Computersystemsicherheit

Lena Thuy Trang Vo

Wintersemester 2024/25

Inhaltsverzeichnis

Thema 1: Einführung

Themenübersicht

- Begriffsbedeutung
- Warum ist Sicherheit wichtig?
- Fallbeispiele für Sicherheitsvorfälle
- Sicherheitsprinzipien
  - Kenne die Angreifer
  - Berücksichtige menschliche Faktoren
  - Sicherheit ist wirtschaftliche Abwägung
  - Detektieren falls nicht verhinderbar
  - Defense in depth (gestaffelte Verteidigung)
  - Fail-safe Standard

Begriffsbedeutung

1.2.1 Was bedeutet Sicherheit?

Betriebssicherheit / Safety

- Schutz gegen Fehler/Unfälle
- Fehler meist unabsichtig verursacht
- Gegenmaßnahme: Verifikation, Testen

Angriffsicherheit/Security

- Schutz gegen worst-case Angreifer
- meist Schadabsicht
- Verifikation und Testen hilft wenig

Security und Safety können im Konflikt zueinander stehen
Beispiel Notausgang

- Safety: Im Notfall können Personen aus dem Gebäude
- Security: Für Gebäudeschutz am besten gar keine Tür

1.2.2 Sicherheitseigenschaften

Sicherheit kann vieles bedeuten...

1. Vertraulichkeit von Daten/Nachrichten (z.B. von Whatsapp Nachrichten)
   - Sicherstellung, dass das System keine unautorisierte Informationsgewinnung ermöglicht
2. Anonymität von Benutzern (z.B. beim Surfen im Web)
3. Integrität von Daten/Berechnungen (z.B. bei Überweisungen im Online-Banking)
   - Gewährleistung, dass nicht autorisierte Subjekte ein Objekt nicht unbemerkt ändern können
4. Authentizität von Dateien (z.B. Software-Updates)
   - Echtheit und Glaubwürdigkeit eines Objektes, die kryptografisch überprüfbar ist
5. Verfügbarkeit von Diensten (z.B. des Stromnetzes)
   - Gewährleistung, dass autorisierte Subjekte nicht in der Funktionalität beeinträchtigt werden

1.2.3 Wie können wir uns schützen? Allgemeine Sicherheitsprinzipien

Sicherheitsprinzipien

1. Kenne die Angreifer
2. Berücksichtige menschliche Faktoren
3. wirtschafltiche Faktoren beeinflussen Sicherheit
4. Detektieren falls nicht verhinderbar
5. Defense in depth (gestaffelte Verteidigung)
6. Fail-safe Standards

1. Kenne die Angreifer

Um ein effektives Bedrohungsmodell zu entwickeln, ist es wichtig, die **potenziellen Angreifer** und deren **Motivationen** zu verstehen.

Ressourcen:

- Individuum
- Organisierte Gruppen
- Terroristen
- staatlich geförderte Organisationen

Motivation:

- Geld
- politische Maßnahmen
- Vergeltung
- aus Spaß

Annahmen über Angreifer sind schwer zu treffen

- rechtzeitiges Erkennen von Angriffen schwierig: Angreifer kann unbemerkt mit dem System interagieren
- Angreifer kennt das System: Welches Betriebssystem wird verwendet, welche Hardware? (kennt Schwachstellen)
- Kann Glück haben: bei Chance 1:1.000.000 kann der Angreifer es 1.000.000 mal Probieren

2. Berücksichtige menschliche Faktoren

Einschränkung der Sicherheit durch menschliches Verhalten möglich

- als **Benutzer:in**
  - neigen dazu, Sicherheitsmechanismen zu umgehen, wenn diese die Nutzung erschweren
  - Beispiel: Wahl einfacher und wiederverwendeter Passwörter
- als **Programmierer:in**
  - Programmierer können Fehler machen, die Sicherheitslücken schaffen
  - benutzen Tools, die erkauben Fehler zu machen (z.B. Sprache ohne Typsicherheit)
- als **Angreifer:in**

  - Angreifer nutzen oft menschliche Eigenschaften wie Vertrauen oder Leichtgläubigkeit aus, um Informationen zu stehlen oder Zugang zu Systemen zu erlangen (Social Engineering)

Ergo: alle verwendeten Tools und Systeme sollten narrensicher sein.

3. wirtschaftliche Faktoren beeinflussen Sicherheit

- organisierte Cyberkriminialität nimmt zu
- Angrifffsziele von organisierter Cyberkriminalität: **wirtschaftliche Interessen**
  - sei es zur direkten finanziellen Bereicherung oder um einem Wettbewerber oder Land zu schaden
- aus Sicht der angreifenden Partei:
  - Angriff teurer als Belohnung $\longrightarrow$ kein Angriffsversuch
- aus Sicht der verteidigenden Partei:
  - viel Sicherheit kostet viel Geld
  - Abwägung zwischen Kosten-/Nutzen
    - Nutzen der Sicherheitsmaßnahmen proportional zu Kosten eines erfolgreichen Angriffs

4. Detektieren, falls nicht verhinderbar

1. **Abschrecken:** Einen Angriff abschrecken, bevor dieser stattfindet.
2. **Verhindern:** Falls Angriff stattfindet, verhindere dessen Erfolg
3. **Detektieren:** Stelle fest, falls ein Angriff stattgefunden hat
   - Falls nicht verhinderbar, dann wenigstens feststellen, dass ein Angriff stattgefunden hat
   - es ist essenziell, ihn schnell zu erkennen, um den Schaden zu minimieren
4. **Reagieren:** Reaktion auf stattgefundenen Angriff
   - Detektion ohne Reaktion ist nutzlos: Es ist entscheidend, nach der Erkennung eines Angriffs sofortige Maßnahmen zu ergreifen, um weitere Schäden zu verhindern.

5. Defense in Depth

- verschiedene Sicherheitsmaßnahmen implementieren
- schichtweiser Aufbau
  - Sicherheitsmaßnahmen übereinander legen, sodass ein Angreifer alle Schichten durchbrechen muss, um erfolgreich zu sein.
- Sicherheit ist oft weniger als die Summe aller Teile
  - Trotz der Vielzahl an Schutzmaßnahmen kann die Sicherheit oft nur so stark sein wie das schwächste Glied in der Kette.

6. Fail-Safe Standards

Dieses Prinzip sorgt dafür, dass ein System bei einem **Ausfall** oder einer Anomalie in einen Zustand übergeht, der den **geringstmöglichen Schaden** verursacht.

Beispiele:

mechanisches Zugsignal:
Bei einem mechanischen Zugsignal fällt das Signal auf "Halt", wenn das Zugseil reißt. Dies stellt sicher, dass Züge bei einem technischen Defekt automatisch gestoppt werden und keine Gefahr entsteht.

elektronisches Nummernschloss:
Bei einem elektronischen Nummernschloss ist es nicht immer einfach zu entscheiden, was der sichere Zustand ist. Bei einem Stromausfall könnte das Schloss entweder offen bleiben, um den Zugang zu ermöglichen, oder geschlossen bleiben, um unbefugten Zutritt zu verhindern. Die Entscheidung hängt von der spezifischen Anwendung und den damit verbundenen Risiken ab.

Thema 2: Einführung Kryptographie

Themenübersicht

- Was ist Kryptographie?
- Ziele der Kryptographie
- Klassiche Chiffren
- Ansätze der modernen Kryptographie

Was ist Kryptographie?

Kryptographie ist die Wissenschaft der **Verschlüsselung und Entschlüsselung** von Informationen. Sie dient dazu, Daten und Kommunikation vor unbefugtem Zugriff und Manipulation zu schützen.

- unzählige Anwendungen in der Praxis
  - grundlegender Baustein jedes Sicherheitssytems
  - z.B. ohne Kryptographie keine Sicherheit im Internet

Klassische vs. moderne Kryptographie

2.3.1 Klassische Kryptographie

- bezieht sich auf ältere Verschlüsselungsmethoden, die hauptsächlich zur **sicheren Kommunikation über unsichere Kanäle** verwendet wurden
- Hauptanwendung: Militär

2.3.2 Moderne Kryptographie

- bietet **starke Sicherheitsgarantien für Daten und Berechnungen**, selbst in Anwesenheit eines Angreifers
- wird in nahezu jedem Lebensbereich angewendet

Ziele der Kryptographie

1. Vertraulichkeit: Angreifer kann den Inhalt der Nachricht **nicht lernen**
   - nur autorisierte Parteien haben Zugang zu den Informationen, während unbefugte Dritte ausgeschlossen werden
2. Integrität: Angreifer kann Nachricht nicht ändern, ohne das Änderung bekannt wird
3. Authentizitä: Angreifer kann nicht die Nachricht von einer anderen Person stammen lassen
   - sicherstellen, dass der Absender einer Nachricht wirklich derjenige ist, für den er sich ausgibt

Schlüssel

- **Kurzer Schlüssel:** Kryptoverfahren verwenden zufällig gewählten, kurzen Schlüssel
- **Symmetrische Kryptographie:** Gleicher Schlüssel zum Ver- und Entschlüsseln
- **Asymmetrische Kryptographie:**
  - öffentlicher Schlüssel: z.B. im Internet veröffentlicht
  - geheimer Schlüssel: nur einzelnen Nutzern eines Kryptoverfahren bekannt

Kerkhoff'sches Prinzip

- ein Kryptoverfahren soll sicher bleiben, selbst wenn der Angreifer den Kryptoalgorithmus kennt
- alles ist öffentlich außer ein kurzer Schlüssel k, der zufällig gewählt wurde

Warum Kerckhoff?

- in kommerziell eingesetzten Produkten ist es schwar, die Spezifikation geheim zu halten
  - Reverse Engineering: Algorithmen können rekonstruiert werden
- kurze Schlüssel sind einfacher zu **schützen**, zu **erzeugen** und **auszutauschen**
- die Sicherheit des Designs kann **öffentlich analysiert** werden

Kerkhoff's Grundsatz verletzt $\Longrightarrow$ Sicherheit bei Verschleierung

Klassische Chiffren

Shift-Chiffre

- zyklischer Shift jedes Buchstaben um k Stellen im Alphabet
- Caesars Chiffre: $k = 3$
- alle Schlüssel ausprobieren bei Angriff $\longrightarrow$ **Brute-Force-Angriff**

Erweiterte Shift-Chiffre

Benutze Wort als Schlüssel und verschiebe jeden Buchstaben, um die durch den Schlüssel gegebene Differenz

Substitutionschiffre

- ist eine Erweiterung der Shift-Chiffre, bei der jeder Buchstabe des Alphabets durch einen anderen Buchstaben ersetzt wird, basierend auf einer **beliebigen Permutation** des Alphabets
- keine feste Verschiebung
- Anzahl an Schlüsseln: $26! \approx 2^{88} \Longrightarrow$ zu viele Schlüssel für ausprobieren

Moderne Kryptographie

2.8.1 Anwendung Heute

- Sichere Kommunikation im Internet
- Digitale Zertifikate
- Sichere Datenspeicherung, z.B. für die Cloud
- Zugriffskontrolle, z.B. als Autoschlüssel
- E-Commerce und Online-Banking
- Digitale Signaturen
- Hashfunktionen
- ...

2.8.2 Ansatz der modernen Kryptographie

1. **Formale Definition:**
   - Ziel des Angreifers: z.B bei Verschlüsselung sollte Angreifer nichts über Klartext lernen
   - Angreifermodell: Was kann der Angreifer tun und sehen (z.B. Angreifer sieht nur Chiffretexte)
2. **Konstruktion:**
   - z.B.Konstruktion komplexer Kryptoverfahren aus einfachen Kryptoprimitiven
3. **Sicherheitsbeweis:**
   - **Annahme hält** $\Longrightarrow$ Kryptoverfahren ist sicher gemäß der formalen Definition (z.B. zahlentheoretische Annahme)
   - Reduktionsbeweis: Angreifer gegen Kryptoverfahren $\Longrightarrow$ Annahme hält nicht

2.8.3 Was sind kryptographische Annahmen?

- Kryptoverfahren nutzen Annahmen, auf denen Sicherheit basiert
  - stellt sich heraus, dass Annahme falsch ist oder effizient gelöst werden kann, so wäre das Verfahren **nicht mehr sicher**
- **Einwegfunktion:** in eine Richtung einfach zu berechnen, in die andere praktisch unmöglich
- Annahme, praktisch unmöglich, Eingabe aus Ausgabe zu berechnen

Häufig genutzte Annahmen:

- Primfaktorzerlegung großer natürlicher Zahlen ist schwer
- Berechnung des Diskreten Logarithmus ist schwer
- Allgemein: Schwierigkeit mathematischer Probleme

2.8.4 Kryptographische Primitive und Konstruktionen

Kryptograhische Primitive

- ist eine **abstrakte, fundamentale Funktion** mit **spezifischen kryptographischen Eigenschaften**
- **Beispiele:** Blockchiffren, Hashfunktionen, Digitale Signaturen etc.

Kryptograhische Konstruktion

- beschreibt, wie **kryptographische Primitive instanziiert** und miteinander kombiniert werden, um **komplexe kryptographische Systeme** zu erstellen
- **Beispiele:** AES, SHA-256, Schnorr-Signaturen etc.

Thema 3: Symmetrische Kryptographie

Themenübersicht

- Was ist symmetrische Kryptographie?
- Definition Symmetrische Chiffre
- One-Time-Pad
- Block-Chiffren
- Modes of Operation

- Kryptographische Hashfunktionen
- Message Authentication Codes (MACs)
- Authenticated Encryption

Symmetrische Kryptographie: Es gibt nur **einen** Schlüssel für alle Algorithmen

Definition Symmetrischer Chiffren

3.2.1 Schutzziel

Welches kryptographische Schutzziel möchten wir mit symmetrischen Chiffren erreichen?

- **Vertraulichkeit:** Angreifer kann den Inhalt der Nachricht nicht lernen

3.2.2 Funktionale Definition

- beschreibt **Input/Output-Verhalten** der Algorithmen
- Algorithmen: Gen, Enc, Dec

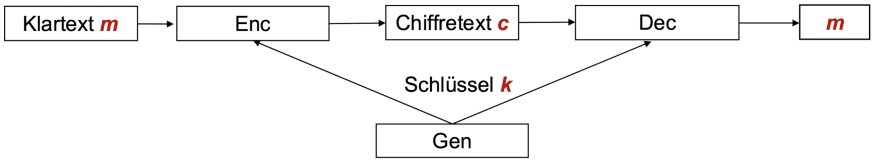

Abbildung 1: Funktionale Definition

- **Gen:** generiert einen zufälligen Schlüssel k, der später sowohl für die Verschlüsselung als auch für die Entschlüsselung verwendet wird
- **Enc (Verschlüsselung):** nimmt den Klartext m und Schlüssel k als Eingabe und erzeugt einen Chiffretext c
- **Dec (Entschlüsselung):** nimmt den Chiffretext und denselben geheimen Schlüssel als Eingabe und stellt den ursprünglichen Text wieder her

Korrektheit:
Die Entschlüsselung eines gültigen Chiffretextes resultiert in die original verschlüsselte Nachricht

$Dec(k, Enc(k, m)) = m$ für alle Nachrichten m und Schlüssel $k \longleftarrow Gen$

Effizienz: Verschlüsselung und Entschlüsselung sind effizient (1 GB/s)

3.2.3 Sicherheitsdefinition

- **Ziel des Angreifers:** Was ist ein erfolgreicher Angriff?
- naive Option: Angreifer lernt den Schlüssel k nicht
- in der Kryptographie: Angreifer lernt **nichts Neues** über m

Angreifermodell:

- beschreibt die Fähigkeiten und Ressourcen des Angreifers
- Angreifer lernt **nur Chiffretext** (known ciphertext attack), z.B. durch Abhören des Kanals
- Angreifer lernt **Paare von Klartexten/Chiffretexten** (known plaintext/ciphertext attack), z.B. bestimmter Teil der verschlüsselten Nachricht kann bekannt sein
- Angreifer **wählt Klartexte und lernt zugehörige Chiffretexte** (chosen plaintext attack), z.B. Angteifer überzeugt Challenger davon, Nachrichten seiner Wahl zu verschlüsseln

3.2.4 Sicherheitsspiel (IND-CPA)

- Sicherheit wird in der Kryptographie durch sein **Spiel** zwischen **Angreifer** und **Challenger** definiert

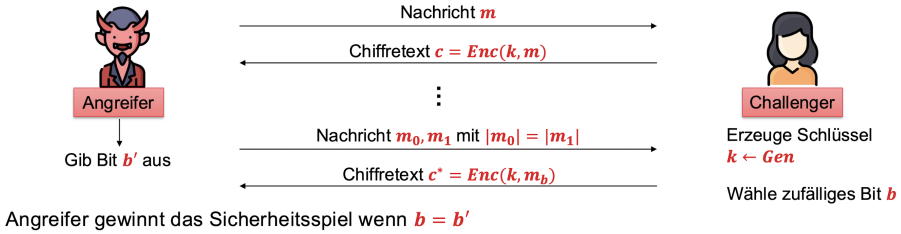

Abbildung 2: Sicherheitsspiel

Sicherheit:

Symmetrische Chiffre ist **IND-CPA sicher**, falls alle **effizienten** Angreifer das Sicherheitsspiel maximal mit der Wahrscheinlichkeit $\approx \frac{1}{2}$ gewinnen können

Was bedeutet effizient?

- effizient: Laufzeit **polynomiell** in der Schlüssellänge
- nicht effizient: Laufzeit **exponentiell** in der Schlüssellänge

3.2.5 Bietet IND-CPA die stärkste Sicherheit?

- Nein, es gibt noch stärkere
- IND-CPA bietet grundlegenden Schutz gegen passive Angriffe (nur Zugriff auf Verschlüsselungen)
- Gefahr : Chosen Ciphertext Angriff
  - Angreifer hat auch Zugang zu **Entschlüsselung** von (bestimmten) Chiffretexten
  - Angreifer kann diese Informationen nutzen, um **sensitive Informationen** zu erlangen
- Beispiel: **Padding Orakel Angriff**
  - Angreifer kann durch gezielte **Entschlüsselungsanfragen** Informationen über den Klartext gewinnen

3.2.6 Stärkere Sicherheit: IND-CCA

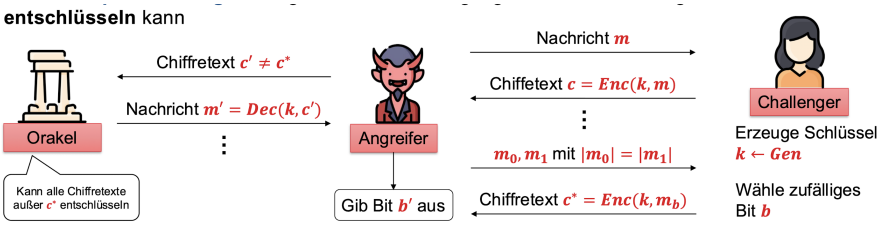

Abbildung 3: Sicherheitsspiel

- **Chosen Ciphertext Angriff**
- geht einen Schritt weiter als IND-CPA und schützt auch vor Angreifern, die zusätzlich die Möglichkeit haben, **bestimmte Chiffretexte zu entschlüsseln**

3.2.7 Unterschied zwischen IND-CPA und IND-CCA

- IND-CPA Sicherheit:
  - schützt vor Angriffen, bei denen Angriefer **Verschlüsselungen** seiner Wahl erzeugen kann
  - reicht nicht aus, wenn Angreifer auch **Zugriff auf Entschlüsselungen** hat
- IND-CCA Sicherheit:
  - stärkere Sicherheit
  - schützt selbst dann, wenn Angreifer zusätzlich **Entschlüsselungen anfordern** kann (außer dem zu entschlüsselnden Chiffretext)
- Fazit
  - IND-CCA Sicherheit ist notwendig, wenn Angreifer auf **Entschlüsselungsoperationen zugreifen kann**

Wie zeigen wir (Un-)sicherheit?

- Unsicheres Verfahren
  - konstruiere effizienten Angreifer, der mit mit Wahrscheinlichkeit signifikant größer als $\frac{1}{2}$
- Sicheres Verfahren:
  - Zeige, dass **alle Angreifer** das Sicherheitsspiel mit Wahrscheinlichkeit $\approx \frac{1}{2}$ gewinnen
  - Reduktionsbeweis auf Annahmen

One-Time-Pad Verschlüsselung

Wiederholung: XOR

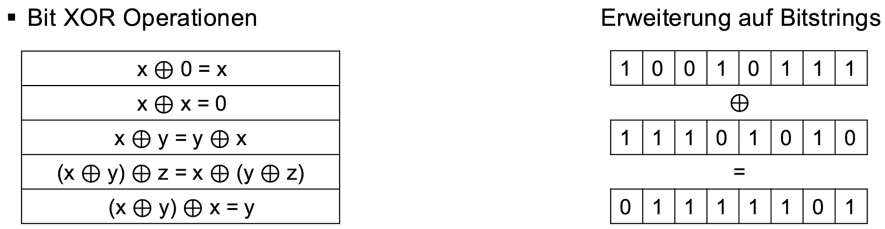

Abbildung 4: XOR

3.3.1 One-Time-Pad

- auch häufig **Vernam-Chiffre** genannt
- symmetrisches Verschlüsselungsverfahren
- zur Verschlüsselung von **Bitstrings der Länge** n

Funktionsweise

- Gen: in zufälliger **Schlüssel k** wird aus der Menge $\{0,1\}^n$ erzeugt, wobei n die Länge des Klartextes m ist
  - Schlüssel muss mindestens so lang wie der Klartext sein und darf nur einmal verwendet werden

Enc: Verschlüsselung erfolgt durch eine **XOR-Operation** zwischen dem **Klartext m** und dem **Schlüssel k**

$\text{Enc}(k, m) = k \oplus m \quad \Longrightarrow$ Ergebnis ist der **Chiffretext c**

- Dec: um den Chiffretext zu **entschlüsseln** wird erneut eine **XOR-Operation** zwischen dem **Chiffretext c** und dem **Schlüssel k** durchgeführt

  $\text{Dec}(k, c) = k \oplus c$

- da bein XOR die Operation **invertierbar ist** $(x \oplus x = 0)$, erhält man den ursprünglichen Klartext zurück.

3.3.2 Sicherheit

- beschränktes Sicherheitsspiel: Angriefer erhält **keinen Zugriff auf Chiffretexte**
- Angreifer gewinnt das Sicherheitsspiel, wenn $b = b'$
- Analyse ergibt: Perfekte Sicherheit - $c*$ gibt keine Information über m_b preis

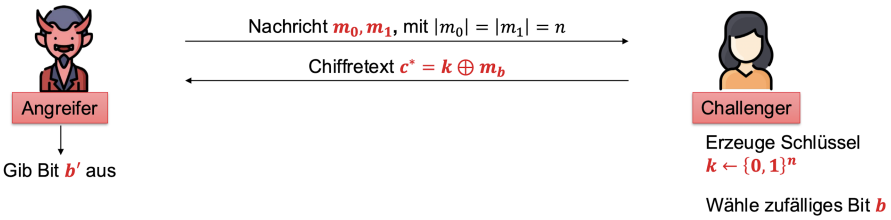

Abbildung 5: beschränktes Sicherheitsspiel

3.3.3 Schlüssel nur einmal verwenden

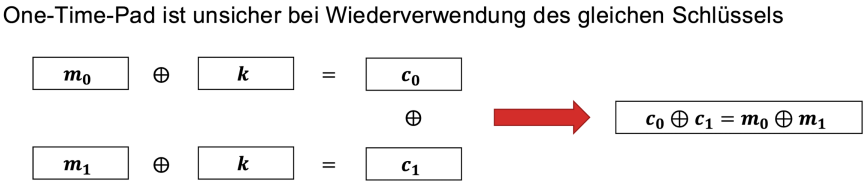

Abbildung 6: One-Time-Pad Schlüssel nur einmal verwenden

- Lernen des XORs kann nützliche Informationen preisgeben
- z.B. welche Bits von m_0 und m_1 gleich sind
- wenn m_0 bekannt ist, dann ist auch m_1 bekannt (und vice versa)
- **Moral:** zur Verschlüsselung jeder Nachricht muss ein neuer zufälliger Schlüssel gewählt werden

3.3.4 Venona-Projekt: Risiken von One-Time-Pads

- **Venona-Projekt (1943-1980)**
  - Ziel: Entschlüsselung sowjetischer Kommunikation durch USA und Großbritannien
  - Sowjetische Nachrichten wurden mit One-Time-Pads verschlüsselt

  - Fehler: Schlüssel wurden unter Zeitdruck mehrfach verwendet

- **Risiken von One-Time-Pads**
  - Mehrfachnutzung von Schlüsseln: Führt dazu, dass Angreifer durch XOR der Chiffretexte Informationen über die Klartexte gewinnen können
  - US-Geheimdienste nutzten diesen Fehler, um sowjetische Nachrichten zu entschlüsseln
- **Lektion:** One-Time-Pad ist nur sicher, wenn jeder **Schlüssel einmalig** verwendet wird

3.3.5 Nachteile von One-Time-Pad

1. Schlüssel ist so lang wie Nachricht
   - für große Mengen von Daten müssen **lange zufällige Schlüssel** gespeichert und ausgestauscht werden
   - gute Zufälligkeit zu erzeugen, ist sehr aufwendig
2. Schlüssel kann nur einmal benutzt werden
   - mehrfache Verwendung kann etwa über Klartexte preisgeben
   - für viele Nachrichten benötigt man viele Schlüssel
3. Sicheheit im beschränkten Angreifermodell
   - Chiffretext-Only Angriffe