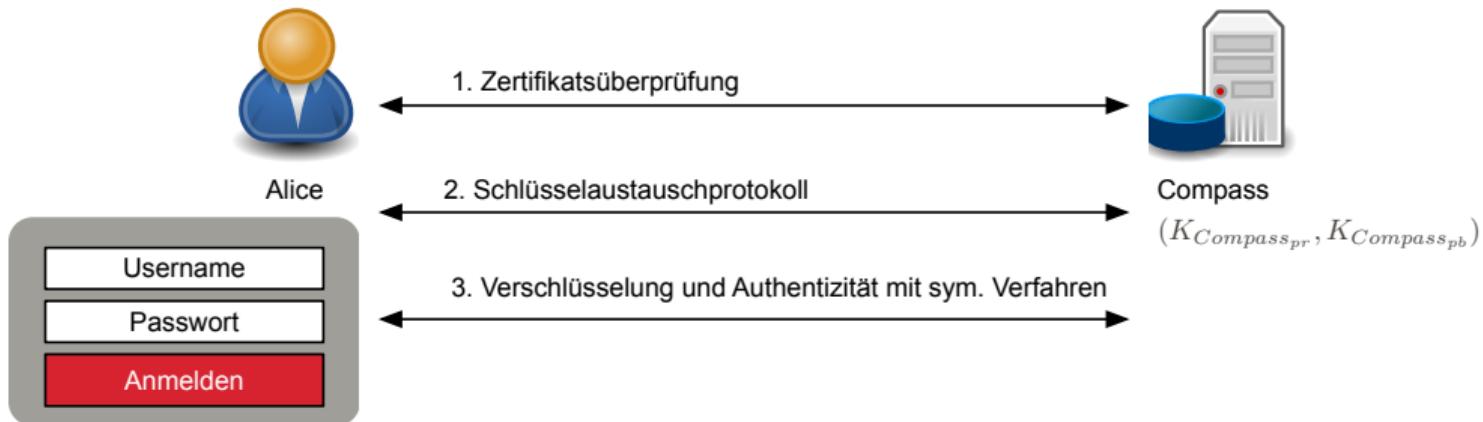
Whitfield Diffie

MOTIVATION

- Bisher: Protokolle zur Authentifikation von Personen im Internet
- Heute: Protokolle zum Aufbau eines sicheren Kommunikationskanals
 - Start: Alle Nachrichten abhör- und manipulierbar
 - Ziel: Sicherer (vertraulicher, authentischer und integrier) Kommunikationskanal



GROBABLAUF SICHERE KOMMUNIKATIONSPROTOKOLLE



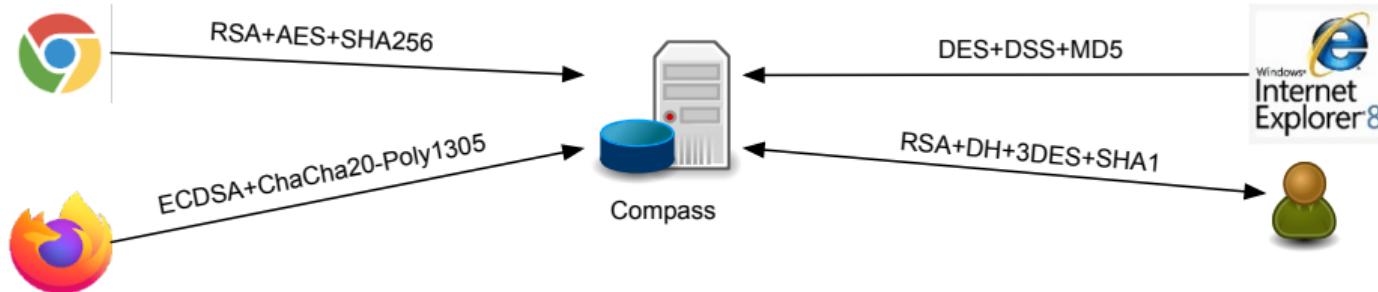
HERAUSFORDERUNGEN FÜR SICHERE KOMMUNIKATIONSPROTOKOLLE

Herausforderungen für standardisierte, sichere Kommunikationsprotokolle

- Geräte und Anforderungen im Internet sind sehr heterogen (Leistung, Bandbreite, Plattform...)
- Einzelschritte der Protokolle müssen sicher zusammengeführt werden
- Einbettung der Protokolle im Netzwerkstack ist komplex

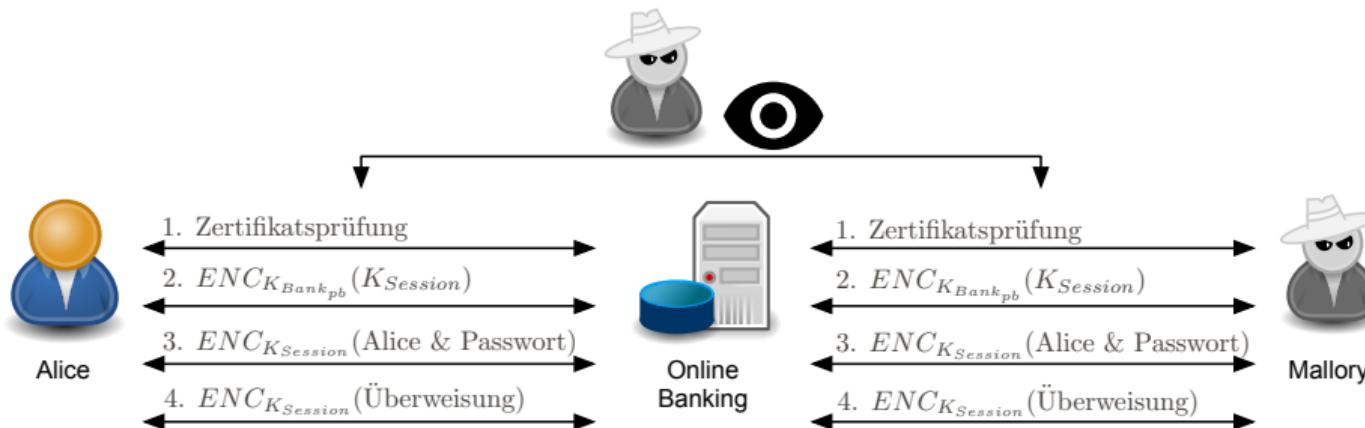
MOTIVATION SICHERE PROTOKOLLE HETEROGENITÄT IM INTERNET

- Geräte im Internet sind sehr heterogen
 - Interoperabilität mit alten Systemen muss gewährleistet sein
 - Unterschiedliche Krypto Verfahren müssen unterstützt werden
 - Manche Anwendungen erfordern Zertifikats-basierte Authentifikation beider Parteien



MOTIVATION SICHERE PROTOKOLLE SICHERE VERBINDUNG EINZELSCHRITTE

- Die Einzelschritte müssen sicher zusammengefügt werden
 - Ansonsten können kleinste Schwachstellen für Angriffe ausgenutzt werden
 - Beispiel: Mallory liest die Nachrichten von Alice und sendet sie erneut (Replay Angriff)



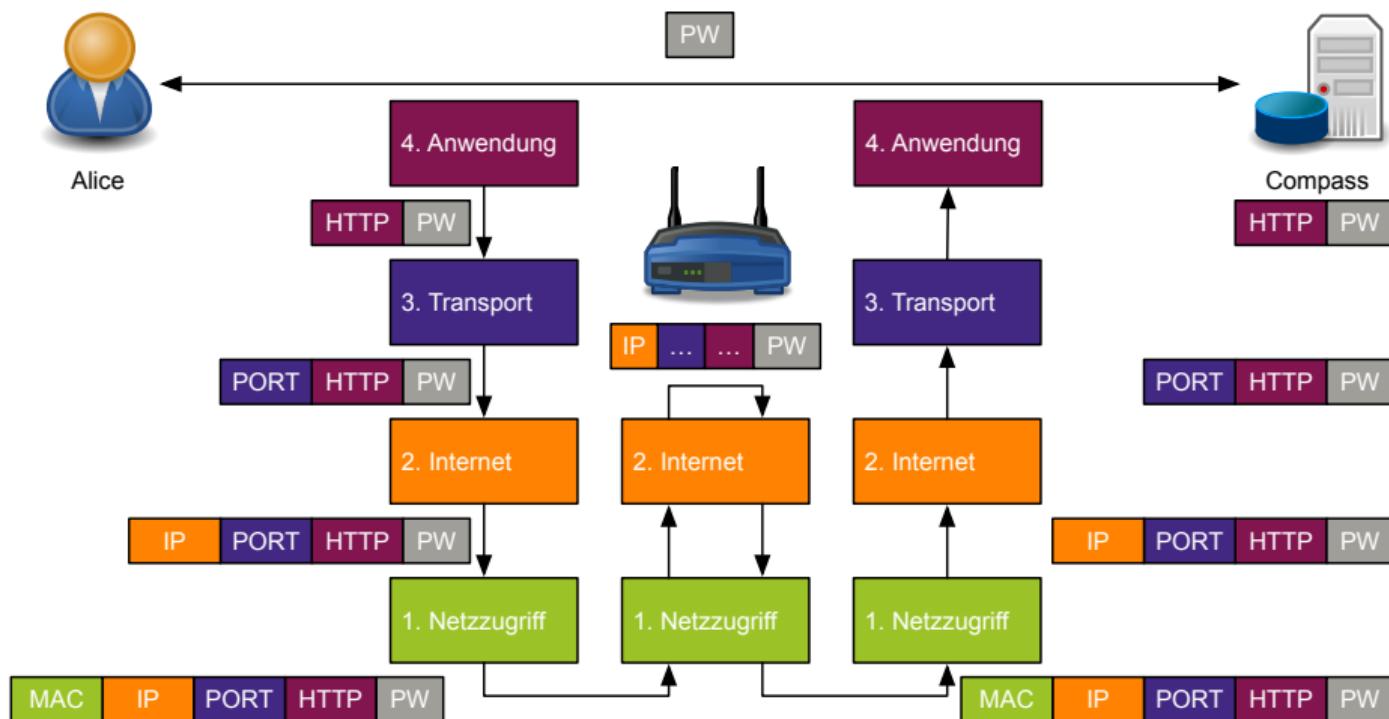
MOTIVATION SICHERE PROTOKOLLE SICHERE VERBINDUNG EINZELSCHRITTE

- Komplexe Vorgänge in der Kommunikationstechnik werden in Schichten eingeteilt
 - OSI Modell
 - TCP/IP Modell
- Schichten werden nacheinander ausgeführt und bieten darüberliegenden Schichten bestimmte Dienste an
 - Transportschicht: Steuerung des Datenflusses
 - Internetschicht: Adressierung von Paketen
 - Netzzugriff: Zugriff auf das Netzwerk



Figure: TC/IP Modell

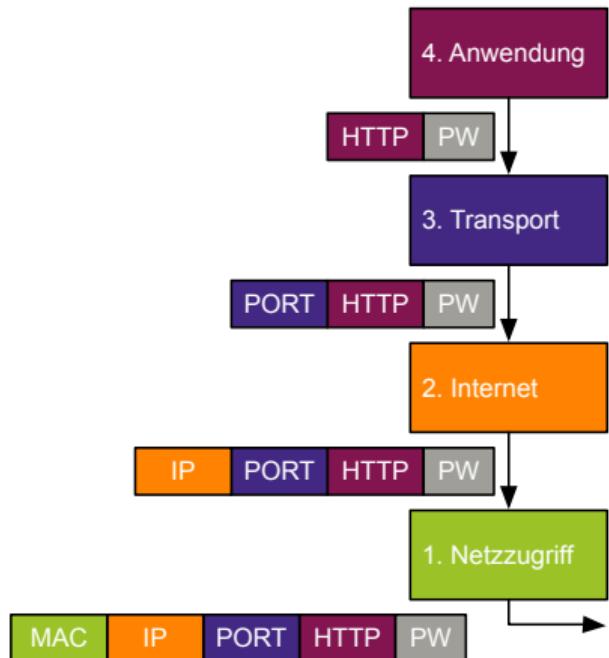
DATENÜBERTRAGUNG IM SCHICHTENMODELL



ABSICHERUNG DER PAKETDATEN

→ **Frage:** In welcher Schicht soll die Absicherung stattfinden?

- Je weiter unten, desto mehr Daten werden abgesichert
- Je weiter oben, desto länger bleiben die Daten abgesichert
- Die sinnvollste Schicht zur Absicherung kann je nach Kontext und Anwendung variieren



VARIANTE DER ABSICHERUNG DER NUTZDATEN

→ Beispiele

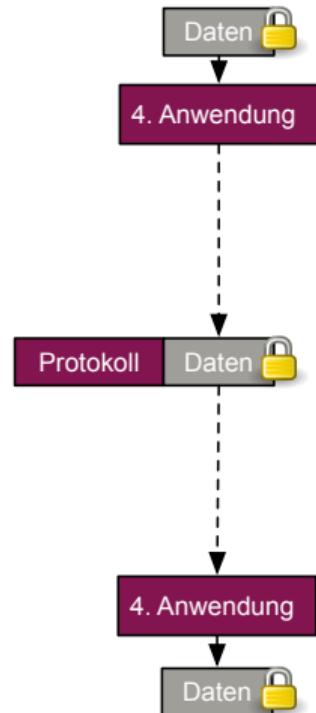
- Chat-Nachrichten oder Dateien absichern
- Kann vor oder in der Anwendungsschicht geschehen

→ Einsatzzwecke

- Mögliche Ende-zu-Ende Verschlüsselung (E2E)
- Der Anwendung wird nicht vertraut (Speicherung in der Cloud)

→ Limitierungen

- Protokoll- und Metadaten sind lesbar (wer sendet Chatnachricht)
- Applikationsspezifische Sicherheitsprotokolle nötig bei Absicherung in der Anwendung



VARIANTE DER ABSICHERUNG NUTZ- UND PROTOKOLLDATEN

→ Beispiele

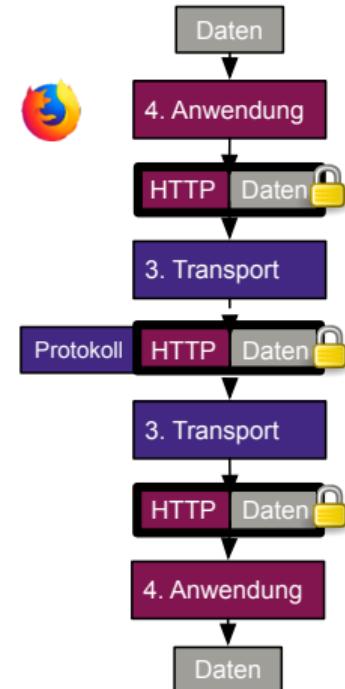
- Webseiten, E-Mails

→ Einsatzzwecke

- Sichere Verbindung zwischen Anwendungen inkl. Protokolldaten

→ Limitierungen

- Port, IP- und MAC Adressen les- und änderbar
- Eine sichere Verbindung je Anwendung wird benötigt
- Code im Kontext der gleichen Anwendung hat Zugriff auf die Nutzdaten (z.B. andere Webseiten)



VARIANTE DER ABSICHERUNG PORT- UND IP ADRESSE

→ Beispiele

- Sicheres Virtual Private Network (VPN)

→ Einsatzzwecke

- Sichere Verbindung zwischen Rechnern
- Absicherung Port: Rechner zu Rechner
- Absicherung IP: Rechner/Netzwerk zu Netzwerk

→ Limitierungen

- MAC Adressen les- und änderbar
- Nutz- und Protokolldaten sind am Ziel ungesichert
- Komplexere Konfiguration



VARIANTE DER ABSICHERUNG MAC ADRESSE

→ **Beispiele**

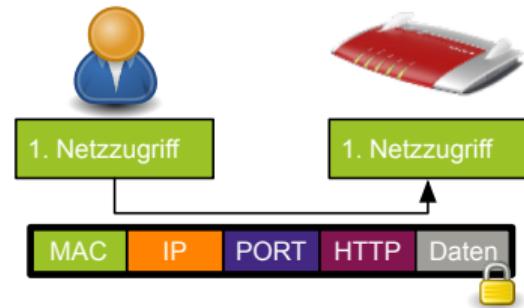
- Sicherer Zugang zum Internet (MACSec, WPA2/3, ...)

→ **Einsatzzwecke**

- Sichere Verbindung zum Router bzw. nächsten Hop
- Absicherung des lokalen Netzwerkes

→ **Limitierungen**

- Schlüssel müssen auf Geräte verteilt werden, damit sie Zugang zum Netzwerk erhalten
- Kommunikation nur abgesichert bis zum Internetzugang



ÜBERSICHT ABSICHERUNG DER PAKETDATEN

→ Welche Paketdaten sollen abgesichert werden?



Abgesicherte Daten	Einsatzzweck	Limitierungen	Protokolle
Nutzdaten	Sichere Ende-zu-Ende (E2E) Kommunikation für Anwendung (z.B. eMail oder WhatsApp)	Protokolldaten lesbar (HTTP GET-/POST), Anwendungsspezifisch	Signal
+ Protokolldaten (z.B. HTTP)	Sichere Verbindung zu einer Anwendung (z.B. Webserver)	Eine sichere Verbindung pro Dienst wird benötigt (z.B. Unternehmens-IT)	Transport Layer Security (TLS)
+ Port und IP	Sichere Verbindung zu einem Host/Netzwerk (z.B. VPN)	Komplexe Netzwerkadministration, Absicherung geht nicht bis zu Anwendung	Internet Protocol Security (IPSec)
+ MAC Adressen	Absicherung des lokalen Netzwerkes (z.B. im Fahrzeug)	Komplexe Netzwerkadministration aufgrund vorher verteilter Schlüssel	WPA2/3, MACsec (MAC Security)

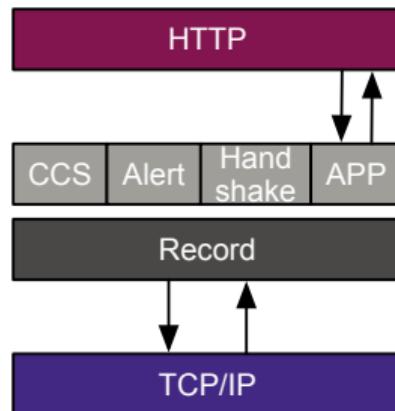
TRANSPORT LAYER SECURITY (TLS – 1/2)

- TLS ist das meist genutzte Protokoll für sichere Kommunikation im Internet
 - Früher bekannt als Security Socket Layer (SSL)
- Browser über HTTPs
- eMail Clients über SMTP/IMAP/POP3
 - Früher bekannt als Security Socket Layer (SSL)
- TLS speichert **Zustandsinformationen** in **Sitzungen**, von denen mehrere gleichzeitig aktiv sein können (z.B. eine Sitzung pro Webseite)

Standard	Nutzungs-zeitraum	Unterstützende Webseiten (Dez22)
SSL1.0	1994 - ?	-
SSL2.0	1995 - 2011	0,2%
SSL3.0	1996 - 2015	2,1%
TLS1.0	1999 - 2021	343,0%
TLS1.1	2006 - 2021	37,0%
TLS1.2	2008+	99,9%
TLS1.3	2018+	58,9%

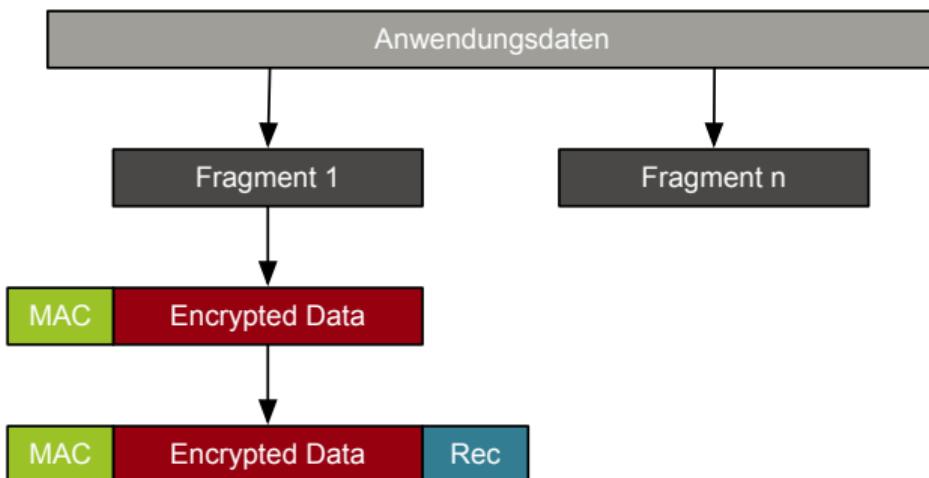
TRANSPORT LAYER SECURITY (TLS – 2/2)

- TLS liegt zwischen Anwendungs- und Transportschicht und besteht aus fünf verschiedenen Protokollen
 1. **Change Cipher Spec (CCS)**: Aushandlung der genutzten Krypto Verfahren
 2. **Alert Protocol**: Fehlerbehandlung und Verbindungsabbruch
 3. **Handshake**: Aushandlung der Sitzungsinformationen und des Sitzungsschlüssels
 4. **Application**: Transparente Kommunikation mit Anwendung
 5. **Record Layer**: Teilt Daten in Fragmente und sorgt für deren Absicherung



TLS - RECORD LAYER PROTOKOLL

- Das Record Layer Protokoll fragmentiert Anwendungsdaten transparent und nutzt symmetrische Kryptographie, um die Sicherheit der Daten zu gewährleisten



Der TLS Record enthält:

- Typ des überliegenden Protokolls (CCS, Alert, Handshake, Applikation)
- TLS Versionsinformationen
- Länge der Nutzdaten

TLS - HANDSHAKE PROTOKOLL

- Pro Sitzung müssen verschiedene Informationen ausgehandelt werden
 - Verwendete kryptographische Verfahren
 - Wer muss sich authentifizieren? (Keiner, nur Server, Alice und Server)
 - Symmetrischer Schlüssel für Record Layer Protokoll (sog. Sitzungsschlüssel)
- Kryptographische Verfahren werden mittels der Cipher Suite ausgehandelt
 - Beispiel: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- Eine Cipher-Suite definiert
 - **Schlüsselaustausch** (Diffie-Hellman mit Schlüssellösung – sog. DH Ephemera)
 - **Authentifizierung** (RSA Signaturen)
 - **Verschlüsselung** (AES-128 GCM)
 - **Hashfunktion** (SHA256)

HANDSHAKE PROTOKOLL TLS1.2 RSA SIGNATUR UND DH SCHLÜSSELAUSTAUSCH



Alice

Prüfe das Zertifikat
Verifizierte Signatur

RSA: $B \stackrel{?}{=} Verify_{K_E}(S)$

Wähle Client Schlüssel

DH: $A = g^a \mod p$

Sitzungsschlüssel
berechnen

Bekannt: Öffentliche DH-Primzahl p und Basis g

Client Hello:(Nonce_C, Liste Cipher Suites)

TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_PSK_WITH_AES_256_GCM_SHA384
TLS_PSK_WITH_CHACHA20_POLY1305_SHA256



Compass

DH: $B = g^b \mod p$

Wähle Server Schlüssel
Signiere Server Schlüssel

RSA: $S = Sing_{K_D}(B)$

Enthält K_E , Domain-Name, Gültigkeit

TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

Sever Hello:(Zertifikat, Nonce_S, gewählte Cipher Suite, Server Schlüssel, Signatur)

ClientKeyExchange: (Client Schlüssel)

$K = \text{HMAC-SHA256}(g^{ab}, \text{Nonce}_C, \text{Nonce}_S)$

Sitzungsschlüssel
berechnen

Sicherer Kanal mit AES_GCM_K

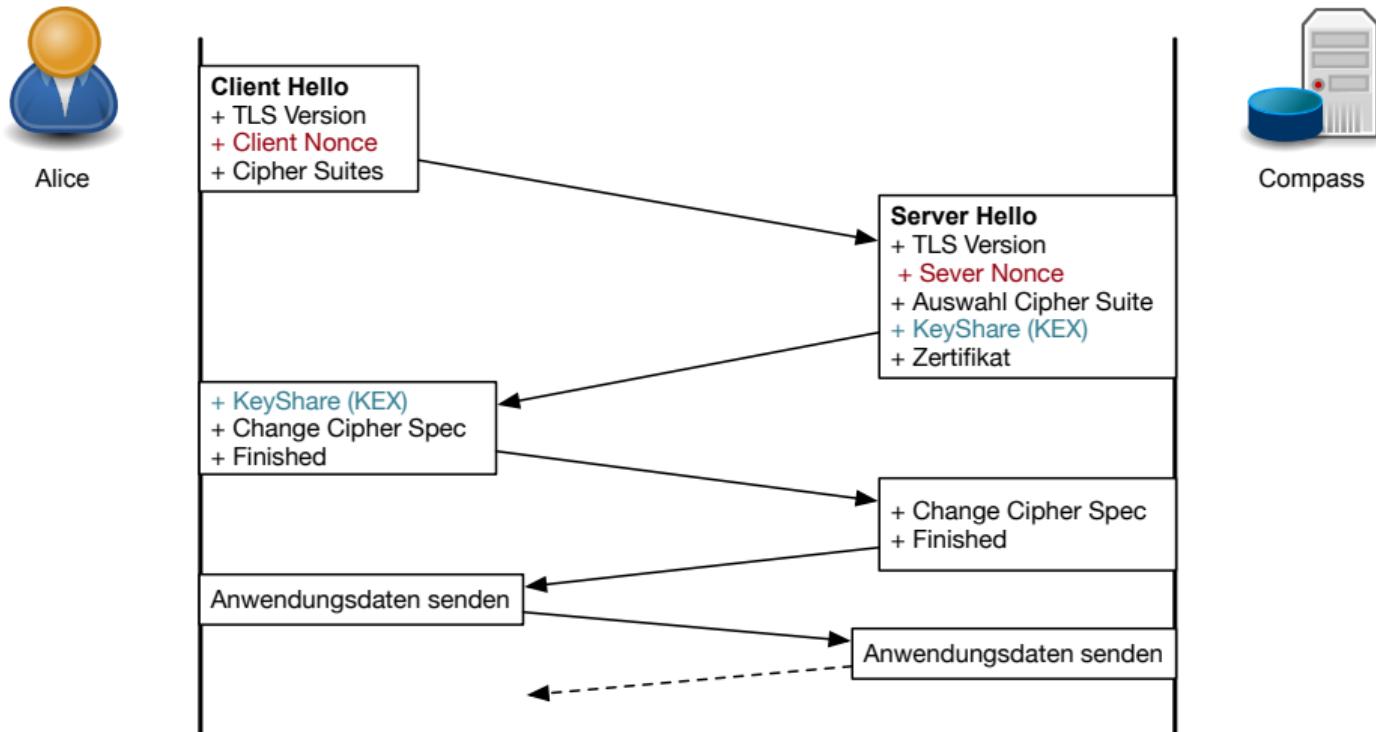


DISKUSSION IN KLEINEN GRUPPEN

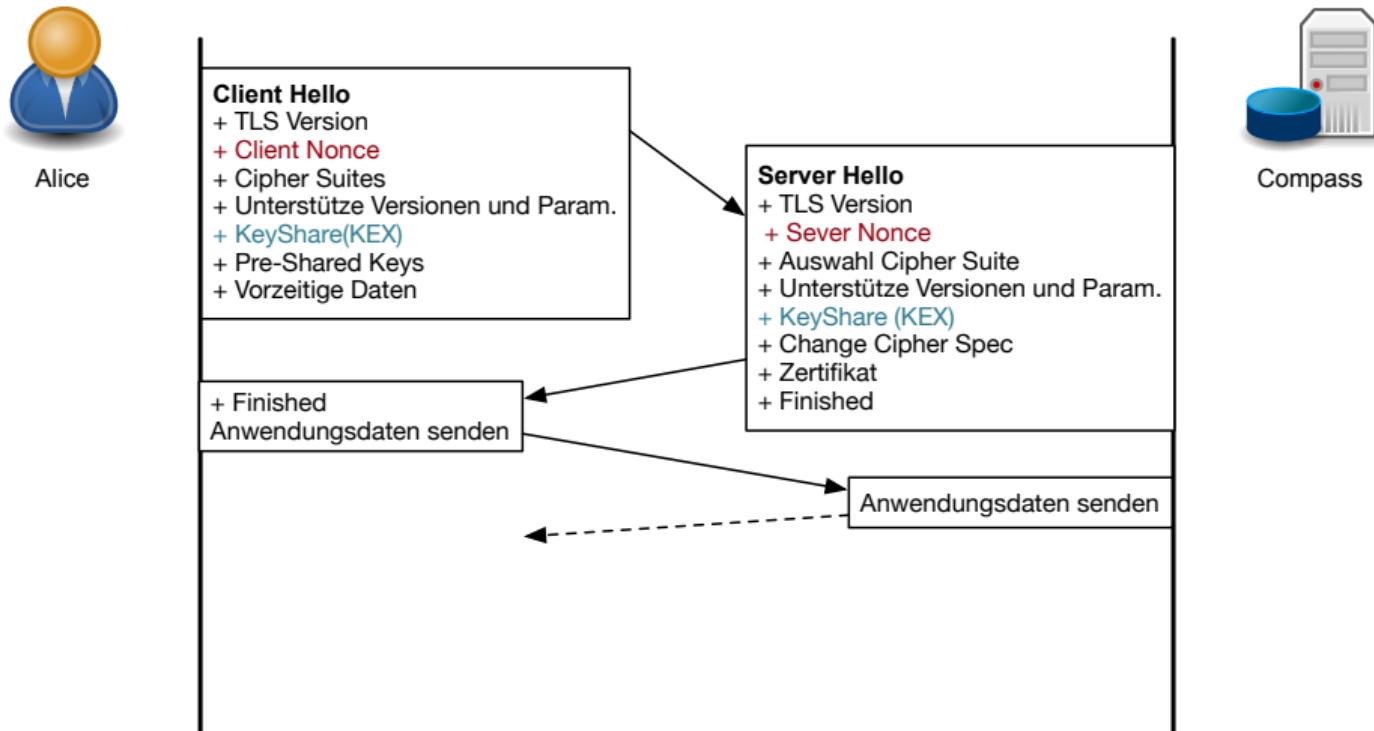
Sicherheit von TLS

- Wie wird verhindert, dass ein Angreifer die Nachrichten einer alten Sitzung einspielt, um einen Replay-Angriff durchzuführen?
- Wieso muss der Server Schlüssel B signiert werden?
- Wieso kann ein Angreifer den Sitzungsschlüssel K nicht berechnen?

HANDSHAKE PROTOKOLL TLS1.2



HANDSHAKE PROTOKOLL TLS1.3



TLS1.3 VS. TLS1.2

Hauptunterschied zwischen TLS1.3 [RFC8446] und TLS 1.2 [RFC5246]:

- Unsichere veraltete Verfahren wurden rausgenommen
 - Die Cipher Suite wurde auf fünf Sets reduziert
 - Kein statischer Schlüsselaustausch erlaubt
 - Schlüsselaustauschverfahren nur noch mit (EC)DHE, PSK-only und PSK mit (EC)DHE
- Verschlüsselung der Kommunikation nach Handshake Nachricht ServerHello
- Kryptographische Verfahren basieren auf Elliptischen Kurven und gehören zum Basisset
- Reduktion des Handshake-Protokolls zum schnelleren Aufbau des gesicherten Kommunikationskanals

SICHERHEITSPROBLEME BEI TLS

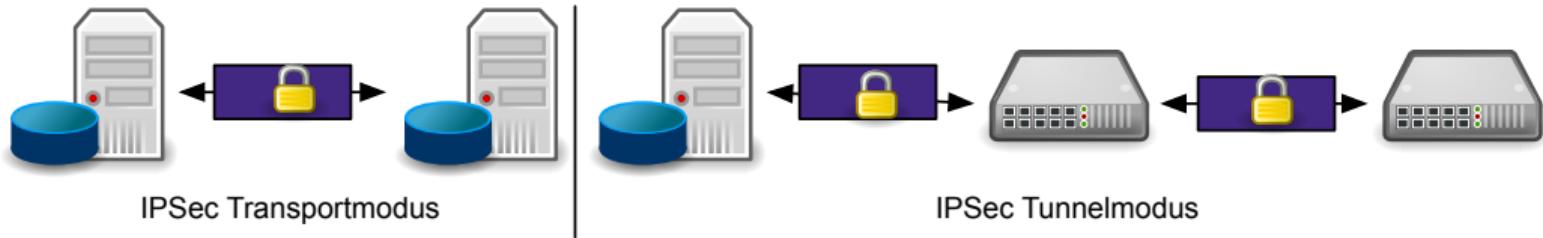
- Viele bekannte Angriffe gegen alte TLS / SSL Versionen
 - Bleichenbacher (\leq SSL3.0): Angriffe auf RSA Padding Verfahren [Bleichenbacher]
 - Beast (\leq TLS1.2): Angriff auf Cipher-Block-Chaining (CBC) Initialisierungsvektor [BEAST]
 - Poodle (\leq TLS1.0): Angriff auf Padding Verfahren in CBC [POODLE]
- Häufiger Angriffsvektor Downgrade: Angreifer bringt Opfer und Server dazu, eine alte TLS Version oder anfällige Cipher-Suite zu nutzen [Logjam]
 - Gegenmaßnahme: Abschalten alter TLS Versionen und Cipher-Suites
- Implementierungsfehler in TLS Bibliotheken
 - Heartbleed: Softwarefehler, der Auslesen zufälliger Bereiche im RAM ermöglichte [HB]

INTERNET PROTOCOL SECURITY (IPSEC)

- IPSec ist eine Familie von Protokollen, zur sicheren Kommunikation, die auf der Internetschicht arbeiten
 - **Internet Key Exchange (IKE)**: Protokoll zum Schlüsselaustausch und Überprüfung der Authentizität der Endgeräte
 - **Authentication Header (AH)**: Authentizität und Integrität der Kommunikation
 - **Encapsulation Security Payload (ESP)**: Vertraulichkeit, Authentizität und Integrität der Kommunikation

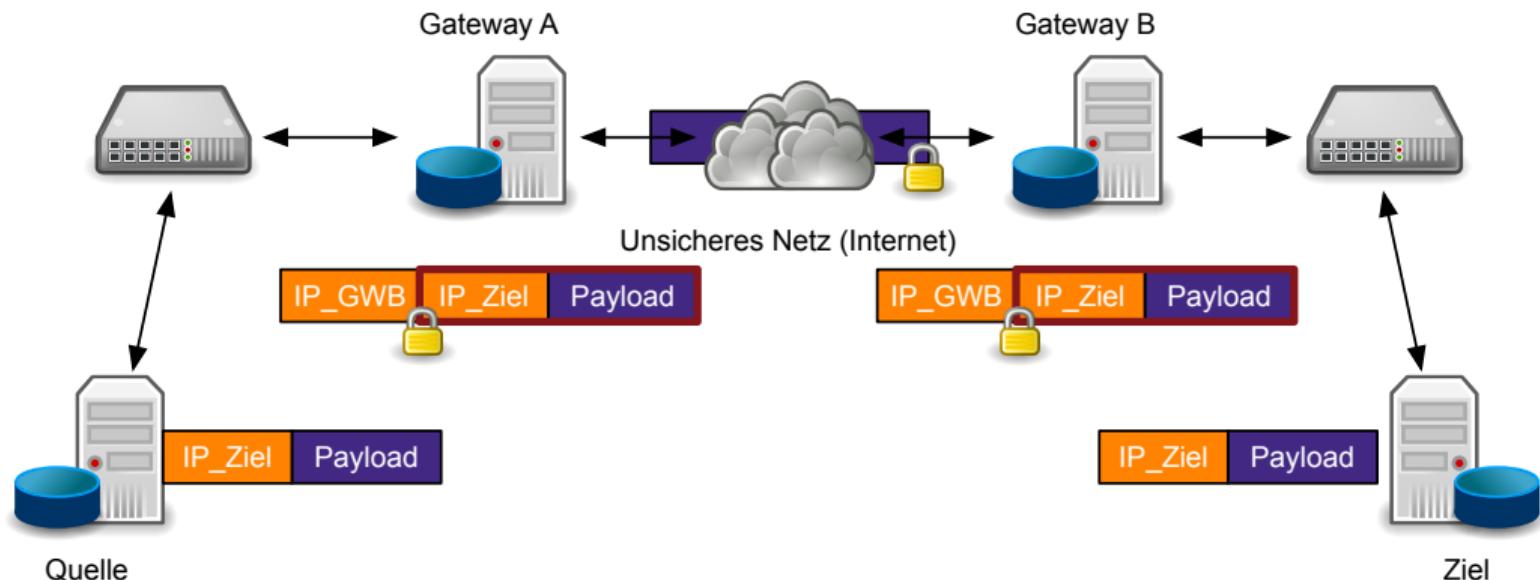
INTERNET PROTOCOL SECURITY (IPSEC)

- IPSec unterstützt zwei verschiedene Modi
 - **Transportmodus**: Sichere Verbindung zweier Geräte
 - **Tunnelmodus**: Sichere Verbindung in Netzwerke (Virtual Private Network - VPN)
- AH und ESP unterscheiden sich je nachdem, ob sie für den Transport- oder Tunnelmodus eingesetzt werden



ESP IM TUNNELMODUS ARCHITEKTUR

ESP im Tunnelmodus verschlüsselt die Ziel IP und den Payload und leitet das Paket an das Ziel Gateway weiter



ESP IM TRANSPORT- UND TUNNELMODUS

Paket abgesichert im
Transportmodus

Verschlüsselt und Authentisch

Authentisch
Ungesichert



Original Paket
auf Schicht 2

Paket abgesichert im
Tunnelmodus

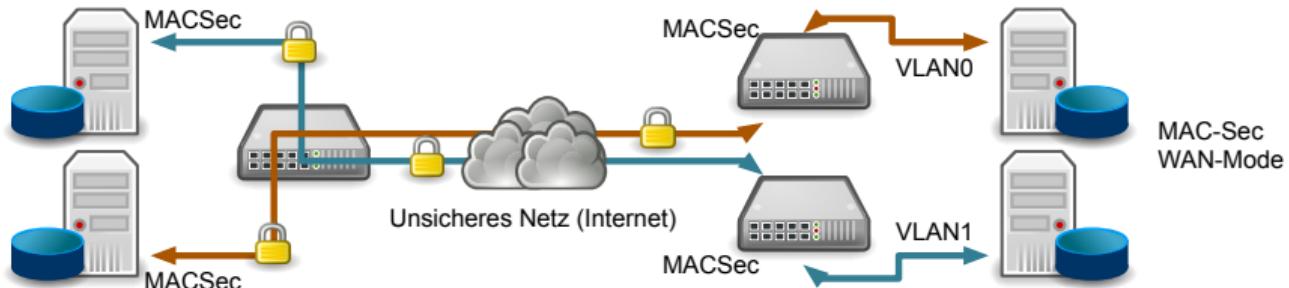
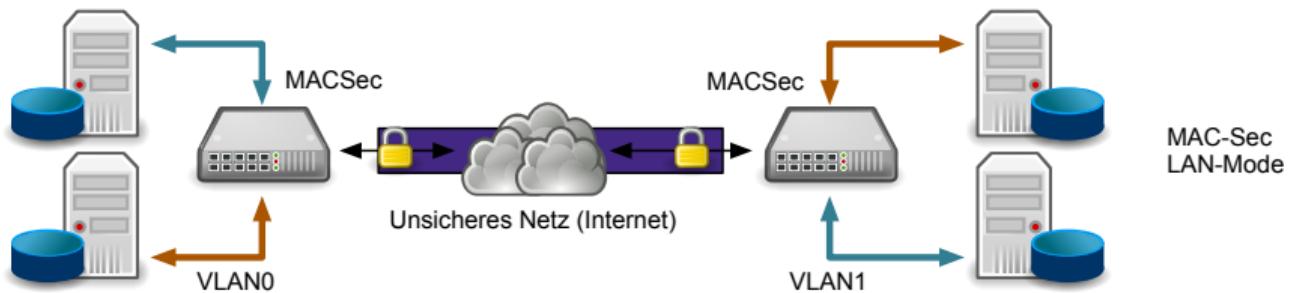


MEDIA ACCESS CONTROL SECURITY (MACSEC)

- MACSec ist in IEEE 802.AE standardisiert zur sicheren Kommunikation, die auf der Netzzugriffsschicht arbeitet
 - MACSec basiert auf einem Standard Ethernet Frame und wird um zwei Felder erweitert
 - **MACsec Security Tag (SecTAG)**: Kontrollfeld mit Konfigurationsinformationen
 - **Integrity Check Value (ICV)**: Authenzitätstoken 16 Byte
 - Der MACSec Frame kann mit AES-GCM gesichert werden und somit verschlüsselt und authentisch sein
 - Schlüssel für die Absicherungen können statisch vorab geteilt werden (PSK) oder über einen Schlüsselserver mit Authentisierungsprotokollen (EAP) via IEEE 802.1X

MACSEC MODUS

MACSec hat zwei Betriebsmodi, welche sich auf die Nutzung von VLANs auswirken.



MACSEC PAKETSTRUKTUR

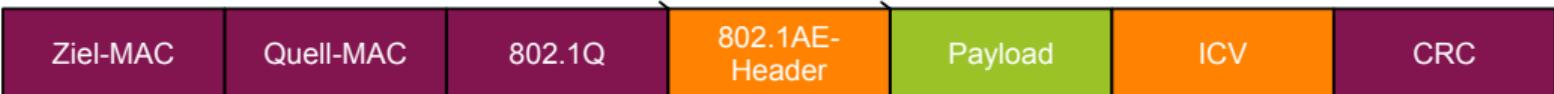


TCI/AN: Konfiguration ob verschlüsselt und/oder authentisch
 SL: Größe der verschlüsselten Daten
 PacketNumber: Packetnummer
 SCI: Secure Channel ID - Virtueller Port

MACSec LAN-Mode



MACSec WAN-Mode



ZUSAMMENFASSUNG

- Herausforderungen für Protokolle zur sicheren Kommunikation
- Einsatzzwecke und Limitierungen bei der Absicherung in verschiedenen TCP/IP Schichten
- Sichere Kommunikationsprotokolle der verschiedenen TCP/IP Schichten
- Grobe Funktionsweise von TLS, um einen sicheren Kommunikationskanal zu etablieren
- IPSec Protokollfamilie sowie den Tunnel- und Transportmodus
- Konzept hinter dem IPSec Tunnelmodus
- Konzept hinter MACSec



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Softwaresicherheit

June 23, 2023

Marc Stöttinger

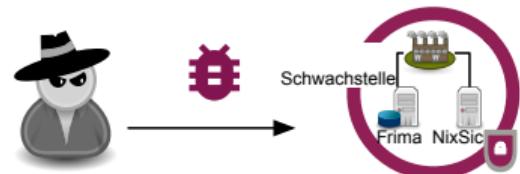


Security is not a product, but a process. It's about building a culture of secure coding, continuous testing, and proactive vulnerability management throughout the software development lifecycle.

Bruce Schneier

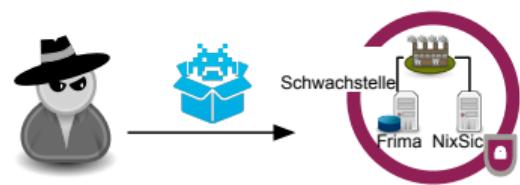
VORGEHEN EINES ANGREIFERS

1. Analyse des Ziels auf Software **Schwachstellen**



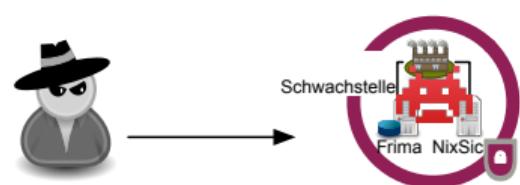
2. Ausnutzung der Schwachstellen durch **Exploits**

→ Schwachstellen und Exploits bedingen sich gegenseitig



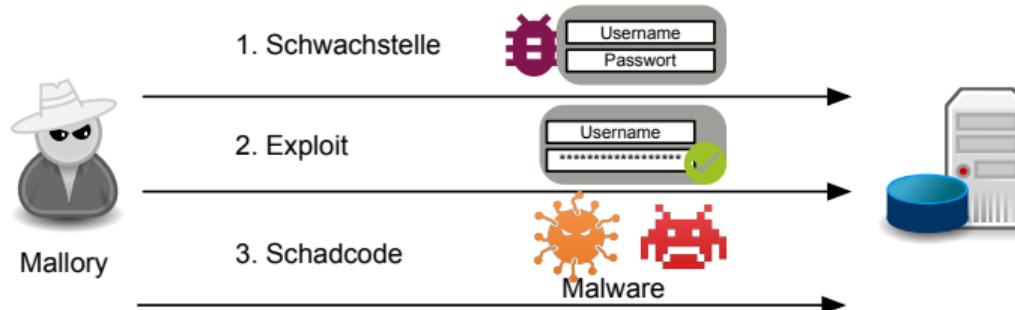
3. **Schadhafte Aktionen**

→ Malware kann Exploits beinhalten, um sich selbstständig zu verbreiten



BEISPIEL ANGRIFF [XBOX]

1. **Schwachstelle**: Login ohne Passwort mittels langem String möglich
2. **Exploit**: Angreifer loggt sich mittels langem String ein
3. **Schadcode**: Angreifer installiert Programm, das Daten verschlüsselt



MELDUNG VON SCHWACHSTELLEN

Entdeckte Schwachstellen und zusätzliche Informationen werden via verschiedener Schwachstellendatenbanken publiziert

- Bekannteste Datenbank: „Common Vulnerabilities and Exposures (**CVE**)“
- Datenbank ermöglicht gezielte Suche. z.B. Schwachstellen bestimmter Programme

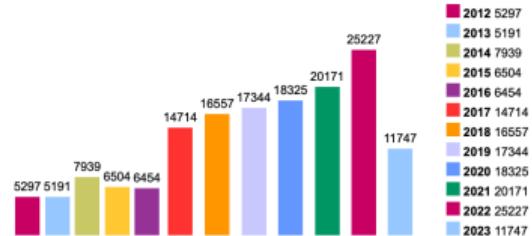
Suchergebnisse nach Schwachstellen

Search Results

There are **641** CVE Records that match your search.

Name	Description
CVE-2023-31127	libspdm is a sample implementation that follows the DMTF SPDM specifications. A vulnerability has been identified in SPDM session establishment in libspdm prior to version 2.3.1. If a device supports both DHE session and PSK session with mutual authentication, the attacker may be able to establish the session with 'KEY_EXCHANGE' and 'PSK_FINISH' to bypass the mutual authentication. This is most likely to happen when the Requester begins a session using one method (DHE, for example) and then uses the other method's finish (PSK_FINISH in this example) to establish the session. The session hashes would be expected to fail in this case, but the condition was not detected. This issue only impacts the SPDM responder, which supports 'KEY_EX_CAP=1 and 'PSK_CAP=10b' at same time with mutual authentication requirement. The SPDM requester is not impacted. The SPDM responder is not impacted if 'KEY_EX_CAP=0' or 'PSK_CAP=0' or 'PSK_CAP=1b'. The SPDM responder is not impacted if mutual authentication is not required. libspdm 1.0, 2.0, 2.1, 2.2, 2.3 are all impacted. Older branches are not maintained, but users of the 2.3 branch may receive a patch in version 2.3.2. The SPDM specification (DSP0274) does not contain this vulnerability.
CVE-2023-28238	Windows Internet Key Exchange (IKE) Protocol Extensions Remote Code Execution Vulnerability
CVE-2023-24859	Windows Internet Key Exchange (IKE) Extension Denial of Service Vulnerability

CVE Meldungen pro Jahr



Quelle: <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Exchange> und <https://www.cvedetails.com/browse-by-date.php>

EFFEKTE VON SCHWACHSTELLEN UND EXPLOITS

- Schwachstellen können verschiedene **Effekte** zulassen:
 - Veränderung der Funktionsweise des Systems (z.B. Preise reduzieren)
 - Auslesen geheimer Informationen (z.B. Ausgabe aller Daten einer Datenbank)
 - Störung der Verfügbarkeit des Systems (z.B. Absturz eines Programms)
 - Ausführen eigener Programme auf dem Zielsystem (z.B. Ausführen von Malware)
- Stark variierende **Voraussetzungen zur Ausnutzung** einer Schwachstelle
 - Automatisiert über Internet auf alle Installationen eines Programms
 - Lokal am PC mit Adminrechten bei spezieller Konfiguration
- Priorisierung der Schwachstellen ist notwendig, wenn viele gleichzeitig anfallen

BEWERTUNG VON SOFTWARE SCHWACHSTELLEN

- der Basic Score vom Common Vulnerability Scoring System (**CVSS**) bewertet Software Schwachstellen basierend auf
 - Komplexität der Ausnutzung
 - Auswirkungen auf Vertraulichkeit, Integrität und Verfügbarkeit
- Die komplette Bewertung im [CVSS] berücksichtigt den Basic Score, einen Temporal und einen Environmental Score.

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	23965	11.70
1-2	1198	0.60
2-3	8337	4.10
3-4	9494	4.70
4-5	42988	21.10
5-6	34098	16.70
6-7	27167	13.30
7-8	35998	17.60
8-9	898	0.40
9-10	19973	9.80
Total	204116	

Quelle: <https://www.cvedetails.com/>

[cvss-score-distribution.php](#)

- Komplexität der Ausnutzung wird angegeben via:
 - **Angriff möglich aus**: Internet, LAN, System Zugang, Physischer Zugang
 - **Angriffskomplexität**: keine Bedingungen, Wissen / Zugang benötigt
 - **Benötigte Rechte**: Keine, User, Admin
 - **Opfer Interaktion nötig**: Keine, Interaktion benötigt
 - **Reichweite**: Beschränkt auf anfällige Komponente, Sprung aus Komponente möglich

CVSS BEISPIELE

- **Log4j**: Nachladen und Ausführen von Programmen möglich, wenn ein bestimmter String an Anwendung mit Log4j Framework gesendet wird (z.B. Apache Webserver oder Minecraft Server)
- **WhatsApp**: Interaktion mit WhatsApp via Siri möglich, selbst bei aktivem Sperrbildschirm auf iPhone

Kriterium	Log4j: CVE- 2021-44228	WhatsApp: CVE- 2020-1908
Angriff möglich aus	Netzwerk	Physikalischer Zugang
Angriffskomplexität	Niedrig	Niedrig
Benötigte Rechte	Keine	Keine
Opfer Interaktion nötig	Keine	Keine
Reichweite	Sprung möglich	Kein Sprung möglich
Vertraulichkeit	Hoch	-
Integrität	Hoch	Hoch
Verfügbarkeit	Hoch	-
Gesamtscore	10,0	4,6

HÄUFIG AUFTRETENDE SCHWACHSTELLEN

Häufig auftretende Schwachstellen werden analysiert und publiziert

- **Generelle Schwachstellen:** Common Weaknesses Enumeration [CWE]
- **Schwachstellen in Webanwendungen:** Open Web Application Security Project [OWASP]

Platz	Schwachstellenbeschreibung	Häufigkeit in 2022	Mittl. CVSS 2022
1	Schreiben außerhalb des festgelegten Bereichs	4123	7,93
2	Cross-Site Scripting	4740	5,73
3	SQL Injection	1263	8,66
4	Unzureichende Eingabeverifikation	1520	7,19
5	Lesen außerhalb des festgelegten Bereichs	1489	6,54

HÄUFIGE SOFTWAREFEHLER

- Zugriffe außerhalb eines festgelegten Bereiches beinhalten häufig die Fehler
 - Buffer Overflow [Buf] und
 - Fehlerhafte Integer Behandlung [Calc]
- Fokus auf die Programmiersprache C, aufgrund weiter Verbreitung und einfacher Erklärung
- Viele Fehlerquellen in C werden in moderneren Sprachen vermieden, z.B.:
 - **Java**: Laufzeitprüfung und Exception Handling (z.B. ArrayIndexOutOfBoundsException)
 - **Rust**: Kapazitätsprüfung der Puffer vor Zugriff und Ownership Mechanik

C BASICS

- Primitive Datentypen können unterschiedliche Wertebereiche darstellen

Daten-typ	Länge (Bit)	Wertebereich (unsigned)	Wertebereich (signed)
char	8	-128 bis 127	0 bis 255
short	16	-32.769 bis 32.767	0 bis 65.535
int	32	-2.147.483.648 bis 2.147.483.647	0 bis 4.294.967.295
long	64	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807	0 bis 18.446.744.073.709.551.615

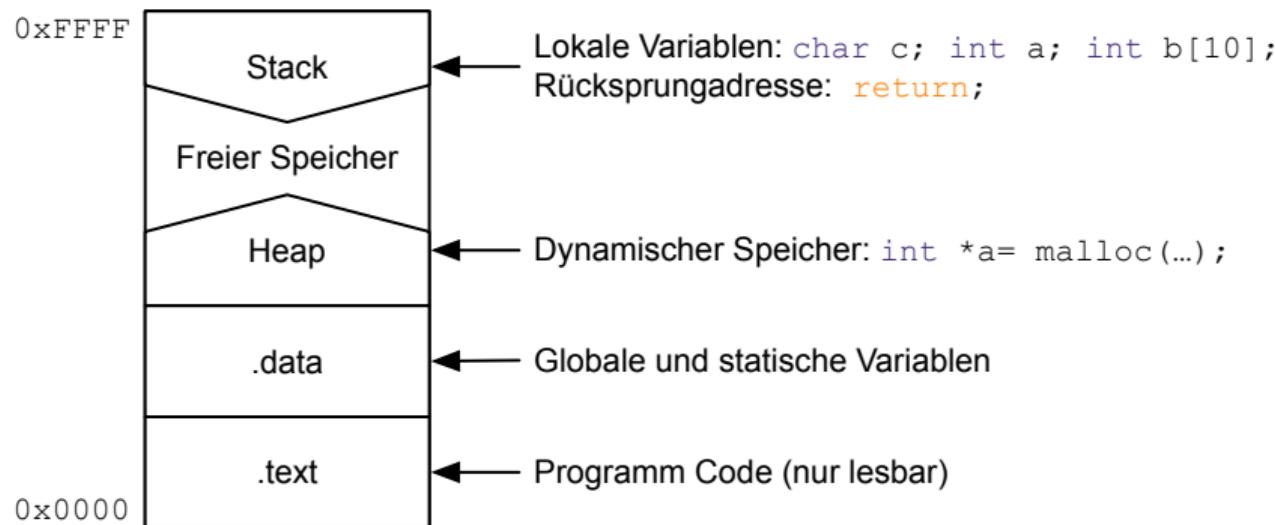
Wertebereiche für eine 32-Bit Maschine

- Direkte Operation auf Speicher möglich

- Reservierung: `malloc(len)` – Reserviere einen Speicherbereich von len Bytes
- Kopieren: `memcpy(Ziel, Quelle, len)` – Kopiere len Bytes von Quelle nach Ziel

SPEICHERLAYOUT EINES C-PROGRAMMS

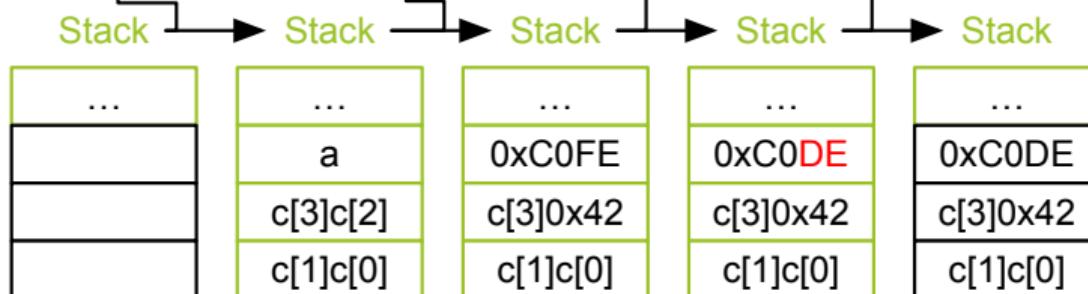
Die Speicherorganisation eines C-Programms ist wie folgt aufgebaut:



STACK- UND HEAP BUFFER OVERFLOWS

```
void foo() {  
    short a;  
    char c[4];  
  
    a = 0xC0FE;  
    c[2] = 0x42;  
  
    c[5] = 0xDE;  
  
    return;  
}
```

short a;
char c[4];
a = 0xC0FE;
c[2] = 0x42;
c[5] = 0xDE;
return;



Stack Buffer Overflow

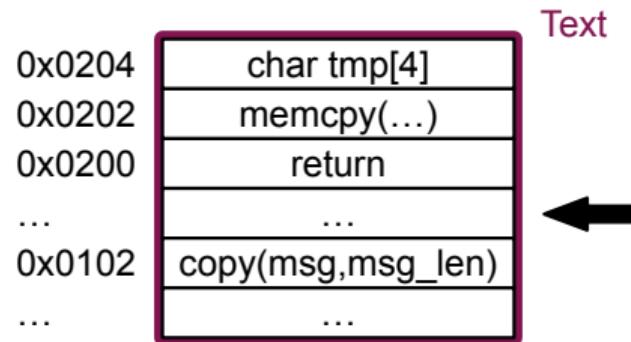
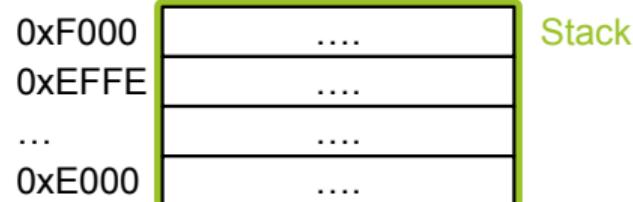
→ Heap Buffer Overflow: Analog für dynamisch allozierten Speicher im Heap

AUSNUTZUNG VON BUFFER OVERFLOWS

- Bei Buffer Overflows werden Daten außerhalb des festgelegten Bereiches geschrieben
- Wie können Buffer Overflows ausgenutzt werden?
 - Programmabstürze aufgrund inkonsistenter Daten (z.B. falsche Pointer)
 - Gezielte Modifikation der Daten im Speicher (z.B. Änderung eines Preises)
 - Ausführung wahlfreien Codes (z.B. Ausführung der Eingabedaten)
- **Beispiel im Folgenden:** Ausführung wahlfreien Codes

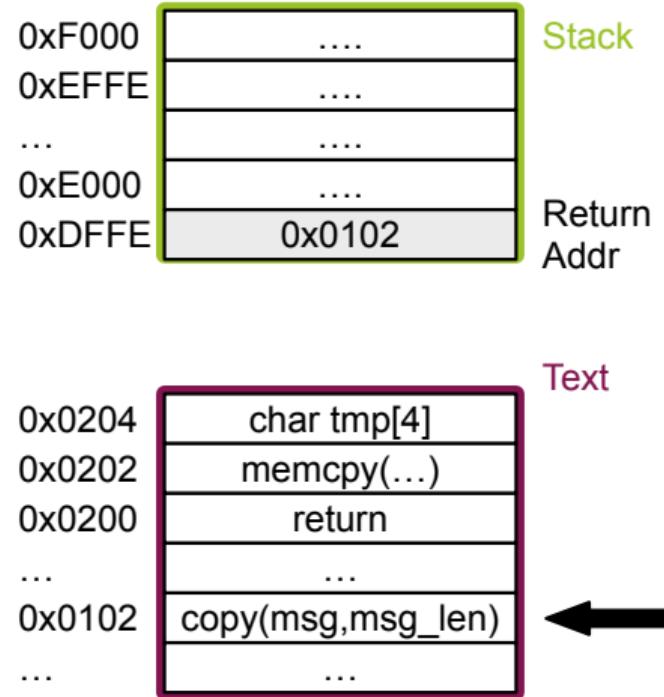
STACK BUFFER OVERFLOW AUSNUTZUNG I

```
char msg[] = {0xAA, 0xBB, 0xCC, 0xDD,  
              0x00, 0xF0, "Schadcode",  
              "Ausführen"};  
  
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg, msg_len);  
    return;  
}  
  
int main(){  
    ...  
    copy(msg, sizeof(msg));  
    ...  
}
```



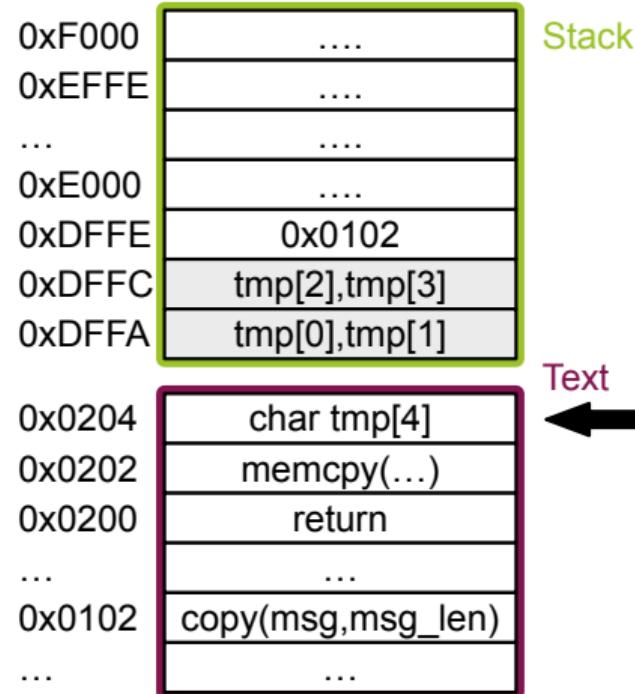
STACK BUFFER OVERFLOW AUSNUTZUNG II

```
char msg[] = {0xAA, 0xBB, 0xCC, 0xDD,  
              0x00, 0xF0, "Schadcode",  
              "Ausführen"};  
  
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg, msg_len);  
    return;  
}  
  
int main(){  
    ...  
    copy(msg, sizeof(msg));  
    ...  
}
```



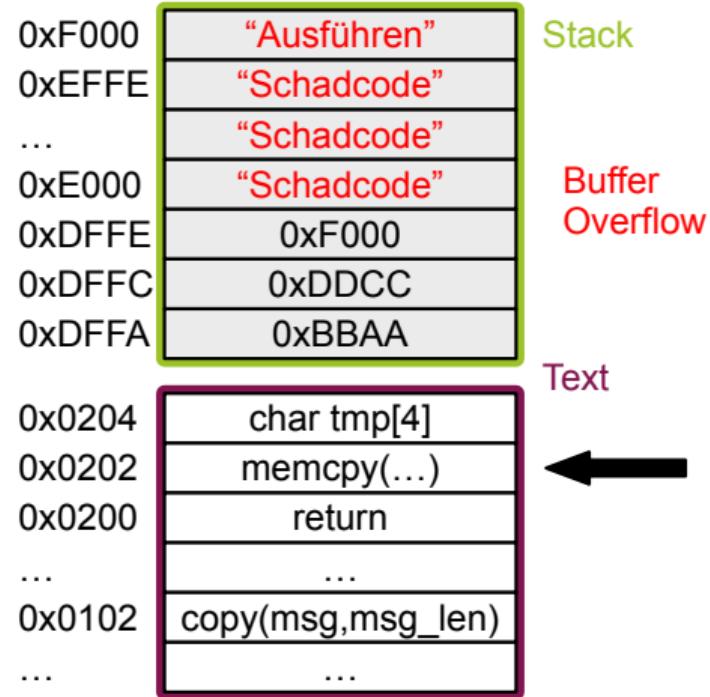
STACK BUFFER OVERFLOW AUSNUTZUNG III

```
char msg[] = {0xAA,0xBB,0xCC,0xDD,  
              0x00,0xF0,"Schadcode",  
              "Ausführen"};  
  
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg,msg_len);  
    return;  
}  
  
int main(){  
...  
    copy(msg, sizeof(msg));  
...  
}
```



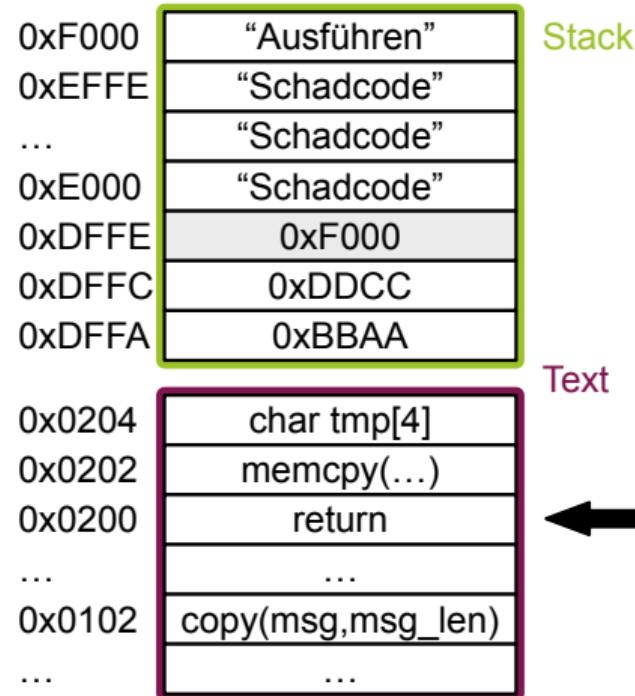
STACK BUFFER OVERFLOW AUSNUTZUNG IV

```
char msg[] = {0xAA, 0xBB, 0xCC, 0xDD,  
              0x00, 0xF0, "Schadcode",  
              "Ausführen"};  
  
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg, msg_len);  
    return;  
}  
  
int main(){  
...  
    copy(msg, sizeof(msg));  
...  
}
```



STACK BUFFER OVERFLOW AUSNUTZUNG V

```
char msg[] = {0xAA, 0xBB, 0xCC, 0xDD,  
              0x00, 0xF0, "Schadcode",  
              "Ausführen"};  
  
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg, msg_len);  
    return;  
}  
  
int main(){  
...  
    copy(msg, sizeof(msg));  
...  
}
```



STACK BUFFER OVERFLOW AUSNUTZUNG VI

```
char msg[] = {0xAA, 0xBB, 0xCC, 0xDD,  
              0x00, 0xF0, "Schadcode",  
              "Ausführen"};  
  
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg, msg_len);  
    return;  
}  
  
int main(){  
...  
    copy(msg, sizeof(msg));  
...  
}
```

0xF000	"Ausführen"
0xEFFE	"Schadcode"
...	
0xE000	"Schadcode"
0xDFFE	"Schadcode"
0xDFFC	0xF000
0xDFFA	0xDDCC
	0xBAAA

Stack

0x0204	char tmp[4]
0x0202	memcpy(...)
0x0200	return
...	...
0x0102	copy(msg,msg_len)
...	...

Text

AUSLÖSUNG EINES BUFFER OVERFLOWS

- Einfache Vermeidung von Buffer Overflows: Prüfe zu kopierende Menge an Daten gegen Puffergröße

```
void copy(char *msg, int msg_len){  
    char tmp[4];  
    memcpy (tmp, msg,msg_len);  
    return;  
}
```



```
void copy(char *msg, int msg_len){  
    char tmp[msg_len];  
    memcpy (tmp, msg,msg_len);  
    return;  
}
```

- Häufig werden mehrere Fehler mit dem Buffer Overflow ausgenutzt
 - Hafnium Hack zu Microsoft Exchange: 4 Schwachstellen [HAF]

DISKUSSION IN KLEINEN GRUPPEN

In welchen Beispiel wird ein Buffer Overflow ausgelöst und warum?

Beispiel 1

```
int main() {
    int num = receive(4);
    int* buf = (int*) malloc(num * 4);
    for (int i = 0; i <= num; i++)
        buf[i] = receive(4);

    return 0;
}
```

Beispiel 3

```
int main() {
    short buf_bytes = 1000;
    char* buf = (char*) malloc(buf_bytes);
    int bytes = receive(4);
    char* source = receive(bytes);

    if((short) bytes > buf_bytes)
        return 1;

    memcpy(buf, source, bytes);
    return 0;
}
```

Beispiel 2

```
int main() {
    int bytes = receive(4);
    char *source = receive(bytes);
    int obj_size = 1000;
    int num_obj = bytes / obj_size;

    char* obj_buf = (char*) malloc(num_obj * obj_size);

    memcpy(obj_buf, source, bytes);
    return 0;
}
```

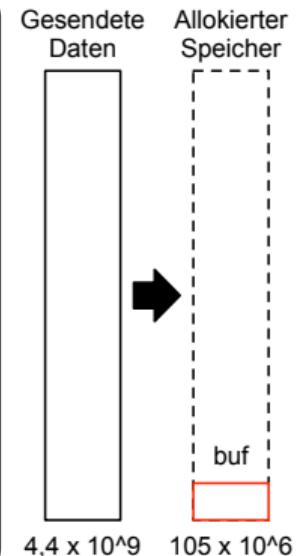
Hinweis:
Die Funktion `receive(int n)`
gibt `n` Bytes zurück.

INTEGER WRAP AROUND/OVERFLOW

- Primitive Datentypen besitzen begrenzten Zahlenraum
 - Verlassen des Zahlenwertes muss überprüft und abgefangen werden

```
//Empfange einen Vektor von Integer Werten
bool receive_integer_vector() {
    //Anzahl der Integer Werte die empfangen werden müssen (als 4 Byte Wert)
    unsigned int num_values = receive(4);
    //Alloziere Puffer richtiger Größe
    int* buf = (int*) malloc(num_values * 4);
    //Prüfe ob ein Puffer alloziert wurde
    if(buf == NULL)
        return false;
    //Empfange jeden Integer Wert (4 Byte) und speichere ihn im Puffer
    for(unsigned int i = 0; i < num_values; i++)
        buf[i] = receive(4);
    free(buf);
    return true;
}
```

-1 < unsigned int < 4.294.967.296
 1.100.000.000
 4.400.000.000 > 4.294.967.295
 Differenz= 105.032.705
 Ab I = 105.032.705/4= 262.258.176
 Buffer Overflow



FEHLINTERPRETATION BEI INTEGER TYPECASTING

- Beim Arbeiten mit verschiedenen primitiven Datentypen können Werte fehlinterpretiert werden
 - Verlust von Informationen beim expliziten Typecasting auf einen kleineren Typen
 - Kontraintuitive Interpretation beim impliziten Typecasting

Typecasting auf kleineren Typen

```
int main() {
    short a = 42;
    int b = -65494;

    if(a == (short) b) {
        printf("a ist gleich b\n");
    } else {
        printf("a ist ungleich b\n");
    }
    return 1;      >gcc equals.c
}                  >./a.out
                  a ist gleich b
```

a = 0x002A
 b = 0xFFFF FFFF FFFF 002A

Implizites Typecasting

```
int main() {
    short a = 42;
    int b = -1;

    if(a == b) {
        printf("a ist gleich b\n");
    } else {
        printf("a ist ungleich b\n");
    }
    return 1;      >gcc compare.c
}                  >./a.out
                  b ist größer als a
```

a = 0x002A
 b = 0xFFFF

Vergleich beider Werte im
unsigned short Format

AUSLÖSUNG EINES BUFFER OVERFLOWS

- Weitere, Integer-bezogene Fehler:

- Kleine Offsets, die nicht erkannt werden (z.B. Off-by-one Fehler [SSH])
- Division durch 0 nicht abgefangen (z.B. Media Player DoS [MP])
- Inkonsistente Eingaben (z.B. 64 KB Nachricht mit 16 Byte Größe angegeben [HB])
- Erwartete Formate werden gebrochen (z.B. Nachricht ist immer 1000 Byte lang [BG])

- Weitere Programmierfehler:

- Fehler beim Umgang mit Strings und Character Arrays ([Java], [C])
- Fehler beim Umgang mit Expressions ([Java], [C])
- Fehler beim Umgang mit Datei Input / Output ([Java], [C])

ERKENNUNG UND VERHINDERUNG VON SOFTWARE EXPLOITS

- Software Schwachstellen \subseteq Software Bugs
 - Weniger Bugs → Weniger Schwachstellen
 - **Aber:** Alle Bugs in einem Programm zu beseitigen ist schwer
- Möglichkeiten zur Erkennung von Software Schwachstellen
 - Security Tests (z.B. Penetration Tests)
 - Code Reviews (4-Augen Prinzip)
 - **Statische-** und **dynamische Analyse**
- Möglichkeiten zur Vermeidung der Ausnutzung
 - Technische Mechanismen (z.B. **Stack Canaries** gegen Buffer Overflows)
 - Isolierung kritischer Softwarekomponenten (z.B. Sandboxing durch Virtualisierung)

PROGRAMM- UND CODEANALYSE

- **Grundidee:** Implementierungsfehler sind oft schwer manuell zu erkennen
- Automatisierter Ansatz, um Fehler zu highlighten oder auszulösen
 - **Statische Analyse:** Code wird analysiert aber nicht ausgeführt
 - **Dynamische Analyse:** Code wird mit Eingaben ausgeführt, die speziell gewählt wurden, um eine Fehlreaktion auszulösen (z.B. Grenzwerte, zu hohe Werte, NULL, ...)
- Statische- und dynamische Analyse haben ihre jeweiligen Vor- und Nachteile
 - **Statische Analyse:** Detaillierte Findings, von denen aber viele irrelevant sind
 - **Dynamische Analyse:** Konzentration auf relevante Findings, aber hoher manueller Analyseaufwand, um Ursache zu identifizieren

BEISPIEL STATISCHE ANALYSE

→ Statische Programmanalyse sucht nach Mustern im Code

- **Compilerwarnungen:** Häufige Fehlermuster (z.B. in gcc via „-Wall“ und „-Wextra“)
- **Spezielle Codeanalysetools:** Überprüfung der Regeln von Coding Standards (z.B. CERT-C [CERTC], MISRA-C [MISRA], CERT-Java [CERTJ])

```
int main(){
    unsigned int a = 42;
    Int b = -1;

    if(a < b){
        printf("a ist größer als b\n");
    } else{
        printf("b ist größer als a\n");
    }

    return 0;
}
```

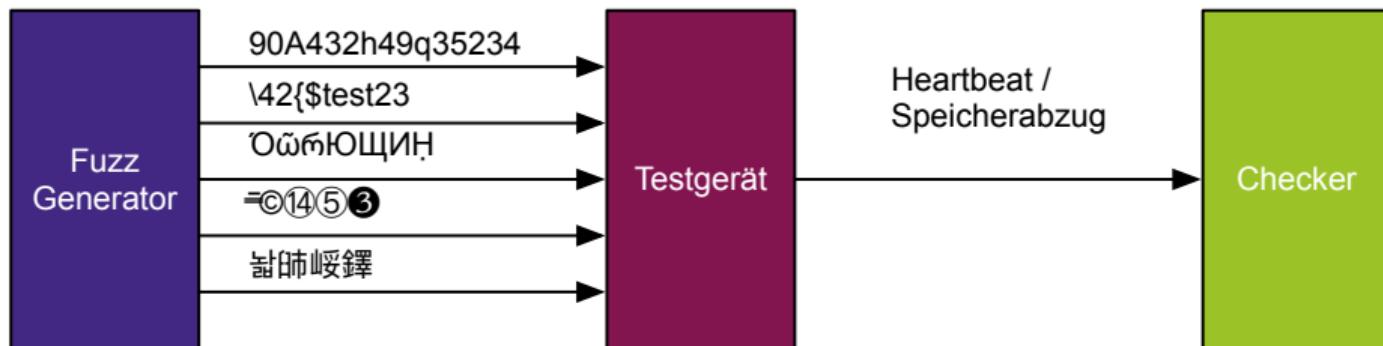
```
>gcc compare.c
>./a.out
b ist größer als a
```

```
>gcc compare.c -Wextra
compare.c: In function ‘main’:
compare.c:7:7: warning: comparison of integer expressions
  of different signedness: ‘unsigned int’ and ‘int’ [-Wsig
n-compare]
      7 |   if(a > b) {
      |   ^
```

BEISPIEL DYNAMISCHE ANALYSE

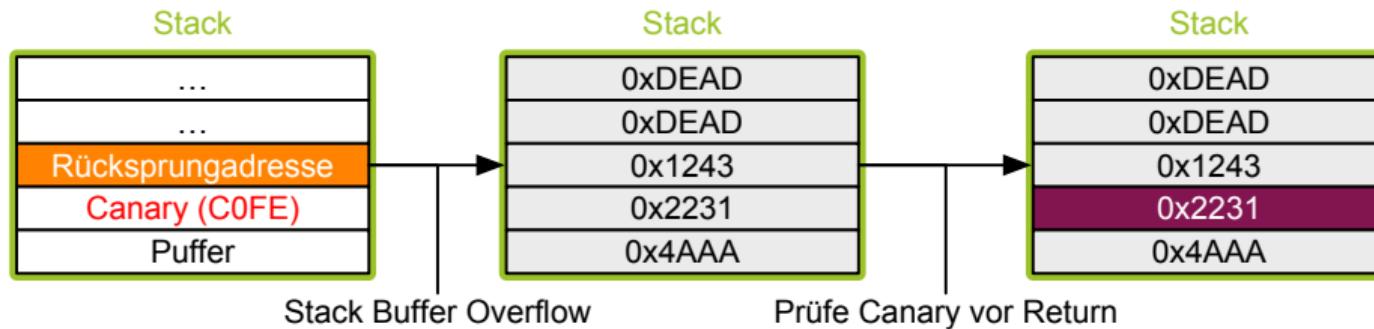
- Dynamische Analyse unterstützt die Fehlersuche durch Ausführung des Codes
 - **Fuzzing**: Automatisiertes Testen mit zufällig und bewusst non-konform gewählten Werten mit dem Ziel Fehlverhalten auszulösen
 - **Taint Analyse**: Analyse des Einflusses durch externe Eingaben
 - **Symbolische Ausführung**: Testabdeckung aller Pfade in einem Programm durch Analyse und Auswahl geeigneter Eingaben

Beispielaufbau- und Ablauf von Fuzzing



TECHNISCHE GEGENMASSNAHMEN BEISPIEL STACK CANARIES

- Stack Canaries sind zufällige Werte, die vor die Rücksprungadresse gelegt werden, um Stack Buffer Overflows vor dem Rücksprung festzustellen
- Die Sicherheit von Stack Canaries hängt davon ab, dass [BKK+18]:
 - Der Canary Wert nicht geraten werden kann
 - Der Referenzwert und Vergleich „sicher“ gespeichert und implementiert sind



ERKENNUNG UND VERMEIDUNG VON SOFTWARE SCHWACHSTELLEN

- Beseitigung aller Software Schwachstellen im Programm ist unmöglich
 - Moderne Software umfasst oftmals hunderttausende bis Millionen Zeilen Code
 - Statische und dynamische Analysen erkennen nur bekannte Muster
 - Zeit- und Kostendruck verhindern häufig weitere Beseitigung
- Selbst wenn der eigene Code frei von Schwachstellen ist
 - Compiler verändern Code und führen Schwachstellen ein [MSC]
 - Inkludierte Bibliotheken können Fehler beinhalten (z.B. log4j)
 - Angreifer können gezielt korrompierte Bibliotheken einfügen (z.B. Supply-Chain Angriffe)
 - Ausführende Hardware kann Fehler beinhalten (z.B. Meltdown/Spectre [MELT])
- Priorisierung der Mechanismen und der untersuchten Softwaremethoden sowie ein gutes Schwachstellenmanagement sind essentiell für sichere Software

ZUSAMMENFASSUNG

- Hintergrund von CVSS und kennen von fünf relevanten Kriterien zur Berechnung
- Die fünf häufigsten auftretenden Software Schwachstellen
- Definition und technischer Ablauf eines Buffer-Overflows in C
- Schwachstellen basierend auf fehlerhafter Integer Behandlung
- Die Effekte von Schwachstellen
- Methoden, um Softwareschwachstellen zu erkennen bzw. zu vermeiden



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Plattformintegriät

June 13, 2023

Marc Stöttinger



Crypto will not be broken, it will be bypassed.

Adi Shamir

MOTIVATION

- Alle Sicherheitskonzepte und -technologien werden auf **einer Plattform** (Software und Hardware) implementiert und betrieben.
- Die Integrität der Plattform ist damit die **Grundvoraussetzung** zur korrekten Funktionsweise der Sicherheitskonzepte im Programm.
- Eine Applikation oder ein Programm kann in der Regel nicht prüfen, ob Sie in einer **integren Umgebung** ausgeführt wird oder kommuniziert.

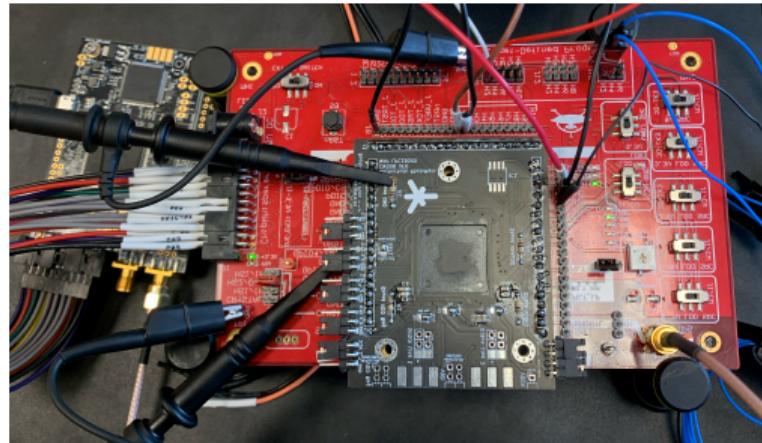


Source:

<https://www.pinterest.de/pin/452682200017263048/>

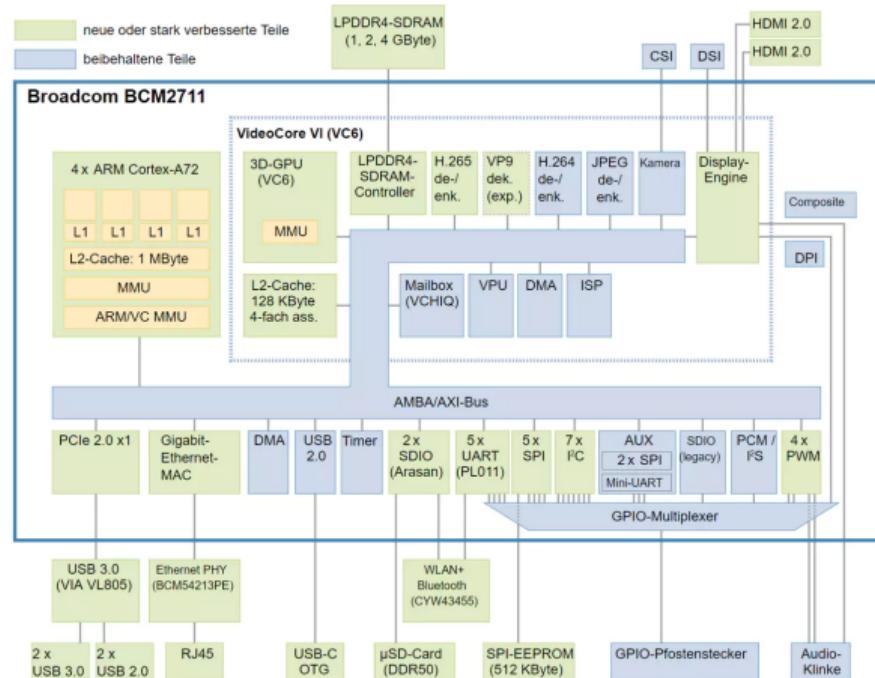
EXEMPLARISCHE ANGRIFFSVEKTOREN

- Reverse Engineering oder Manipulation der Hardware - Hardware Trojaner
- Betrieb ausserhalb der Spezifikation - Fehlerangriffe
- Wartungsschnittstellen ausnutzen - Fuzzing des JTAG interfaces
- Firmware/Betriebssystem manipulieren - RootKit
- Software/Applikationen manipulieren - Malware



SICHERHEITSARCHITEKTUR - BESPIELE

- Moderne SoCs haben externe Speicher
- Einen Vielzahl von Prozessoren
- Viele Konfigurationsmöglichkeiten



Quelle: <https://www.heise.de/hintergrund/Raspberry-Pi-4-Model-B-Blockschaltbild-des-Broadcom-BCM2711-4514399.html>

SICHERHEITSARCHITEKTUR - HARDWARE

1. Speicher - Schlüssel und Credentialsverwaltung

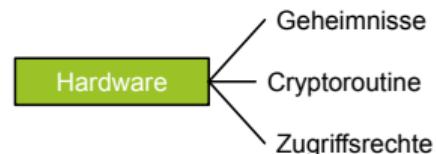
- Schlüssel und Zertifikaten sind (**Root of Trust**)
- Schlüssel nur benutzbar aber nicht lesbar

2. Cryptoprozessoren - Kryptographische Algorithmen

- Bereitstellen von **nicht veränderbaren** Routinen
- Tradeoff zwischen Performance und Ressourcen

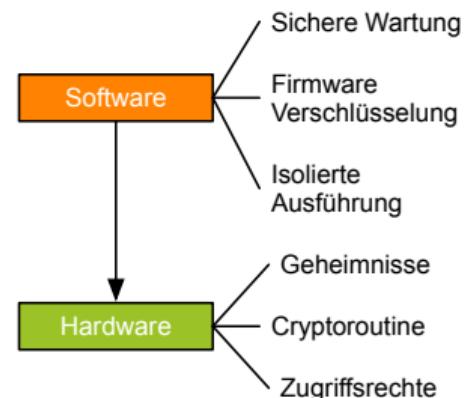
3. Management - Speicher und Resourcenzugriff

- Freigabe von Ressourcen **auf unterster Ebene**
- Verhindern von nicht autorisiertem Zugriff



SICHERHEITSARCHITEKTUR - SOFTWARE

1. Sicheres Instandhaltung und Booten von Software
 - System startet nur mit **authentischer Software**
 - Nur autorisierten Instanzen wird Wartung gestattet
2. Verschlüsselte Firmware oder Filesystem
 - **Erschweren** von Reverse Engineering
 - Zugriffsrechte auf Daten streng geteilt
3. Isolierte Ausführung - Sandboxing
 - Ausführen in isolierter Umgebung
 - **Trennung** von Prozessausführung und Dateizugriff



SICHERHEITSARCHITEKTUR - SICHERHEITSSERVICES

1. Verschlüsselungs- und Authorisierungsservices

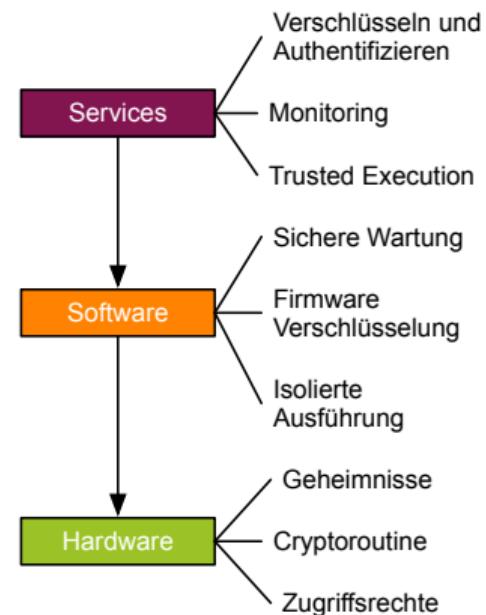
- **Sicheres Speichern** von sensiblen Daten
- Authorisieren von Kommunikation

2. Monitoring und Prüfen des Systems

- **Integritätscheck** zur Laufzeit
- Protokollieren von Zugriffen und Prozessen

3. Etablierung einer vertrauenswürdigen Ausführungsumgebung

- Ausführen von Prozessen in gesicherter Umgebung
- **Verarbeiten** sensibler Daten



ALLGEMEINE SICHERHEITSMASSNAHMEN FÜR SOFTWARE INTEGRITÄT

1. Sicherer Wartungszugang

- Zugriffskontrolle, nur **authentische Entitäten** haben Wartungs- und Analysezugriff auf die Software

2. Sicheres aktualisieren der Software

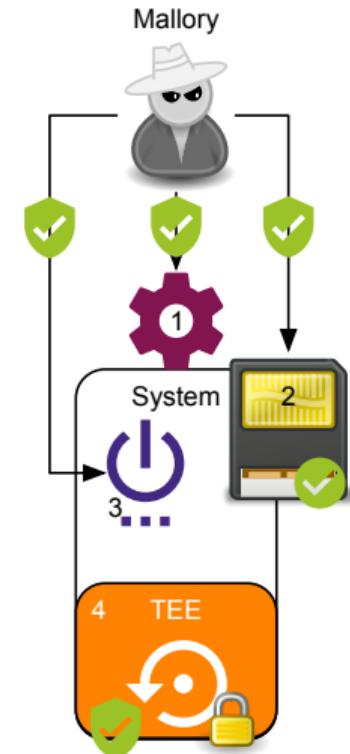
- **Schutz vor Manipulation** der Software, nur vertrauenswürdige Updates werden durchgeführt

3. Sicheres Booten der Software

- Nur **authentische Software** soll ausgeführt werden

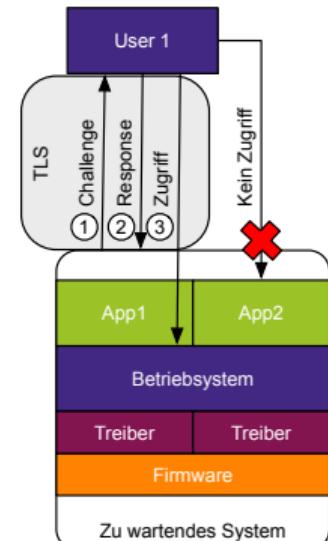
4. Vertrauenswürdige Ausführungsumgebung

- Software kann **isoliert und authentisch** ausgeführt werden



SICHERER WARTUNGSZUGANG - AUTHENTIFIZIERUNG

- Zugriff nur nach Authenzitätprüfung
 - Oft über ein **Challenge-Response-Protokoll** realisiert
 - Wartung unterschiedlicher Software Partitionen
 - **Unterschiedliche** Zugriffsrechte
- Eventuell Wartung nur in Wartungszustand des Geräts möglich
 - Remote-Wartung meist zusätzlich mit **abgesichertem Kommunikationkanal**
 - Mehrere Authentifizierungen je nach Aktion notwendig
 - Zugriff kann über **abgesicherte Logs** dokumentiert werden
- Typischerweise werden **Debug-Ports und -Schnittstellen**, wie JTAG SWD aber auch UART damit abgesichert.



SICHERER WARTUNGSZUGANG - IMPLEMENTIERUNGASPEKTE

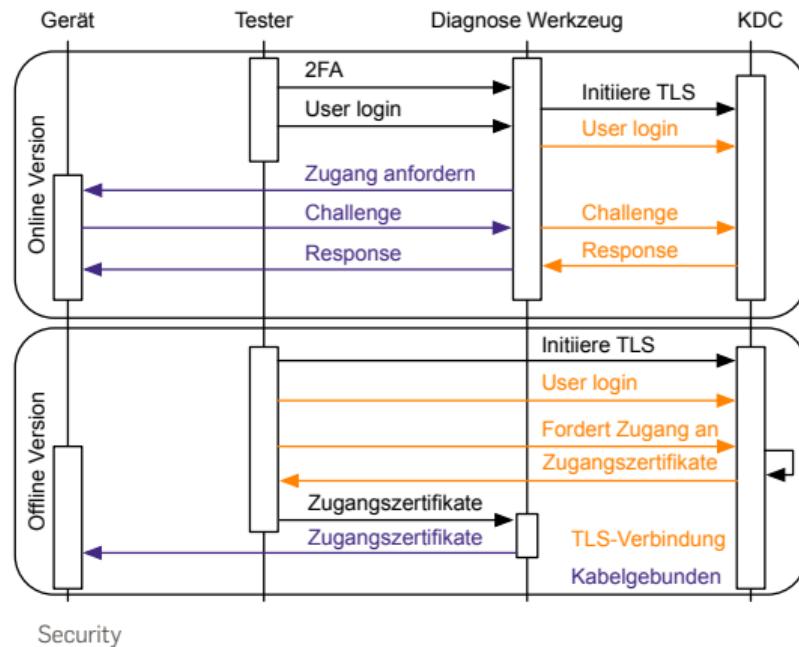
Der Wartungszugang hängt meist vom Szenario ab und hat damit verschiedene Sicherheitsaspekte

→ **Online Variante**

- Benutzer muss sich beim KDC/PKI anmelden
- CR-Protokoll läuft über KDC
- Permanente Online-Verbindung notwendig

→ **Offline Variante**

- Mit PKI - Zertifikat mit kurzer Laufzeit wird über CA/KDC beantragt
- Mit KDC - Session Schlüssel muss beim KDC beantragt werden



SICHERES SOFTWARE UPDATE - PRÜFUNG DER SOFTWARE

Software Updates sind eine geeignete Maßnahme, um neue Schwachstellen und potentielle Exploits zu beheben

- Rechtzeitiges Durchführen von Updates ist wichtig
 - Durch **Vulnerability Management** können frühzeitig neue Schwachstellen und potentielle Exploits identifiziert werden
- Bibliotheken oder verwendete Software-Komponenten müssen ebenfalls aktuell gehalten werden
 - Eine **Software Bill of Material (SBOM)** ist daher notwendig, um betroffene Software zu identifizieren.
 - Aktualisierungsstrategie eventuell notwendig wegen Abhängigkeiten

SICHERES SOFTWARE UPDATE - IMPLEMENTIERUNGSASPEKTE

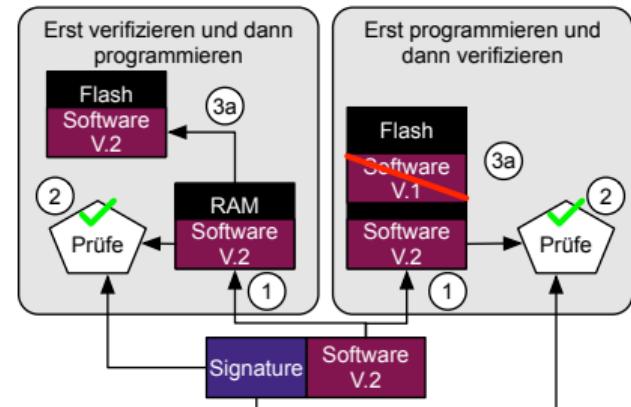
Der Zeitpunkt der Prüfung der Authentizitätsprüfung beim Software Update ist kritisch

→ **Zuerst prüfen** und dann programmieren

- 1 Laden der neuen Software in den RAM
- 2 Verifizieren, ob die Software valide ist
- 3a Software valide: Alte Software im FLASH überschreiben
- 3b Software nicht valide: Neue Software aus RAM löschen

→ **Zuerst programmieren** und dann prüfen

- 1 Laden der neuen Software in den FLASH
- 2 Verifizieren, ob Software valide ist
- 3a Software valide: Alte Software aus dem FLASH löschen
- 3b Software nicht valide: Neue Software dem FLASH löschen

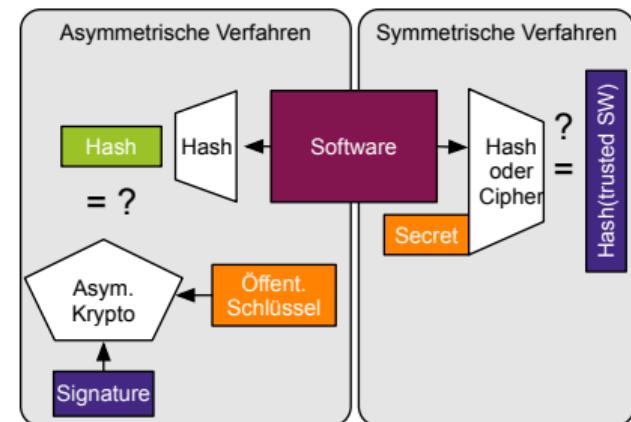


Time-of-Check, Time-of-Use bei der Prüfung beachten!
[TOCTOU] [CWE-367]

SICHERES BOOTEN DER SOFTWARE - PRÜFVERFAHREN

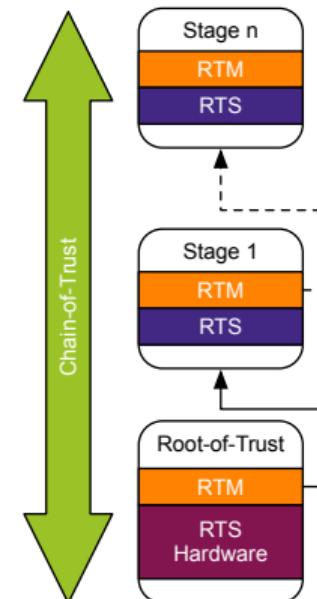
Prüfung der installierten Software auf Authenzität vor dem Ausführen

- Prüfung (einmalig) findet **vor dem Ausführen** statt, nicht zur Laufzeit
- Die Prüfung auf Authentizität kann durch sym. und asym. kryptographische Algorithmen umgesetzt werden.
- Software kann auch in kleineren Partitionen geprüft werden
- Schlüsselmaterial muss **manipulationssicher** auf der Plattform gespeichert sein



SICHERES BOOTEN DER SOFTWARE - ROOT OF TRUST

- **Root of Trust (RoT)** ist eine Komponente, die vertrauenswürdig sein muss und deren Manipulation während des Bootvorgangs nicht erkannt werden kann
- RoT erfordert Hardware-Unterstützung, zum **Schutz vor Manipulation**
- **RoT for storage (RTS)** bietet ein sicheres Speichermedium, das die Integrität des gespeicherten Inhalts schützt und sichert
- Basieren auf der Rot wird einen Vertrauenskette aufgebaut (**Chain-of-Trust**)
- **RoT for measurement (RTM)** misst die Integrität von jedem Element in der Kette (Softwarekomponenten) und überprüft ihre Authentizität.



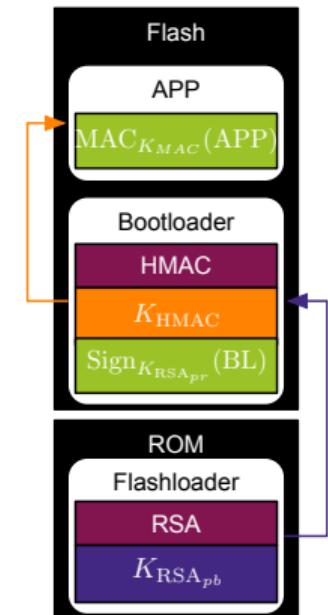
SICHERES BOOTEN DER SOFTWARE - MASSNAHMEN

Unterschiedliche Strategien bei negativer Validierung der Software:

- Software wird auf der Plattform nicht ausgeführt:
 - Schutz vor manipulierter Software oder Malware
 - Prinzip von **Secure Boot**
- Software wird auf der Platform trotzdem gestartet und andere Gegenmaßnahmen eingeleitet:
 - Sicheres Wegschreiben der Manipulation (Secure Logging)
 - Nicht Freigeben von Schlüsselmaterial
 - Eingeschränkter Zugriff auf Ressourcen, z.B. Speicherbereiche
 - Vorgehensweise wird als **Trusted**, **Measured** oder **Authentic Boot** bezeichnet

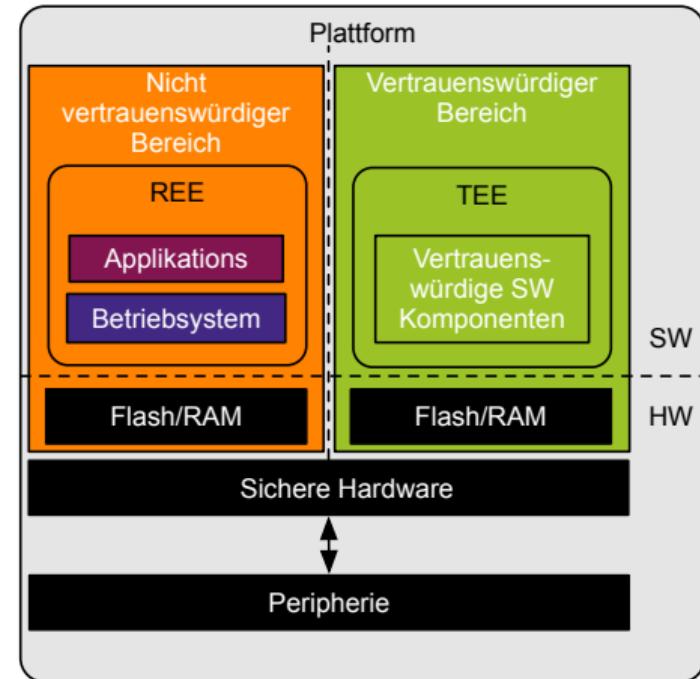
SICHERES BOOTEN DER SOFTWARE - IMPLEMENTIERUNGSASPEKTE

- Beachtung von Race Conditions (TOCTOU) bei der Implementierung von sicherem Booten
 - Die Reihenfolge der Vertrauenskette darf nicht manipulierbar sein
 - Wann und was wird geprüft - Prüfung was vom Flash geladen wird oder was in den RAM geladen wurde
- Wird die Authentizitätsprüfung von jeder Stufe individuell oder zentral durchgeführt?
- Wie stark ist die Root of Trust gegen Manipulation abgesichert?



VERTRAUENSWÜRDIGE AUSFÜHRUNGSUMGEBUNG

- Mit einer vertrauenswürdigen Ausführungs-umgebung (TEE) werden vertrauenswürdige SW-Komponenten in einer gesicherten Umgebung ausgeführt
 - Hardwareseitige Trennung der Komponenten
 - Eigener Speicherbereich
 - Separate Prozessausführung auf dem Prozessor
- Trustzone und SXG sind aktuelle Beispiel für TEEs
- Einige Secure Enclaves erweitern das TEE-Konzept um einen separaten Prozessor exklusiv für vertrauenswürdige Ausführungen



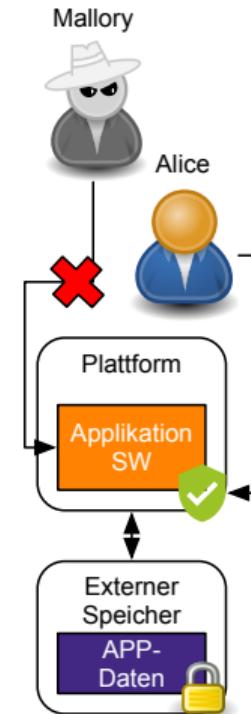
ANDERE SCHUTZMASSNAHMEN ZUR LAUFZEIT

- Security Manifest
 - Prüft welche Applikation mit welcher Applikation kommunizieren darf
 - Welche Systemservices von den Applikationen genutzt werden dürfen
- Stack Canaries und Address Space Randomisation ASLR
 - Verhindert das Ausnutzen von Buffer Overflow Attacks
 - Randomisiert die Segmentadressen und beugt damit dem Ausnutzen von festen Sprungadressen vor
- Integritätscheck zur Laufzeit
 - Attestierung von Applikationen
 - Speicherseiten Hashen und Vergleichen
- Sandboxing
 - Applikationen haben nur im Nicht-Super-User-mode Zugriff auf eigene Daten
 - Kein Zugriff auf Systemdaten

ALLGEMEINE SICHERHEITSSERVICES AN DIE SOFTWARE

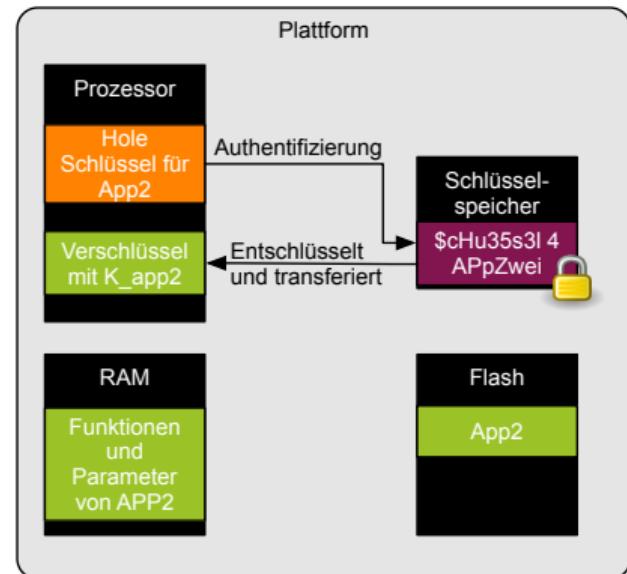
Zum Schutz der Software-Applikationen kann mit Hilfe der Plattform-Integrität auch die Integrität/Authenzität der Anwendungen gewährleistet werden:

- Sichereres Aufbewahren von Schlüsseln → Geheimnisse müssen nicht im Code gespeichert werden
- Verschlüsseln der Firmware → Schutz der Daten in ungeschützten Speichern
- Attestierung der Software → Authenzitätsnachweis gegenüber dritten Instanzen



SCHLÜSSELMANAGEMENT

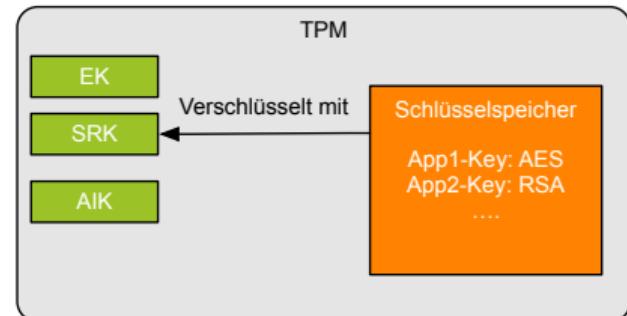
- Sicherheit von Verschlüsselungsalgorithmen und Authentisierungsprotokollen basiert auf der Geheimhaltung des Schlüssels
- Der Schlüsselspeicher verwaltet sicher die Schlüssel von Anwendungen und Services
- Die Schlüssel sind verschlüsselt gespeichert und werden nur bei Bedarf entschlüsselt zur direkten Verwendung an den Prozessor übergeben
- Erst nach einer Authentifizierung werden die Schlüssel zur Bearbeitung freigegeben



SCHLÜSSELMANAGEMENT MIT TPM

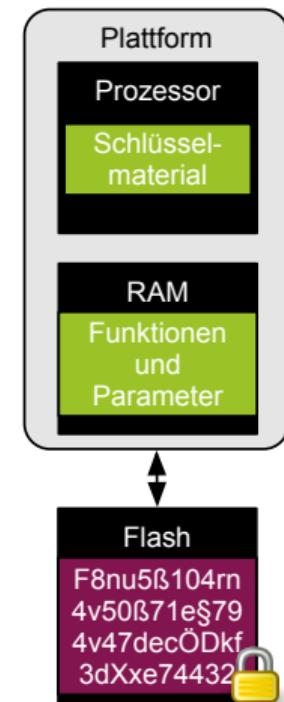
Ein TPM kann Schüssel für Verschlüsselung und Authentifizierung eigenständig erzeugen und im eigenen Schlüsselspeicher verwalten

- Endorsement Key (EK_{pr} , EK_{pb})
 - Identifikationsschlüssel des TPMs zur eindeutigen Identifikation des Moduls über EK_{pb}
 - Der Schlüssel wird vom Hersteller signiert
 - EK_{pr} verlässt nie das Modul
- Storage Root Key (SRK_{pr} , SRK_{pb})
 - Hauptschlüssel des Schlüsselspeichers im TPM
 - Wurzel des Schlüsselbaum im TPM
 - Mit Besitzerwechsel wird ein neuer SRK erzeugt
- Attestation identity Key (AIK_{pr} , AIK_{pb})
 - Schlüssel für Remote Attestierung
 - Wird pro Benutzer aus dem EK erzeugt



FIRMWARE-VERSCHLÜSSELUNG

- Verschlüsselung der Firmware, um Reverse Engineering zu erschweren
 - Verhindert das Identifizieren von Schwachstellen im Code
 - Schützt für die Ausführung sensible Daten, z.B. IP-Adresse vom Server
- Schutz der Firmware, wenn der Programm- oder Datenspeicher leichter zugänglich ist
 - Flash ist extern, z.B. SoC
 - Speicher ist portable und austauschbar, z.B. SD-Card
- Zusätzlich muss die Authenzität der verschlüsselten Firmware noch sichergestellt werden

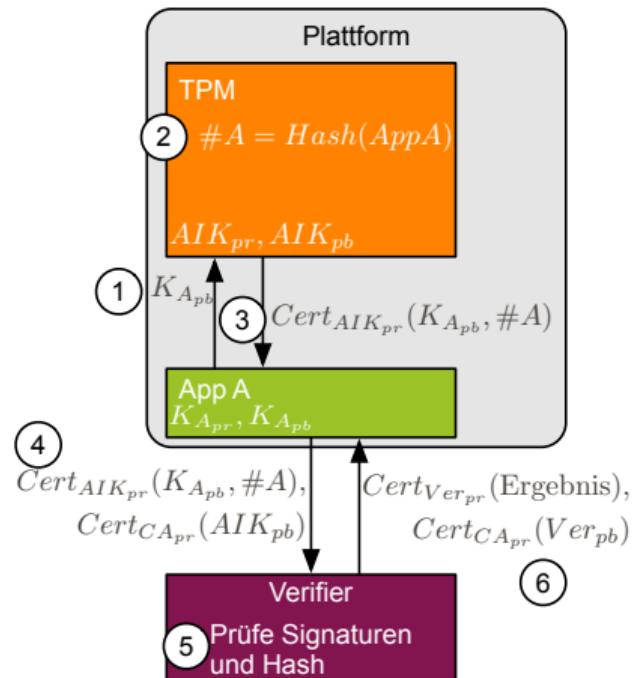


ATTESTATION

- Das sichere Booten kann eine Änderung der Software nach dem Abschluss des Bootprozesses nicht feststellen:
 - **Kein Schutz** gegen Software Manipulation **zur Laufzeit**
- Referenzwert für die Authentizität (z.B. Hash, öffentlicher oder geheimer Schlüssel) ist auch auf der Platform gespeichert
 - Durch die **Manipulation des Referenzwerts** kann die Prüfung manipuliert werden
- Mit einer **Attestierung** kann der aktuelle Zustand der Software und die Authentizität zur Laufzeit gemessen werden
- Ein TPM oder eine Secure Enclave kann genutzt werden, um den **aktuellen Zustand der Software** zur Laufzeit zu **messen**

ATTESTATION MIT TPM

1. Applikation A generiert eigene Schlüssel
 $K_{A_{pr}}, K_{A_{pb}}$
2. TPM berechnet den Hash von Applikation A
3. TPM generiert ein Zertifikat mit dem Hash von A und $K_{A_{pb}}$ und übergibt es an A
4. Applikation A sendet das vom TPM erzeugte Zertifikat zusammen mit einem Zertifikat über das TPM an den Verifier
5. der Verifier prüft die Zertifikate, um die Authentizität der Daten zu prüfen und vergleicht den Hash von A
6. Das Ergebnis wird verifiziert zurückgeschickt



HARDWAREUNTERSTÜZUNG FÜR SICHERHEITSMECHANISMEN

Alle bisher vorgestellten **Konzepte** für Plattform-Integrität und Services benötigen

- Kryptographische Algorithmen
- Protokolle
- Geschützte Komponenten, z.B. Speicher

Umsetzung der Services wirkt sich auf die **Architektur und Komponenten** der Plattform aus:

- Externe oder interne Beschleuniger für kryptographische Algorithmen
- Eigene unabhängige Instanz zum Umsetzen von Protokollen und Services
- Erweiterung des Prozessors um einen vertrauenswürdigen Bereich

ÜBERSICHT DER SICHERHEITSMODULE

→ **Crypto-Beschleuniger**

- Stellt kryptographische Algorithmen zur Verfügung
- Nicht programmierbar

→ **Trusted Platform Modul - TPM**

- Stellt kryptographische Algorithmen und Funktionalitäten zur Verfügung
- Nicht programmierbar

→ **Hardware Security Modul - HSM**

- Stellt systemspezifische Sicherheitsfunktionen zur Verfügung
- Programmierbar in einer gesicherten Entwicklungsumgebung

→ **Security Enclave - SE**

- Ermöglicht gesicherte Ausführung von Operation auf dem Applikationsrechner
- Funktionalitäten sind nicht programmierbar - jedoch für eigene Programme nutzbar

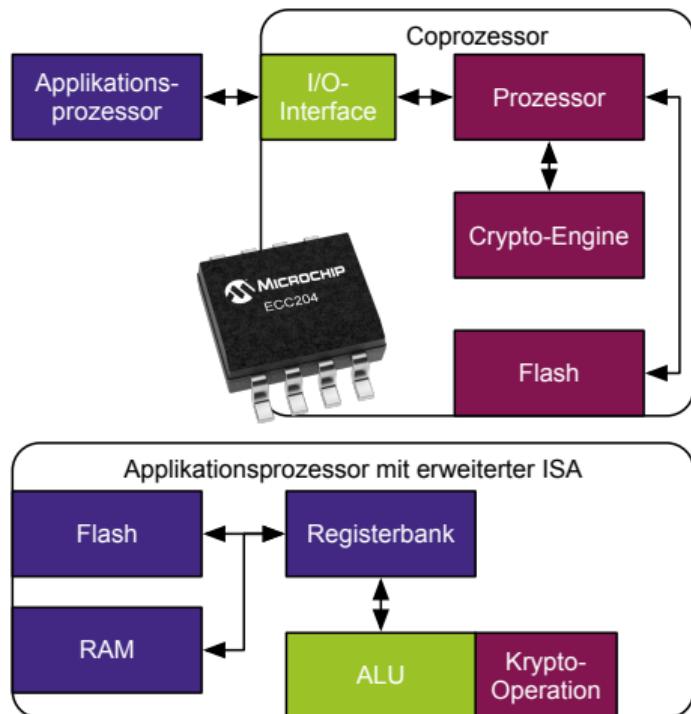
SICHERHEITSMODULEN UND ALLGEMEINE SICHERHEITSMASSNAHMEN

- Nicht jedes (Hardware)Sicherheitsmodul kann für alle Sicherheitsmaßnahmen genutzt werden
- Sobald die Sicherheitsmodule außerhalb des IC-Gehäuses sind, wirkt das Ziel nur auf Komponentenebene!

Gegenmaßnahmen	Crypto-Beschleuniger	HSM	TPM	SE
Sicherer Wartungszugang		x	x	
Sicheres Software Update		x	x	
Sicheres Booten		secure boot	measured boot	
Vertrauenswürdige Ausführungsumgebung		x		x
Remote Attestation			x	x

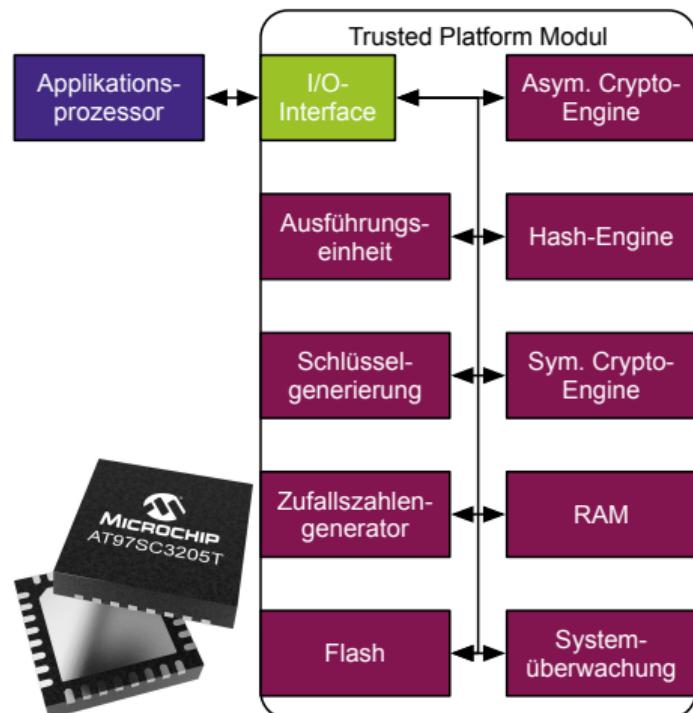
CRYPTO-BESCHLEUNIGER

- Herstellerspezifische Eigenschaften
- Coprozessor
 - Nur kryptographische Algorithmen, keine Protokolle
 - Kombination aus Coprozessor mit Hardwarebeschleuniger
 - eigener Schlüsselspeicher
- Kryptographische Funktionen als Erweiterung der Instruction Set Architektur
 - Basisoperationen kryptographische Algorithmen
 - Teil des Applikationsprozessors
 - Kein eigener Schlüsselspeicher



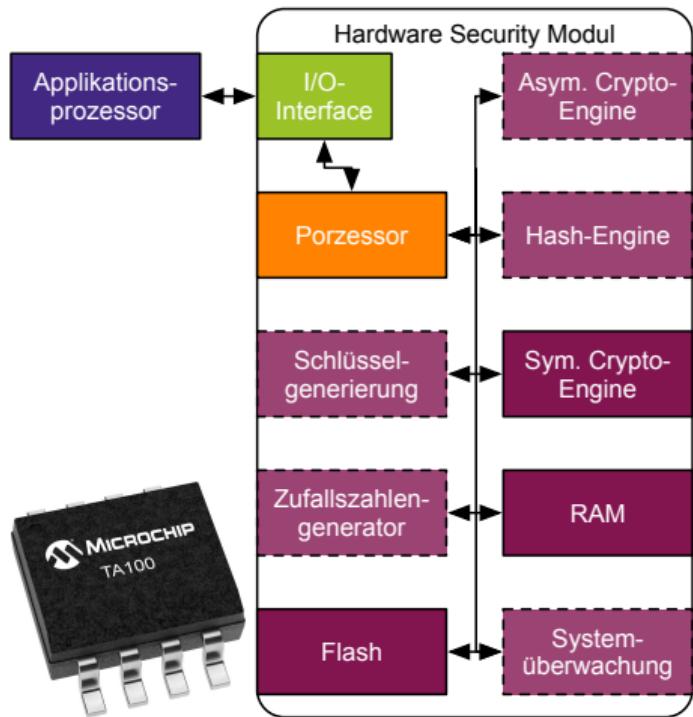
TRUSTED PLATFORMMODUL - TPM

- Externe Komponente (außerhalb des Chips)
- Bietet Sicherheitsservices
 - Attestierung von Komponenten
 - Schlüsselgenerierung und -verwahrung
 - Signaturgenerierung und -verifikation
 - Ver- und Entschlüsselungsservice
- Plattform ist standardisiert bei der Trusted Computer Group Organisation
- Aktuelle Version ist [TPM 2.0], kann zertifiziert werden nach CC



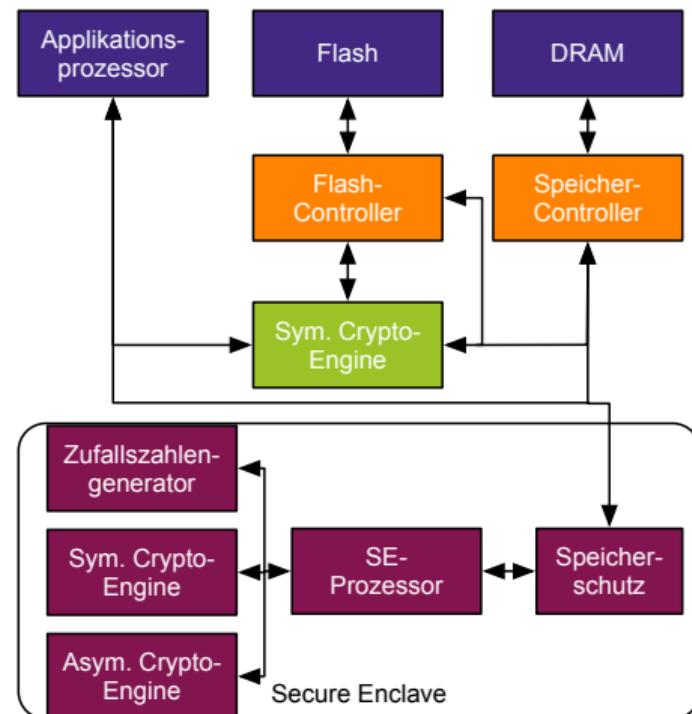
HSM

- Herstellerspezifische Eigenschaften
- Externe Komponente oder intern als Coprozessor im Applikationsprozessor
- Gibt es auch pro Hersteller in verschiedenen Varianten
 - Unterstützte Algorithmen variieren je Baustein
 - Interner Prozessor steuert die Sicherheitsfunktionen
 - Eigene Software kann eingebracht werden
 - Spezielle Entwicklungsumgebung wird benötigt
- Einsatz sowohl in eingebetteten System als auch in IT-Infrastrukturen (dort ist es eine eigenes Gerät)



SECURITY ENCLAVE - SE

- Internes Subsystem zur vertraulichen Behandlung von sensiblen Daten
- Die SE nutzt die Infrastruktur des SoCs oder Prozessors, um virtuell getrennte Maschinen zu erstellen
 - Pro Instanz individuell verschlüsselter und authentischer Speicher
 - Schlüsselspeicher
 - Attestierung von Software
 - Etabliert eine vertrauenswürdige Ausführungsumgebung (TEE)
- Wird standardmäßig in aktuellen Prozessoren von ARM, Apple, AMD und Intel eingesetzt



ZUSAMMENFASSUNG

- Verschiede Angriffsverktoren auf eine Plattform
- Abhängigkeiten zwischen Hardware und Software in einer Sicherheitsarchitektur
- Diskussion von vier Basismaßnahmen und zur Etablierung von Plattformintegrität
- Implementierungsaspekte der Basismaßnahmen
- Sicherheitsservices welche die Sicherheit von Software erhöhen können
- Aufbau und Sicherheitsfunktionen von Hardwaresicherheitsmodulen