IT Security

Jil Zerndt
FS 2025

01 - Intro

Business IT-Risks

- Ransom Demand
- Fraud
- Espionage
- Violation of Regulations
- Misuse of Computing Resources
- Reputation Loss
- System Outage
- Sabotage
- Data Loss
- Brand Misuse

Problems - Overview and their impact on data availability

Physisch

unabsichtlich

- Naturkatastrophen
- Feuer
- Ausfall
- Kaffee auf Server

absichtlich/bösartig

- Feuer
- Vandalismus
- Garantie läuft aus -> absichtlich langsamer
- Social Engineering

Virtuell

unabsichtlich

- Bitflip
- Config Fehler
- Bugs im SW
- Phishing klicken

absichtlich/bösartig

- DDoS
- Malware
- Ransomware
- Phishing senden
- Trojaner

Countermeasures - Overview

Disaster Recovery

- Offline backup solutions
- Restoring from images

Access Control

- Restricted Access Rights
- Multi-Factor Authentication
- Firewalls
- Traffic Management Solutions

Physical Protection

- Physical Access Control (locks, fences, etc.)
- Fire Protection (extinguishers, alarms, etc.)
- Monitoring (CCTV, Guards etc.)

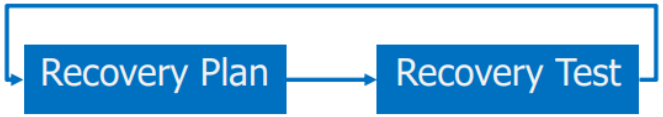
Training Processes

- Employee Training
- Four eyes principle
- Automation of routine processes
- Monitoring
- Preventive maintenance

Redundancy

- Uninterruptable Power Supplies
- High Availability setups
- Load Balancing
- Redundant data center
- Redundant network connections

Recovery Plan and Test



Recovery Plan - description of what to do if something goes wrong

- Roles and responsibilities
- Processes
- Contact details
- Technical instructions

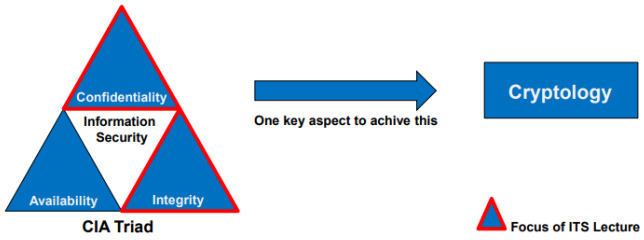
Recovery Test - testing the recovery plan

- Theoretical dry run
- Practical tests
  - turn off a server or DC
  - restore data from backup

Goals of IT Security

Most measures in Information Security have one of the three following high-level goals:

- Ensure data is confidential
- Ensure data is not corrupted
- Ensure data and systems are available



Kryptographie

Einführung in die Kryptographie

In diesem Kapitel schauen wir uns die Ziele, Grundbegriffe und Modelle der Kryptologie an.

Zweige der Kryptologie

Die **Kryptologie** ist ein Teil der Mathematik, welcher sich mit dem sicheren Übertragen und Speichern von Nachrichten beschäftigt.

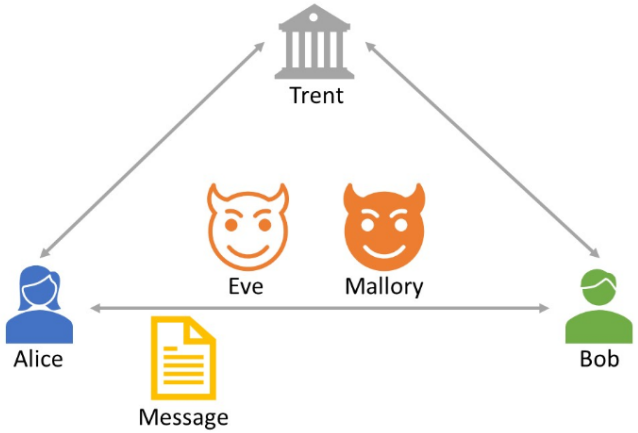
- **Kryptographie**: Die Wissenschaft der Verschlüsselung von Nachrichten (sicheres Speichern und Übertragen) → sichere Protokolle und deren Aufbau
- **Kryptoanalyse**: Die Wissenschaft des Entschlüsselns von Nachrichten (wie können Mechanismen der Kryptographie gebrochen werden?) → alte, unsichere Protokolle und deren Schwachstellen

Ziele der Kryptographie Nachrichten sicher übertragen und speichern. *Sicherheit* hat dabei verschiedene Aspekte:

- **Confidentiality:** Nur berechtigte Personen können eine Nachricht lesen. Unberechtigte Personen können die Nachricht zwar sehen, sie aber nicht entziffern, da die Nachricht für sie nur aus einer zufälligen Zeichenfolge zu bestehen scheint.
- **Integrity:** Eine Nachricht wird vom Empfänger so empfangen, wie sie vom Sender geschickt wurde. Das heisst, eine unberechtigte Person könnte zwar eine Nachricht abfangen, verändern, und dem Empfänger zustellen. Dieser würde dann aber merken, dass die Nachricht nicht der ursprünglichen Nachricht entspricht und sie daher verwerfen.
  → Der Empfänger kann sicher sein, dass die Nachricht nicht verändert wurde.
- **Authenticity:** Der Empfänger kann sicher sein, dass eine Nachricht auch wirklich von der Person stammt, von welcher die Nachricht zu kommen scheint. Das heisst, er kann überprüfen, ob der angegebene Absender auch dem tatsächlichen Absender entspricht.
- **Non-repudiation:** Wenn eine Person eine Nachricht bekommen hat, kann diese Person nicht abstreiten, dass sie die Nachricht erhalten hat. Das heisst, es kann bewiesen werden, dass sie die Nachricht erhalten hat.
- **Freshness:** Alle erhaltenen Nachrichten sind aktuell. Das heisst, ein Attacker könnte zwar eine Nachricht zurückbehalten und später senden, oder eine abgefangene Nachricht duplizieren und zu einem späteren Zeitpunkt noch einmal senden, aber dies würde vom Empfänger bemerkt.
- **Anonymity:** Der Absender und/oder Empfänger einer Nachricht bleiben unbekannt.

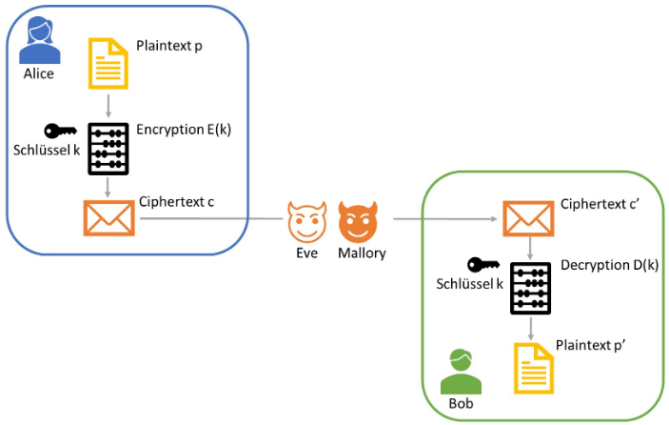
Model der Kommunikation und Verschlüsselung

Kommunikationsmodel



- **Alice** sendet eine Nachricht an Bob. Dabei sollen die oben genannten Ziele, confidentiality, integrity, authenticity, non-repudiation und freshness erfüllt werden.
- **Bob** empfängt die Nachrichten von Allice. Er will überprüfen können, dass die Ziele eingehalten wurden.
- **Eve** ist ein Attacker. Sie kann Nachrichten mitlesen, aber sie nicht verändern.
- **Mallory** ist ein anderer Attacker. Er kann Daten sowohl mitlesen als auch verändern. Er kann auch Nachrichten abfangen und später weitersenden, oder ganz verwerfen. Er kann auch neue Nachrichten generieren.
- **Trent** ist eine Drittperson/Instanz, welcher sowohl Alice und Bob vertrauen. Trent unterstützt Alice und Bob bei der sicheren Kommunikation.

Model der Verschlüsselung Um eine vertrauliche Kommunikation zu erreichen, werden Nachrichten vor dem Senden verschlüsselt und nach dem Empfangen wieder entschlüsselt.



- **Plaintext (Klartext):** Der Plaintext ist der Text, so wie er geschrieben, respektive gelesen werden kann. Er wird mit dem Buchstaben "päbgekürzt.
- **Ciphertext (Verschlüsselter Text):** Der Ciphertext ist der Text, welcher durch die Verschlüsselung entsteht. Er wird mit dem Buchstaben "cäbgekürzt.
- **Encryption (Verschlüsselung):** Die Verschlüsselung macht aus dem Plaintext den dazugehörenden Ciphertext. Dazu wird ein Schlüssel verwendet. Die Verschlüsselung kann wie folgt angegeben werden: $c = E[k](p)$. Der Verschlüsselungsalgorithmus selbst ist öffentlich bekannt und kann von allen analysiert werden, um mögliche Schwachstellen zu finden.
- **Decryption (Entschlüsselung):** Die Entschlüsselung macht aus einem Ciphertext den dazugehörenden Plaintext. Dazu wird ein Schlüssel verwendet. Die Entschlüsselung kann wie folgt angegeben werden p' = D[k](c'). Der Entschlüsselungsmechanismus ist öffentlich bekannt und kann von allen analysiert werden, um mögliche Schwachstellen zu finden.
- **Key (Schlüssel):** Nur mit dem richtigen Schlüssel, kann eine Nachricht richtig entschlüsselt werden. Je nach Art der Verschlüsselung wird derselbe Key für die Verschlüsselung und die Entschlüsselung verwendet (Secret Key Kryptographie) oder es werden unterschiedliche Schlüssel verwendet (Public Key Kryptographie). Damit die Verschlüsselung sicher ist, muss der Schlüssel, welcher für die Entschlüsselung gebraucht wird, geheim bleiben.

Confidentiality ist erreicht, wenn Eve (und Mallory) den Ciphertext c nicht lesen können. **Integrity** ist erreicht wenn der von Bob empfangene Plaintext p' dem von Alice gesendeten Plaintext p entspricht, also p' = p ist. Authenticity ist erreicht, wenn Bob sicher sein kann, dass die Nachricht von Alice stammt. Non-repudiation ist erreicht, wenn Alice nicht abstreiten kann, dass sie die Nachricht gesendet hat. Freshness ist erreicht, wenn Bob sicher sein kann, dass die Nachricht aktuell ist.

Attacktypen auf Kryptosysteme

Die Unterschiede zwischen den Attacken bestehen daraus, auf was der Attacker Zugriff hat.

Ciphertext-only attack Der Attacker kann vom Ciphertext alleine Rückschlüsse auf den Plaintext oder den verwendeten Schlüssel ziehen.

Chosen-ciphertext attack Der Attacker kann Ciphertexte generieren und diese vom System entschlüsseln lassen. Der Attacker bekommt entweder den zum gewählten Ciphertext gehörenden Plaintext (und kann daraus potenziell Rückschlüsse auf den verwendeten Schlüssel machen) oder er bekommt nur Teilinformationen, wie zum Beispiel "Die Entschlüsselung konnte / konnte nicht durchgeführt werden".

Known-plaintext attack Der Attacker kennt sowohl Teile des Plaintext als auch den dazugehörenden Ciphertext (oder zumindest Teile davon). Er kann daraus Rückschlüsse auf andere Plaintexte oder gar den Schlüssel machen.

Chosen-plaintext attack Der Attacker kann Plaintexte wählen, welche er vom System verschlüsseln lassen will. Er erhält dann den dazugehörenden Ciphertext und kann daraus Rückschlüsse auf andere Plaintexte oder gar den Schlüssel machen.

Brute-force attack Der Attacker probiert alle möglichen Schlüssel aus, bis er den richtigen gefunden hat. Dass er den richtigen gefunden hat, erkennt er daran, dass der erhaltene Plaintext sinnvoll erscheint.

Grundsätzlich können alle Verschlüsselungsalgorithmen mittels brute-force Attacken geknackt werden. Ein wichtiges Evaluationskriterium eines Verschlüsselungsalgorithmus ist also, wie lange eine solche brute-force Attacke im Durchschnitt benötigt. Dieser Anzahl sagt man auch kryptographischer Work Factor.

Kryptographischer Work Factor

Work Factor Durchschnittliche Anzahl Versuche, bis der richtige Schlüssel gefunden wird.

Der kryptographische Work Faktor kann als Mass der Stärke für eine Verschlüsselung herangezogen werden. Der Work Faktor bezeichnet die durchschnittliche Anzahl Versuche, bis man den richtigen Schlüssel bei einer Brute Force Attacke gefunden hat.

Ein Algorithmus hat dann einen genügend grossen Work Faktor wenn es unrealistisch ist, den richtigen Schlüssel per brute force zu erraten.

Einflussfaktoren auf den Work-Factor

- **Verschlüsselungsalgorithmus:** Je nach verwendetem Algorithmus ist die Berechnung des Work Faktors anders. Wir werden in den entsprechenden Kapiteln jeweils den dazugehörenden Work Faktor auflisten.
- **Schlüssellänge:** Grundsätzlich gilt: Je länger der Schlüssel, desto höher der Work Faktor
- **Zufälligkeit des Schlüssels:** Der Work Faktor von einem Algorithmus mit einem Schlüssel fixer länge, ist dann maximal, wenn alle Schlüssel gleich wahrscheinlich sind. Sind die Schlüssel nicht gleich wahrscheinlich, wird der Attacker zuerst die wahrscheinlicheren Schlüssel ausprobieren und so durchschnittlich weniger Schlüssel ausprobieren müssen, bis er den richtigen Schlüssel gefunden hat. Daher ist es sehr wichtig, dass die gewählten Schlüssel immer zufällig sind.

Berechnung des Work-Factors als Hilfe zur Berechnung erfindet man ein unrealistisch schnelles System, welches 10^9 Chips hat, welcher jeder in $10^{-12}s$ einen Schlüssel ausprobieren kann.

durchschn. Zeit je nach grösse Work-Factor (Schlüssellänge):

- 2^{64}: $1.8 \cdot 10^{-2}$ Sekunden
- 2^{96}: $7.9 \cdot 10^{7}$ s, ca. 2.5 Jahre
- 2^{128}: $3.4 \cdot 10^{17}$ s, ca. 10^{10} Jahre
- 2^{256}: $1.2 \cdot 10^{56}$ s, ca. 10^{48} Jahre

Aus der Tabelle ersichtlich: Algorithmen mit Work-Factor $> 2^{128}$ als sicher betrachtet werden.

Den Work-Factor kann man auch in **bits** darstellen. Ein Work-Factor von 2^{128} entspricht einem Work-Factor von 128 bits. (2^n entspricht n bits)

Perfect Secrecy

Unter einem Algorithmus welcher Informationstheoretisch sicher ist, versteht man einen Algorithmus, bei dem man den ursprünglichen Plaintext nicht kennt, auch wenn man alle Schlüssel ausprobiert hat. Dazu sagt man auch der Algorithmus habe perfect secrecy. Aktuell gibt es nur einen einzigen Algorithmus, welcher diese Eigenschaft erfüllt, der One-Time-Pad (auch bekannt unter dem Namen Vernam Cipher).

One-Time-Pad

- **Voraussetzung:** Der Schlüssel ist komplett zufällig und genau gleich lang, wie die zu verschlüsselnde Nachricht.
- **Verschlüsselung:** Der Plaintext wird mit dem Schlüssel bitweise verxor-ed $c_j = p_j \oplus k_j \forall j \in \{1, \ldots, n\}$

Zur Erinnerung: Der XOR Operator ist definiert als:

p	k	$p \oplus k$
0	0	0
0	1	1
1	0	1
1	1	0

Perfect Secrecy Ein Verschlüsselungsalgorithmus hat perfect secrecy, wenn für alle möglichen Plaintexte p und Ciphertexte c und für alle Schlüssel k gilt:

$$P[p|c] = P[p] \tag{1}$$

Das heisst, dass die Wahrscheinlichkeit, dass ein bestimmter Plaintext p verschlüsselt wurde, gleich gross ist, wie die Wahrscheinlichkeit, dass ein beliebiger Plaintext p verschlüsselt wurde.

W enn ein Attacker jetzt alle Schlüssel ausprobieren würde, würde er neben vielen offensichtlich falschen Plaintexten auch viele plausible Plaintexte erhalten und wüsste daher nicht, welcher der plausiblen Plaintexte der korrekte ist.

Obwohl dies also genau das Verhalten ist, was wir möchten, kommt der One-Time-Pad nur in sehr seltenen Fällen zum Einsatz. Dies liegt daran, dass der Schlüssel

- genau gleich lang wie der zu verschlüsselnde Text sein muss
- komplett zufällig sein muss (vollständig zufällige Bitfolge)
- vorgängig geheim zwischen Sender und Empfänger ausgetauscht werden muss
- nur ein einziges mal verwendet werden darf (sonst ist er nicht mehr vollkommen zufällig und die ganze Sicherheit geht verloren)

Eigenschaften sicherer kryptographischer Algorithmen Da heute ausser dem One-Time-Pad kein Algorithmus bekannt ist, welcher als informationstheoretisch sicher gilt, werden folgende Eigenschaften für sichere Algorithmen definiert:

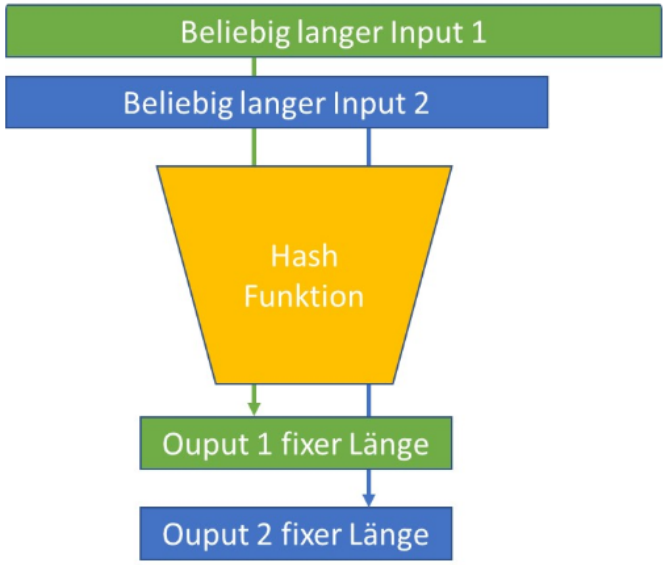
- Allgemein bekannt
- Keine Fehler bekannt
- Work Factor $> 2^{128}$

Daraus lässt sich auch ableiten, dass wir als Noobs keine kryptographischen Algorithmen selber entwickeln/implementieren sollten. Stattdessen soll auf standardisierte Algorithmen zurückgegriffen werden. (Öffentlich verfügbare Libraries)

Kryptographische Hash Funktionen

Kryptographische Hash Funktion ist eine mathematische Funktion mit folgenden Eigenschaften:

- aus einem beliebig langen Input wird Output mit konstanter Länge generiert
- es gibt keine Möglichkeit aus dem Output den Input wieder herzuleiten
- unterschiedliche Inputs ergeben mit sehr hoher Wahrscheinlichkeit völlig unterschiedliche Outputs, auch wenn sich die Inputs nur wenig unterscheiden
- es ist nicht möglich innert nützlicher Zeit zwei unterschiedliche Inputs zu generieren, welche denselben Output haben



Work Factor Da bei Hash Funktionen der Output weniger lang ist als der Input gibt es keinen Algorithmus, welcher für alle Inputs und Outputs die Eigenschaften 3 und 4 erreicht. Das heisst, es wird immer verschiedene Inputs geben, welche denselben Output generieren. Mit dem Work Faktor kann angegeben werden, wie gross der Aufwand ist, um diese Hash Collisions zu berechnen.
Der Work Faktor in bits einer kryptographischen Funktion entspricht der Hälfte der bits des generierten Outputs.
Auch hier gilt: Für eine sichere Hash Funktion sollte der Work Faktor mindestens 128 bit betragen.