

Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

24.04.2017

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

Was ist Sicherheit?



Quelle: Wikipedia

Was ist Sicherheit?



Quelle: Wikipedia

Was ist Sicherheit?



Quelle: Wikipedia

Was ist Sicherheit?



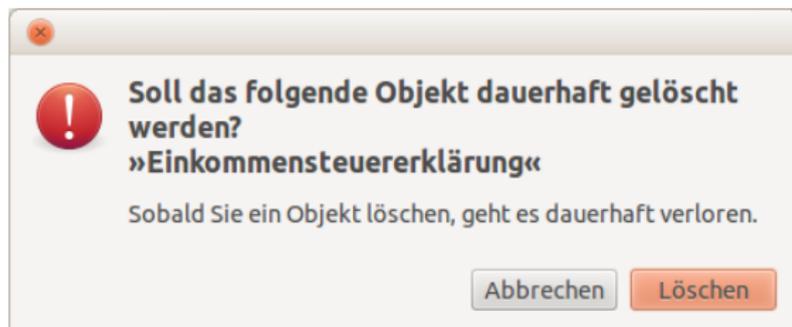
Quelle: Wikipedia

Was ist Sicherheit?



Quelle: Wikipedia

Was ist Sicherheit?



Was Sicherheit für uns ist

- Generell Sicherheit weit gefasster Begriff
- Unser Fokus: Sicherheit in der Informationstechnik
- Keine Definition, sondern Beispiele:
 - Schutz von Daten (z.B. E-Mails) vor unberechtigtem Zugriff
 - Zugriffskontrolle (z.B. Rechteverwaltung von Betriebssystem)
 - Privatsphäre in verteilten Systemen (z.B. sozialen Netzwerken)
- Beobachtung: *Bausteine* (z.B. Verschlüsselung) entscheidend
- Genaue Definition von Szenario abhängig

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

- Fokus: entscheidende *Bausteine* für sichere digitale Systeme
- Insbesondere: kryptographische Bausteine
- **Nicht:** vollständige Systeme
- Beispiele:
 - Verschlüsselungsverfahren (symmetrisch und asymmetrisch)
 - Authentifikationsmechanismen (z.B. Signaturen, Protokolle)
 - Strategien der Zugriffskontrolle (z.B. Bell-LaPadula)
- **Nicht**-Beispiele:
 - Sichere Softwareentwicklung (aber: ausgewählte Fehler)
 - Sichere Systementwicklung (z.B. durch Virtualisierung)
 - Schadsoftware (z.B. Viren)

- Orientiert an bisheriger Sicherheits-Vorlesung
- Augenmerk auf formaler Vorgehensweise
 - Definitionen/Beweise, wenn möglich
 - Aber: generell Überblickscharakter
 - Zur Auflockerung Anwendungen/Ausblick/aktuelle Forschung
- Schnittstelle für weiterführende Vorlesungen

- Vorlesungswebseite

<https://crypto.itl.kit.edu/index.php?id=798>

- Vorlesungsfolien (→ Webseite)
- Skript (→ Webseite)
 - **Achtung:** im Zweifel Aussagen aus der Vorlesung bindend
- Gelegentlich Verweise auf Webseiten
- Buchempfehlung (nur ausgewählte Kapitel):
 - Katz/Lindell: Introduction to Modern Cryptography
 - Anderson: Security Engineering (Link auf Webseite)
- Feedback-Kasten

- Übung: Wiederholung/Vertiefung/Anwendung von Vorlesung
- Mittel: Übungsblätter
 - Nicht bewertet, in Übung besprochen
 - Blatt ab etwa Mittwoch der Vorwoche verfügbar (→ Webseite)
- Übung donnerstags, etwa 14-tgl.
 - Erste Übung am 04.05.2017
- Übungsleiter
 - `Thomas.Agrikola@kit.edu`
- Klausur schriftlich, 60min
 - Klausur: Dienstag, 02.08.2017, 14:30
 - Nachklausur: Termin noch nicht fest
 - Altklausuren auf den Webseiten der vorherigen Vorlesungen

Fragen?

- ...Fragen?

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen**
- 4 Symmetrische Verschlüsselung
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

Motivation: zwei Vorgehensweisen

- „Arms race security“:
 - 1 Designer entwirft System
 - 2 Angreifer findet Schwachstelle
 - 3 Designer repariert Schwachstelle
 - 4 Goto 2
 - Theoretisch unbefriedigend
- „Provable security“:
 - 1 Designer entwirft **Sicherheitsmodell und System**
 - 2 **Designer versucht, System als sicher zu beweisen**
 - 3 Angreifer findet Schwachstelle **im Sicherheitsmodell**
 - 4 Designer repariert **Sicherheitsmodell und System**
 - 5 Goto 2
 - Unvollkommen, aber verlässlicher und theoretisch zugänglicher

- Zentraler Bestandteil von „Provable security“: Modularität
 - Subsysteme geben fest definierte Garantien
 - Beispiel: Chifftrate von Verschlüsselung nicht effizient unterscheidbar von Zufall
 - Sicherheitsgarantien größerer Systeme werden aus Sicherheitsgarantien von Subsystemen abgeleitet
- Deshalb folgendes Vorgehen:
 - Bausteine mit wohldefinierten Sicherheitsgarantien entwickeln
 - Danach komplexere Systeme aus Bausteinen zusammensetzen
 - Beispiel: TLS aus Schlüsselaustausch und Verschlüsselung zusammengesetzt
(Allerdings: TLS nicht das beste Beispiel für modulares Design)

Konkreter inhaltlicher Plan

- 1 Kryptographische Bausteine
 - Symmetrische Verschlüsselung (Stromchiffren, Blockchiffren)
 - Hashfunktionen (Eigenschaften, Konstruktion, Beispiele)
 - Public-Key-Verschlüsselung (RSA, ElGamal, sichere Varianten)
 - Authentifikation von Nachrichten (MACs, Signaturen)
- 2 Anwendungen: komplexere Protokolle
 - Schlüsselaustausch/-verteilung (Kerberos, TLS, Angriffe)
 - Identifikationsprotokolle (auch Zero-Knowledge-Protokolle)
- 3 Weitere Anwendungen:
 - Nutzerauthentifikation (Passwortabfragen, Angriffe, mehr)
 - Zugriffskontrollmechanismen
- 4 Häufige Sicherheitsprobleme (Buffer Overflows, XSS,  et al.)

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

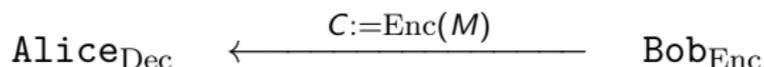
- Sichere Nachrichtenübermittlung auf unsicherem Kommunikationskanal:

Alice \xleftarrow{C} Bob

- Statt Nachricht M wird Chiffprat C übertragen
- Anforderungen:
 - Bob muss Chiffprat aus Nachricht berechnen
 - Alice muss Nachricht aus Chiffprat berechnen
 - Chiffprat soll „Außenseiter“ keinen Hinweis auf Nachricht geben

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren**
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

- Veraltet: Verschlüsselungsverfahren geheimhalten



- Chiffre wird durch Funktion Enc erzeugt: $C := \text{Enc}(M)$
- Nachricht mit Funktion Dec berechnen: $M = \text{Dec}(C)$
- *Wichtig*: nur Alice und Bob kennen Dec und Enc

Geheime Verfahren: Beispiel

Alice_{Dec} ← $C := \text{Enc}(M)$ Bob_{Enc}

■ Cäsar₃-Verschlüsselung:

- Interpretiere M als Zahlenfolge $M = (M_i)_{i=1}^n \in \{0, \dots, 25\}^n$
- Jede Zahl steht für einen Buchstaben ($A \leftrightarrow 0, \dots, Z \leftrightarrow 25$)
- Enc verschiebt jede Stelle von M zyklisch um 3:

$$\text{Enc}(M) := (C_i)_{i=1}^n \quad \text{mit} \quad C_i := M_i + 3 \bmod 26$$

- Dec verschiebt jede Stelle von M zyklisch um -3
 - Beispiel: $\text{Enc}(\text{KIT}) = \text{NLW}$
- ## ■ Fun fact: Cäsar₃ immer noch verwendet

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip**
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

- Problem bei geheimen Verfahren: unsicher, sobald Verfahren bekannt
- Auguste Kerckhoffs (1835–1903): parametrisierte Verfahren
 - Enc und Dec haben Schlüssel K als zusätzliche Eingabe
 - Enc und Dec dürfen bekannt sein, K muss geschützt werden
 - Vorteil: Enc und Dec von vielen Benutzern gleichzeitig nutzbar

(Vorteil nur konzeptionell: grundsätzlich sind parametrisierte Verfahren als unparametrisierte Verfahren interpretierbar)

Symmetrische Verfahren

- Traditionell: symmetrische Verschlüsselung

Alice_K ←^{C:=Enc(K,M)} Bob_K

- Alice und Bob besitzen gemeinsames Geheimnis K
- Chiffre wird durch Funktion¹ Enc erzeugt: $C := \text{Enc}(K, M)$
- Nachricht mit Funktion Dec berechnen: $M = \text{Dec}(K, C)$

¹Enc kann allerdings auch probabilistisch sein

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar**
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

Alice_{Dec} ← $C := \text{Enc}(K, M)$ Bob_{Enc}

■ Cäsar-Verschlüsselung:

- Interpretiere M als Zahlenfolge $M = (M_i)_{i=1}^n \in \{0, \dots, 25\}^n$
- Jede Zahl steht für einen Buchstaben ($A \leftrightarrow 0, \dots, Z \leftrightarrow 25$)
- Enc verschiebt jede Stelle von M zyklisch um K :

$$\text{Enc}(K, M) := (C_i)_{i=1}^n \quad \text{mit} \quad C_i := M_i + K \bmod 26$$

- Beispiel: $\text{Enc}(25, \text{KIT}) = \text{Enc}(-1, \text{KIT}) = \text{JHS}$
- **Frage:** Warum immer noch schlechte Idee?

Unsicherheit des Cäsar-Verfahrens

- Erinnerung: $\text{Enc}(K, M) = (M_i + K \bmod 26)_{i=1}^n$
- Problem: kleiner Schlüsselraum ($K \in \{0, \dots, 25\}$)
- Annahme: M „sinnvoller“ Text
(genauer: $K' \neq K \Rightarrow \text{Dec}(K', C)$ als unsinnig erkennbar)
- Dann: „Durchprobieren“ von $K' \in \{0, \dots, 25\}$ liefert K
- **Frage:** Was, wenn M kein sinnvoller Text?

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère**
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren

Generalisierung: das Vigenère-Verfahren

- Cäsar₃: $(C_1, C_2, \dots) = (M_1 + 3, M_2 + 3, \dots) \bmod 26$
- Cäsar: $(C_1, C_2, \dots) = (M_1 + K, M_2 + K, \dots) \bmod 26$
- Vigenère: $K \in \{0, \dots, 25\}^m$ und

$$(C_1, C_2, \dots) = (M_1, M_2, \dots, M_m, M_{m+1}, \dots) \\ + (K_1, K_2, \dots, K_m, K_1, \dots) \bmod 26$$

- Allgemein: $C_i = M_i + K_{(i-1 \bmod m)+1} \bmod 26$

Unsicherheit des Vigenère-Verfahrens

- Erinnerung:

$$(C_1, C_2, \dots) = (M_1, M_2, \dots, M_m, M_{m+1}, \dots) \\ + (K_1, K_2, \dots, K_m, K_1, \dots) \text{ mod } 26$$

- Beobachtung:

$$(C_1, C_{m+1}, \dots) = (M_1 + K_1, M_{m+1} + K_1, \dots) \text{ mod } 26$$

- Unterfolge $(C_1, C_{m+1}, C_{2m+1}, \dots)$ Cäsar-verschlüsselt mit K_1
- Ansatzpunkt für Kryptoanalyse (rate m , finde K_1, K_2, \dots)

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren**
 - One-Time-Pad
 - Stromchiffren

- Substitutionschiffren (Substitution von Buchstaben)
- Permutationschiffren (Reihenfolge ändern)
- Kombinationen der Konzepte (z.B. $\vec{C} = \mathbf{M} \cdot \vec{M}$)
 - Schlüssel $K = \mathbf{M}$ kann z.B. mit linearer Algebra gefunden werden, wenn Klartext-/Chiffrepaare bekannt

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad**
 - Stromchiffren

- Hauptproblem des Vigenère-Verfahrens: periodische Wiederverwendung des Schlüssels
- One-Time-Pad (auf Bits, d.h. $M \in \{0, 1\}^n$):
 - Schlüssel so lang wie Nachricht: $K \in \{0, 1\}^n$
 - $\text{Enc}(K, M) = C = M \oplus K \in \{0, 1\}^n$
 - $\text{Dec}(K, C) = C \oplus K$
- Wichtig: $K \in \{0, 1\}^n$ zufällig gleichverteilt gezogen

Sicherheit des One-Time-Pad

- Erinnerung: $\text{Enc}(K, M) = C = M \oplus K$
- Gegeben C ist jedes M möglich (und gleich wahrscheinlich)
- ⇒ OTP hat *informationstheoretische* Sicherheitseigenschaften
 - Selbst unbeschränkter Angreifer erhält durch C keine Information über M
 - Kann (und soll) formalisiert werden, heute aber noch nicht
- **Aber:** OTP oft missverstanden als Patentlösung

Unsicherheit des One-Time-Pad

- Erinnerung: $\text{Enc}(K, M) = C = M \oplus K$
- Nachteil: Schlüssel unhandlich (so lang wie Nachricht)
- ... zudem: Schlüssel darf nicht wiederverwendet werden
 - Aus $C = M \oplus K$ und $C' = M' \oplus K$ folgt $C \oplus C' = M \oplus M'$, eine nichttriviale Information über M und M'
- Problem: Chiffre verformbar ($C \oplus X = (M \oplus X) \oplus K$)
 - Beispiel: wenn $M \in \{\text{ja}, \text{nein}\}$, dann kann ein Chiffre C mittels $C \oplus (\text{ja} \oplus \text{nein})$ „semantisch“ verfälscht werden

- 1 Sicherheit
- 2 Struktur der Vorlesung
- 3 Generelle Motivation für Vorgehen
- 4 Symmetrische Verschlüsselung**
 - Ziel
 - Geheime Verfahren
 - Kerckhoffs' Prinzip
 - Cäsar
 - Vigenère
 - Weitere einfache Verfahren
 - One-Time-Pad
 - Stromchiffren**

- One-Time-Pad unhandlich (langer Schlüssel $K \in \{0, 1\}^n$)
- Idee: OTP-„Simulation“ mit kurzem $K \in \{0, 1\}^k$ (mit $k < n$)
 - Erweitere K zunächst: $K' := G(K) \in \{0, 1\}^n$ für geeignetes G
 - Verwende dann K' für OTP: $C := M \oplus K'$
 - Ziel: Pseudozufall $G(K)$ soll „aussehen“ wie echter Zufall
- Gesucht: geeignete Funktionen G
 - Wunsch: G soll guter Zufallsgenerator sein
- **Frage:** kann so Sicherheit (im selben Sinne wie beim OTP) gegen unbeschränkte Angreifer erzielt werden?

Struktur von Stromchiffren

- Verfahren hat internen Zustand $K^{(i)} \in \{0, 1\}^k$
- In jedem Schritt Ausgabe und Zustandsupdate:

$$\text{SC}(K) := (b, K') \in \{0, 1\} \times \{0, 1\}^k$$

- Extrahiere Bitfolge $G(K) := (b^{(1)}, \dots, b^{(n)})$ aus K :

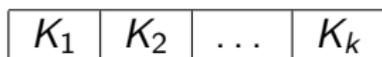
$$K^{(0)} := K$$

$$(b^{(i+1)}, K^{(i+1)}) := \text{SC}(K^{(i)})$$

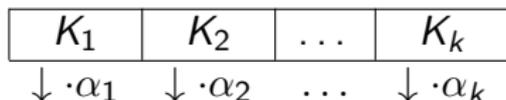
- Setze $C := M \oplus G(K)$

Beispiel: Schieberegister (LFSRs)

- Schlüssel K bitweise in Speicherzellen angeordnet:

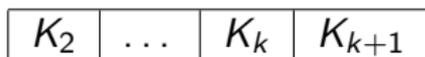


- Bei Zustandsupdate wird neues Bit erzeugt ($\alpha_i \in \{0, 1\}$):



$$\longrightarrow K_{k+1} := \sum_{i=1}^k \alpha_i K_i \text{ mod } 2$$

- Ausgabe ist $b := K_1$, neuer Zustand ist:



- Historisch interessant, aber **unsicher** (selbst wenn α_i geheim, siehe Vorlesung „symmetrische Verschlüsselungsverfahren“)
- LFSRs keine guten **Zufallsgeneratoren** (formalisierbar)

- Schnell (besonders in Hardware) implementierbar
- Gängige Konstruktionen mittels (mehrerer) LFSRs
- Oft algebraische Angriffe möglich
- Wegen Schlüsselupdate Synchronisation nötig
- Wie bei OTP Chiffre verformbar