

Kryptographie und Komplexität

Einheit 2.3

One-Time Pads und Perfekte Sicherheit



1. Perfekte Geheimhaltung
2. One-Time Pads
3. Strombasierte Verschlüsselung

WIE SICHER KANN EIN VERFAHREN WERDEN?

- **Ziel ist (nahezu) perfekte Sicherheit**
 - Klartext eines Schlüsseltextes ist ohne Schlüssel niemals zu ermitteln auch wenn Angreifer beliebig viel Zeit und Rechnerkapazität hat
- **Wie präzisiert man perfekte Sicherheit?**
 - Schlüsseltext enthält **keine Information über zugehörigen Klartext**
 - Jeder mögliche Klartext könnte zu diesem Schlüsseltext passen
 - Zugehörige Schlüssel sind alle gleich wahrscheinlich
 - Eve kann nicht wissen, welcher Schlüssel tatsächlich benutzt wurde
 - Keine Frage der Komplexität sondern des Informationsgehalts
 - Präzisierung benötigt **Wahrscheinlichkeits- und Informationstheorie**
- **Kann perfekte Sicherheit erreicht werden?**
 - Möglich wenn Schlüssel **perfekt zufällig** und **so groß wie Klartext**
 - Unrealistischer Aufwand – reale Verfahren sind niemals perfekt

● Ereignis

- Menge möglicher Ergebnisse eines Zufallsexperimentes

z.B. Erstes Symbol eines Textes ist ein Y:

$$E = \{Y\}$$

Würfel zeigt eine ungerade Zahl:

$$E = \{1, 3, 5\}$$

- Menge S aller möglichen Ergebnisse (**Elementarereignisse**) nicht leer
- **Sicheres Ereignis**: $E = S$ (z.B. Würfel zeigt Zahl zwischen 1 und 6)
- **Leeres Ereignis**: $E = \emptyset$ (z.B. Würfel zeigt eine Zahl größer als 6)
- Ereignisse A und B **schließen sich gegenseitig aus**, wenn $A \cap B = \emptyset$

● Wahrscheinlichkeitsverteilung auf S

- Abbildung $Pr : \mathcal{P}(S) \rightarrow \mathbb{R}$, die jedem Ereignis eine Zahl zuordnet mit
 - $0 \leq Pr(E) \leq 1$ für alle $E \subseteq S$
 - $Pr(\emptyset) = 0$ und $Pr(S) = 1$
 - $Pr(A \cup B) = Pr(A) + Pr(B)$, falls A und B sich ausschließen
- $Pr(E)$ ist die **Wahrscheinlichkeit** des Ereignisses E

● Eigenschaften von Wahrscheinlichkeiten

- $Pr(A) \leq Pr(B)$, falls $A \subseteq B$
- $Pr(S \setminus A) = 1 - Pr(A)$
- $Pr(\bigcup_{i=1}^n A_i) = \sum_{i=1}^n Pr(A_i)$, falls alle A_i sich paarweise ausschließen
- $Pr(A) = \sum_{a \in A} Pr(a)$, für alle $A \subseteq S$ ($Pr(a)$ steht kurz für $Pr(\{a\})$)
- Wahrscheinlichkeitsverteilungen sind durch die Wahrscheinlichkeiten der Elementarereignisse eindeutig definiert

● Gleichverteilung

- Wahrscheinlichkeitsverteilung mit $Pr(a) = Pr(b)$ für alle $a, b \in S$
- Für endliche Mengen S ist $Pr(a) = 1/|S|$ und $Pr(A) = |A|/|S|$
 - z.B. perfekte Würfel: $Pr(i) = 1/6$ für alle $i \in \{1..6\}$
- Verteilung von Buchstaben im Text ist keine Gleichverteilung

- **Bedingte Wahrscheinlichkeit $Pr(A|B)$**

- Wahrscheinlichkeit, daß Ereignis A auftritt, wenn B bekannt ist
z.B. Wahrscheinlichkeit des Klartextes x , wenn Schlüsseltext y vorliegt

$$Pr(A|B) = Pr(A \cap B) / Pr(B)$$

- Wahrscheinlichkeit, daß Würfel eine 4 zeigt, wenn sicher ist, daß die angezeigte Zahl gerade ist, ist $1/3$
- Wahrscheinlichkeit, daß Klartext einer Verschiebechiffre **ENDE** ist, wenn Schlüsseltext **ABCD** vorliegt, ist 0

- **Unabhängigkeit von Ereignissen A und B**

- $Pr(A|B) = Pr(A)$: Wahrscheinlichkeit für A hängt nicht von B ab
z.B. Ergebnis eines zweiten Würfels hängt nicht vom ersten Wurf ab
- Äquivalent zu $Pr(A \cap B) = Pr(A)Pr(B)$
- Die Wahrscheinlichkeit, daß mehrere unabhängige Ereignisse gleichzeitig auftreten, ist das Produkt der Einzelwahrscheinlichkeiten

MATHEMATIK: WAHRSCHEINLICHKEITSTHEORIE (IV)

• Satz von Bayes:

- $Pr(B|A) = Pr(B)Pr(A|B)/Pr(A)$, falls $Pr(A) > 0$

Einfache Rechnung: $Pr(B|A) = Pr(B \cap A)/Pr(A) = Pr(B)Pr(A|B)/Pr(A)$

- Wahrscheinlichkeit eines Klartextes x bei Vorliegen des Schlüsseltextes y ergibt sich aus Wahrscheinlichkeit, daß x zu y verschlüsselt wird

• Geburtstagsparadox

- Wieviele Personen benötigt man in einem Raum, damit mit großer Wahrscheinlichkeit zwei am gleichen Tag Geburtstag haben?
- Wieviele Klartext-/Schlüsselpaare braucht man, um mit hoher Wahrscheinlichkeit mehrmals denselben Schlüsseltext zu generieren?

Analyse: bei k Personen, n Geburtstagen gibt es n^k Elementarereignisse

$(g_1, \dots, g_k) \in \{1..n\}^k$ mit Wahrscheinlichkeit $1/n^k$

Die Wahrscheinlichkeit p , daß alle g_i verschieden sind, ist $\prod_{i=0}^{k-1} (n-i)/n^k$

Wegen $1 + x \leq e^x$ ist p maximal $e^{\sum_{i=0}^{k-1} (-i/n)} = e^{-k(k-1)/(2n)}$

Für $k \geq 1/2 + \sqrt{1/2 + 2n \cdot \ln 2} = 22.9999$ ist $p \leq 1/2$ (Für $k \geq 42$ ist $p \leq 0.1$!)

Mit Wahrscheinlichkeit 50% haben 2 von 23 Personen denselben Geburtstag

● Informationsgehalt von Nachrichten

- $Pr_{\mathcal{P}}$: Wahrscheinlichkeitsverteilung der Klartexte
Abhängig von **Sprache** und **Thematik** (Bank, Uni, Militär,..)
- $Pr_{\mathcal{K}}$: Wahrscheinlichkeitsverteilung der Schlüssel
Unabhängig von $Pr_{\mathcal{P}}$ aber ggf. abhängig von verwendetem System
- $Pr(x, K) := Pr_{\mathcal{P}}(x)Pr_{\mathcal{K}}(K)$
Wahrscheinlichkeit der Verschlüsselung von $x \in \mathcal{P}$ mit $K \in \mathcal{K}$
Spezialfälle: $Pr(x) := Pr(x, \mathcal{K})$, $Pr(K) := Pr(\mathcal{P}, K)$
- $Pr(y) := Pr(\{(x, K) \mid e_K(x) = y\}) = \sum_{K \in \mathcal{K}} Pr(d_K(y))Pr(K)$
Wahrscheinlichkeit, daß eine Verschlüsselung den Schlüsseltext y ergibt

● Perfekte Geheimhaltung eines Kryptosystems

- Kein Schlüsseltext sagt etwas über den zugehörigen Klartext aus
Mathematisch: Für alle $x \in \mathcal{P}, y \in \mathcal{C}$ ist $Pr(x|y) = Pr(x)$
Mit dem Satz von Bayes auch: $Pr(y) = Pr(y|x) = Pr(\{K \mid e_K(x)=y\})$

Ein einfaches Beispielsystem

Wahrscheinlichkeiten und Verschlüsselung durch Tabelle gegeben

$Pr_C \backslash Pr_K$		K_1	K_2	K_3	K_4	K_5
		.2	.4	.1	.2	.1
A	.2	1	2	3	4	5
B	.5	2	3	4	6	1
C	.2	3	4	6	5	2
D	.1	4	5	1	2	3

$$\mathcal{P} = \{A, B, C, D\},$$

$$\mathcal{C} = \{1, 2, 3, 4, 5, 6\}$$

$$\mathcal{K} = \{K_1, K_2, K_3, K_4, K_5\}$$

Schlüsselwahrscheinlichkeit unabhängig von Klartextwahrscheinlichkeit

– Wahrscheinlichkeiten der Schlüsseltexte:

$$\cdot Pr(1) = Pr(\{(A, K_1), (B, K_5), (D, K_3)\}) = .04 + .05 + .01 = .10$$

$$\cdot Pr(2) \dots Pr(6) = .22, .27, .19, .10, .12$$

– Wahrscheinlichkeiten der Klartexte bei bekannten Schlüsseltexten:

$$\cdot Pr(A|1) = Pr(\{(A, 1)\}) / Pr(1) = .04 / .10 = .40$$

$$\cdot Pr(B|1) \dots Pr(D|1) = .50, .00, .10$$

$$\cdot Pr(A|2) \dots Pr(D|2) = .364 (8/22), .454, .091, .091$$

Keine perfekte Geheimhaltung, da i.a. $Pr(x|y) \neq Pr(x)$

PERFEKT SICHERE KRYPTOSYSTEME

- **Die Verschiebechiffre ist perfekt geheim**

.. aber nur, wenn jeder Schlüssel mit gleicher Wahrscheinlichkeit vorkommt und das Chiffrierverfahren für jeden Buchstaben neu gestartet wird

- **Beweis:**

– Wegen $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{27}$ gilt für jedes $x \in \mathcal{P}, y \in \mathcal{C}$

$$Pr(y|x) = Pr(\{K | x+_n K=y\}) = Pr(y-_n x) = 1/27 \text{ und}$$

$$\begin{aligned} Pr(y) &= \sum_{K \in \mathcal{K}} Pr(d_K(y)) Pr(K) = \sum_{K \in \mathcal{K}} Pr(y-_n K)/27 \\ &= \sum_{x \in \mathcal{P}} Pr(x)/27 = 1/27 \end{aligned}$$

– Da beide Werte gleich sind, ist die Verschiebechiffre perfekt sicher selbst wenn keine Gleichverteilung der Klartexte vorliegt

- **Was sind die Kernargumente des Beweises?**

- $Pr(y|x)$: Für alle $x \in \mathcal{P}, y \in \mathcal{C}$ gibt es genau einen Schlüssel mit $e_K(x)=y$
- $Pr(y)$: $Pr(K)$ ist eine Konstante (Schlüssel sind gleichverteilt)
- Klartext- und Schlüsselmenge sind gleich groß und endlich

PERFEKTE SICHERHEIT: DER SATZ VON SHANNON

Ein Kryptosystem mit $|\mathcal{K}|=|\mathcal{P}|=|\mathcal{C}|<\infty$ und $Pr(x)>0$ für alle $x \in \mathcal{P}$ ist genau dann perfekt geheim, wenn die Schlüssel gleichverteilt sind und für alle $x \in \mathcal{P}, y \in \mathcal{C}$ genau ein Schlüssel $K \in \mathcal{K}$ mit $e_K(x)=y$ existiert

$\Rightarrow$: Wir nehmen an, das Kryptosystem sei perfekt geheim

- Gäbe es für ein $x \in \mathcal{P}, y \in \mathcal{C}$ keinen Schlüssel $K \in \mathcal{K}$ mit $e_K(x)=y$, dann wäre $Pr(x|y)=0 \neq Pr(x)$. Also gibt es mindestens ein K mit $e_K(x)=y$
Wegen $|\mathcal{K}| = |\mathcal{C}|$ gibt es dann **genau einen Schlüssel mit $e_K(x)=y$**
- Sei $K_x(y)$ der eindeutige Schlüssel K mit $e_K(x)=y$
Wegen $|\mathcal{K}| = |\mathcal{P}|$ gilt $\{K_x(y) \mid x \in \mathcal{P}\} = \mathcal{K}$ für jedes $y \in \mathcal{C}$ und
 $Pr(y) = Pr(y|x) = Pr(\{K \mid e_K(x)=y\}) = Pr(K_x(y))$ für alle $x \in \mathcal{P}$
Damit haben alle Schlüssel die gleiche Wahrscheinlichkeit

$\Leftarrow$: Wir zeigen die Umkehrung

- Es gilt $Pr(y|x) = Pr(\{K \mid e_K(x)=y\}) = Pr(K_x(y)) = 1/|\mathcal{K}|$
und $Pr(y) = \sum_{x \in \mathcal{P}} Pr(x)Pr(K_x(y)) = \sum_{x \in \mathcal{P}} Pr(x)/|\mathcal{K}| = 1/|\mathcal{K}|$
für alle $x \in \mathcal{P}, y \in \mathcal{C}$. **Also ist das Kryptosystem perfekt geheim**

ONE-TIME PADS

Perfekte Geheimhaltung mit großem Aufwand

- **Einfaches Verschlüsselungsverfahren**

(©Vernam, 1917)

- Bei n -bit Texten wähle $\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0, 1\}^n = \mathbb{Z}_2^n$
- Ver-/entschlüssele bitweise: $e_K(x) = x \oplus K$, $d_K(y) = y \oplus K$
- Schlüssel werden zufällig (mit Gleichverteilung) gewählt
- **Perfekte Geheimhaltung** folgt aus Satz von Shannon

- **Nicht wirklich praktikabel**

- Jede neue Nachricht braucht **neuen Schlüssel gleicher Größe**
 - Wiederverwendung ermöglicht known plaintext Attacke ($K = x \oplus y$)
- Schlüssel muß separat ausgetauscht werden
 - **Hoher Speicheraufwand** für Lagerung von Schlüsseln
- Verwendung wenn Sicherheitsanforderungen hohe Kosten rechtfertigen

- **Wie erzeugt man Zufallszahlen?**

- **Hardware-Zufallsbit Generatoren**: physikalische Quellen (Radioaktivität)
- **Software-Zufallsbit Generatoren**: Zeit zwischen Keyboardanschlägen
- **Pseudozufallszahlen**: algorithmisch erzeugte Zahlen (effizienter)

Systematisch erzeugte “One-Time Pads”

- **Generiere “zufälligen” Schlüsselstrom $k_1 k_2 k_3 \dots$**

- Verschlüsselung: $e_K(x_1 x_2 \dots x_n) = e_{k_1}(x_1) e_{k_2}(x_2) \dots e_{k_n}(x_n)$

- Entschlüsselung: $d_K(y_1 y_2 \dots y_n) = d_{k_1}(y_1) d_{k_2}(y_2) \dots d_{k_n}(y_n)$

- Schlüssel $k_1 \dots k_n$ wird systematisch aus Anfangsschlüssel K berechnet

- **Berechnung des Schlüsselstroms**

- Anfangsschlüssel K und bisherige Klartextfragmente können eingehen

- $k_i = f(K, x_1 \dots x_{i-1})$ für eine feste Schlüsselerzeugungsmethode f

- Alice berechnet $k_1 = f(K, \epsilon)$, $y_1 = e_{k_1}(x_1)$, $k_2 = f(K, x_1)$, $y_2 = e_{k_2}(x_2)$, ...

- Bob berechnet $k_1 = f(K, \epsilon)$, $x_1 = e_{k_1}(y_1)$, $k_2 = f(K, x_1)$, $x_2 = e_{k_2}(y_2)$, ...

- Alice und Bob müssen nur den Anfangsschlüssel K austauschen

- Schlüsselaustausch erheblich einfacher als bei One-Time Pads

- Effiziente Ausführung und große Diffusion und Konfusion möglich

KLASSIFIZIERUNG VON STROM CHIFFREN

- **Asynchrone Erzeugung des Schlüsselstroms**

- Klartext wird in Schlüsselerzeugung mit einbezogen
z.B. letzter Klartextblock wird Schlüssel für nächsten Block

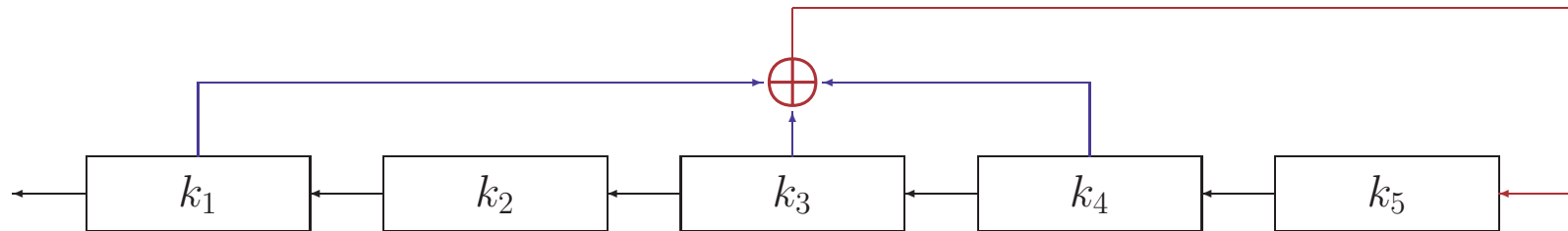
- **Synchrone Erzeugung des Schlüsselstroms**

- Keine Abhängigkeit des Schlüsselstroms vom Klartext
- Schlüsselstrom wird ausschließlich aus Basisschlüssel K erzeugt
z.B. Fibonaccizahlen modulo n : 1 2 3 5 8 13 21 7 1 8 9 17 26 16 15 ...

- **Periodische Erzeugung des Schlüsselstroms**

- Teilschlüssel wiederholen sich mit Periode m : $k_{i+m} = k_i$ für alle i
- Blockchiffren der Länge m sind Schlüsselströme mit Periode m
- Gewichtete Summe $k_{i+m} = \sum_{j=0}^{m-1} c_j k_{i+j} \bmod n$ der letzten m Schlüssel (mit Anfangsschlüssel $K = k_1..k_m, c_0..c_{m-1}$)
kann einen Schlüsselstrom der Periode $n^m - 1$ liefern

LSFR STROMCHIFFRE



- **Verwende Lineares Feedback Shift Register**

- Periodische Stromchiffre mit Anfangsschlüssel $K = k_1..k_m, c_0..c_{m-1}$
- In jeder Phase verwende k_1 als aktuellen Schlüssel und berechne $k'_i := k_{i+1}$ (Shift) und $k'_m := \sum_{j=0}^{m-1} c_j k_{j+1} \bmod n$ (Lineares Feedback)
- Kann für $n=2$ sehr effizient mit Hardwareregistern realisiert werden
- Liefert bei guten Anfangsschlüsseln einen Strom der Periode $2^m - 1$

- **Anwendungsbeispiel**

- Anfangsschlüssel $K = 10000, 10100$ liefert den Schlüsselstrom
1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0 0 0 ...
- Anfangsschlüssel $K = 10000, 10110$ liefert den Schlüsselstrom
1 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0 0 0 ... (Periode 12!)

AUTOKEY CHIFFRE

Einfacher asynchroner Strom Chiffre

- **Vigenere Chiffre mit “Klartext als Schlüssel”**

- Wähle $k_1 = K$ (der geheime Schlüssel) und setze $k_{i+1} = x_i$

- $$e_K(x_i) = x_i + k_i \bmod n, \quad d_K(y_i) = y_i - k_i \bmod n$$

- ENDE UM ELF $\hat{=}$ [4;13;3;4;26;20;12;26;4;11;5]

- liefert mit $K=3$ [3;4;13;3;4;26;20;12;26;4;11] als Schlüsselstrom

- und ergibt [7;17;16;7;3;19;5;11;3;15;16] $\hat{=}$ HRQHDTFLDPQ

- **Relativ sicher gegenüber statistischen Analysen**

- Regelmäßigkeit des Alphabets wird aufgehoben

- Brute-Force Attacken durch längere Anfangsschlüssel vermeidbar

- Wähle $K = k_1..k_m$ und setze $k_{i+m} = x_i$

WIE SICHER SIND STROMCHIFFREN?

- **Stromchiffren erzeugen beliebig lange Schlüssel**
 - 32-bit Anfangsschlüssel liefern “One-Time Pad” für 500MB Daten
 - Eine wichtige Voraussetzung von Shannons Theorem ist erfüllt
 - Liefern Stromchiffren nahezu perfekte Sicherheit?
- **Große Schlüssel alleine reichen nicht**
 - Stromchiffren erzeugen keinen echten Zufall (keine Gleichverteilung)
 - Stromchiffren können nicht jeden 500MB großen Schlüssel erzeugen
 - Pro Klartext kann es nicht mehr Schlüssel als Anfangsschlüssel geben
 - Es können nicht alle möglichen Schlüsseltexte erzeugt werden
 - Beide Annahmen von Shannons Theorem sind verletzt
- **Stromchiffren können attackiert werden**
 - Schlüssel- enthalten zu viele Regelmäßigkeiten
 - Schlüsseltexte enthalten wertvolle Strukturinformation für Angreifer

• Known plaintext Attacke

- Zur Bestimmung des Anfangsschlüssels $K = k_1..k_m, c_0..c_{m-1}$ benötigt Eve nur ein Klar-/Schlüsseltextpaar $(x_1..x_{2m}, y_1..x_{2m})$ der Länge $2m$
- Wegen $y_i = x_i \oplus k_i$ ist $k_i = x_i \oplus y_i$ für alle i leicht zu berechnen
- Wegen $k_{m+i} := \sum_{j=0}^{m-1} c_j k_{j+i} \bmod 2$ hat Eve m lineare Gleichungen:

Für $Z := \begin{pmatrix} k_1 & k_2 & \dots & k_m \\ k_2 & k_3 & \dots & k_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ k_m & k_{m+1} & \dots & k_{2m-1} \end{pmatrix}$ gilt $(k_{m+1}..k_{2m}) = (c_0..c_{m-1}) \star_2 Z$

und da Z invertierbar ist, folgt $(c_0..c_{m-1}) = (k_{m+1}..k_{2m}) \star_2 Z^{-1}$

• Anwendungsbeispiel für $m = 3$

- Eve hat Schlüsseltext **1110111111** und Klartext **1011001101**
- Berechneter Schlüsselstrom ist **010 1110010**
- Berechne Inverse von $Z := \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ als $Z^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$
- Es folgt $(c_0, c_1, c_2) = (1, 1, 1) \star_2 Z^{-1} = (1, 1, 0)$

EINFACHE KRYPTOSYSTEME IM RÜCKBLICK

- **Buchstabenorientierte Systeme**

- Substitution von Buchstaben durch andere Symbole des Alphabets
- Mono- und polyalphabetische Variante
- Anfällig für Brute-Force Attacken oder **statistische Analysen**

- **Blockbasierte Verschlüsselung**

- Permutationen und affin-lineare Chiffren
- Anfällig für known plaintext Attacken mit **Matrix-Invertierung**

- **Strombasierte Verschlüsselung**

- Approximation von One-Time Pads durch lange Schlüsselströme macht statistische Analysen nahezu undurchführbar
- Schlüsselerzeugung mit und ohne Verwendung des Klartextes
- Zu brechen, wenn Erzeugungsverfahren für Schlüsselstrom bekannt

Keine Sicherheit im Computerzeitalter

- **Buchstabenorientierte Chiffrierung reicht nicht**

- Kryptosystem muß große Klartextblöcke auf einmal verschlüsseln
- Chiffrierung darf nicht affin-linear sein (auch nicht zufällig)
- Mehrere Klartextblöcke sollten nicht identisch verschlüsselt werden

- **Hohe Diffusion und Konfusion ist wichtig**

- Annähernde Gleichverteilung der Schlüssel und statistisch geringe Abhängigkeit zwischen Klar- und Schlüsseltext
- Perfekte Sicherheit bleibt unerreichbar, da One-Time Pads zu teuer

- **Systeme müssen sehr komplex werden**

- Hohes Maß an Sicherheit gegenüber jeder möglichen Attacke
 - Aufwendige Verschlüsselungsalgorithmen mit großen Schlüsseln
 - Schlüssel dürfen nur mit Hilfe von Zufallsgeneratoren bestimmt werden
- Ver-/Entschlüsselung nur noch mit Computerunterstützung möglich
 - Große Datenmengen müssen effizient verarbeitet werden können