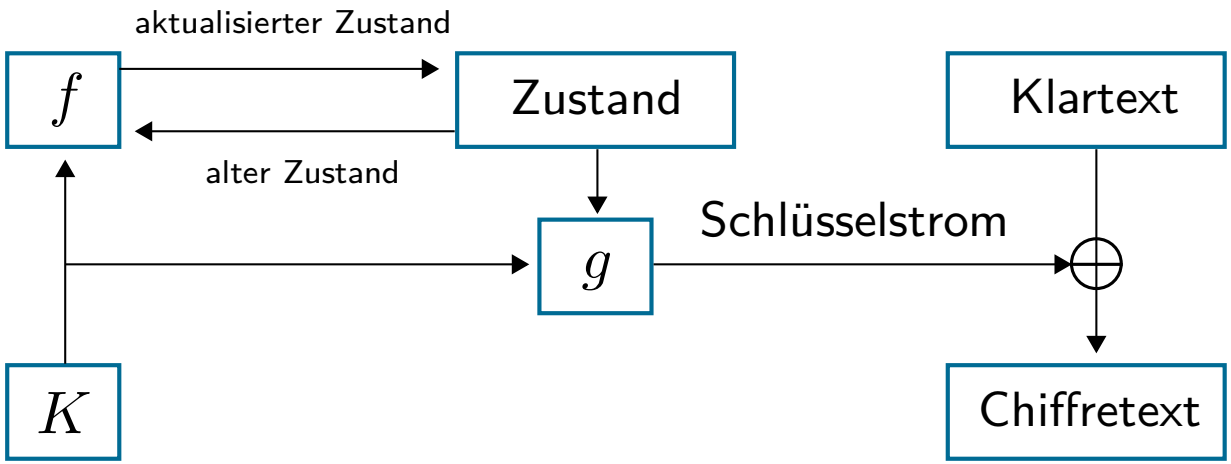
2: Stromchiffren
2.1: Synchrone Stromchiffren



- Die Vernam-Chiffre (das OTP) ist theoretisch perfekt, aber kaum praktikabel wegen des langen Schlüssels.
- Deshalb erzeuge einen beliebig langen **pseudozufälligen Schlüsselstrom** aus einem **kurzen zufälligen Schlüssel**.

Synchrone Stromchiffre (2)

- Synchronisation zwischen Sender und Empfänger muss gewährleistet sein.
  ($\rightarrow$ ggf. zusätzl. Maßnahmen)
- Die Änderung eines Chiffretext-Bits entspricht der Änderung eines Klartext-Bits, genau wie bei der Vernam-Chiffre.
  ($\rightarrow$ kein Schutz der Authentitzität)
- Es gibt auch sogenannte *selbstsynchronisierende* Stromchiffren.
  (kein Thema in dieser Vorlesung)

Katastrophaler Fehler bei Synchronen Stromchiffren:

- Mehrfache Verwendung eines Startzustandes.
- Genauso schlimm wie bei der Vernam-Chiffre/OTP.
- Diesen Fehler trifft man in der Praxis erstaunlich oft an!

Beispiel:

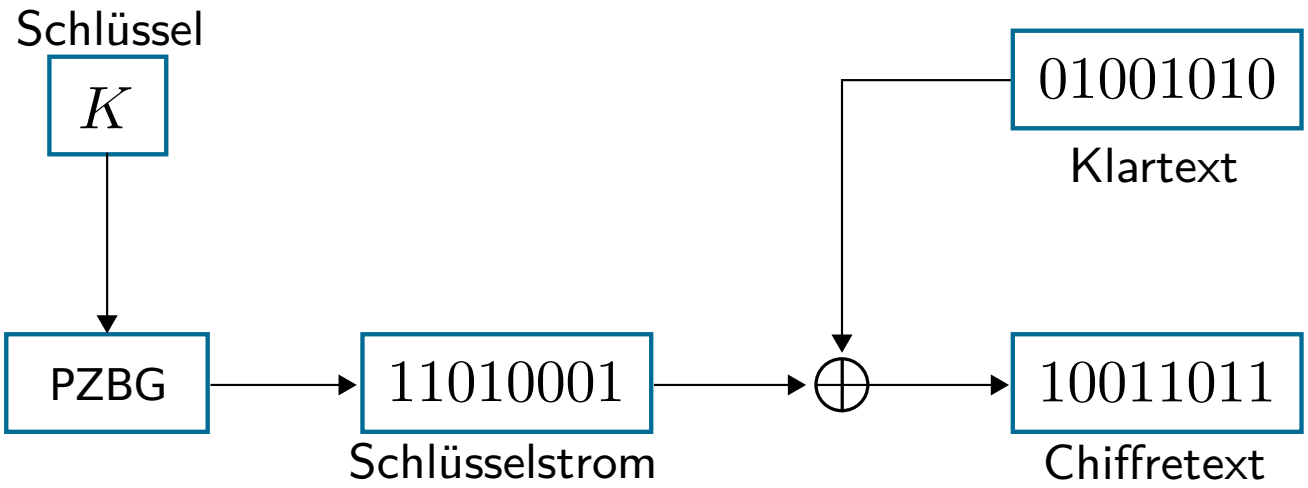
`http://www.heise.de/security/artikel/Verschusselt-statt-verschluesselt-270058.html`

Allgemeine Beschreibung des Phänomens:

`http://www.heise.de/security/artikel/XOR-Verschluesselung-270952.html`

Abstrakte PZBGs

PZBG = PseudoZufallsBitGenerator



Ein PZBG ist **kryptographisch sicher** wenn man den Schlüsselstrom ohne Kenntnis des Schlüssels nicht von einem zufälligen Bit-Strom *("Würfe mit einer fairen Münze")* unterscheiden kann.

Wir werden noch beweisen, dass eine derartige Stromchiffre sicher ist, wenn der PZBG kryptographisch sicher ist.

2.2: Theorie: Die Sicherheit eines PZBGs als Stromchiffre

Satz 5

PZBG kryptographisch sicher $\Rightarrow$ *binäre additive Stromchiffre sicher.*

Beweisidee:

Wenn die Schlüsselstrom-Bits "echt zufällig" sind, ist die Chiffre sicher ($\rightarrow$ Vernam-Chiffre).

Kann man die Chiffre "knacken", dann hat man auch ein Kriterium, den Schlüsselstrom von einem Strom "echt zufälliger" Bits zu unterscheiden. □

Was heißt hier “sicher”?

Ein Kryptosystem gilt als *sicher* gegen eine bestimmte Klasse von Angriffen, wenn es keine *effizienten* Algorithmen gibt, die bei einem derartigen Angriff mit *signifikanter* Wahrscheinlichkeit erfolgreich sind.

Die Begriffe **effizient** und **signifikante Wahrscheinlichkeit** lassen sich grundsätzlich mit konkreten Vorstellungen identifizieren (“10^{25} *MIPS-Jahre*”, “*Wahrscheinlichkeit kleiner als* 10^{-25} ”).

Die Begriffe haben aber auch eine *streng formale Definition* in der Komplexitätstheorie.

Effizient und Signifikant

- **Abstrakt** berachten wir nicht einzelne Kryptosysteme, sondern Folgen von Kryptosystemen, die indiziert werden durch ihre Schlüsselgröße k (und ggf. weitere Sicherheitsparameter).
- Entsprechend sind Laufzeit und Erfolgswahrscheinlichkeit eines Angreifers Funktionen in k.

Effizient Ein Algorithmus $f_k(\cdot)$ ist **effizient**, wenn seine Laufzeit $L_f(k)$ **polynomiell beschränkt** ist, d.h., wenn ein Polynom $p(k)$ existiert, so dass für fast alle k gilt:

$$L_f(k) \leq p(k).$$

Signifikant: Eine Folge von Wahrscheinlichkeiten $\Pr(k)$ ist **signifikant**, wenn $\frac{1}{P(k)}$ polynomiell beschränkt ist. Die Differenz von Wahrscheinlichkeiten $\Pr_1$ und $\Pr_2$ ("Vorteil") ist **signifikant**, wenn $\frac{1}{|P_1(k)-P_2(k)|}$ polynomiell beschränkt ist.

Kern der Komplexitätstheorischen Sicherheit

- Angreifer:
  - **Probabilistischer polynomieller Algorithmus**
  - Polynomiell = effizient, beschränkte Ressourcen!
  - Probabilistisch = Muss nicht terminieren, nicht immer korrekt sein
  - Vgl: Angreifer in Informationstheorie hat *unbegrenzte* Ressourcen
- Sicherheit:
  - Algorithmus $f_k(\cdot)$ mit Sicherheitsparameter k ist **exponentiell sicher**
    $\Longrightarrow$ es existiert kein effizienter Angreifer auf f_k, der k nicht kennt

PZBG (Definition)

Ein **Pseudozufallsbitgenerator** (PZBG) ist eine Familie von effizient berechenbaren Funktionen

$$f_k : \{0,1\}^k \to \{0,1\}^{\lambda(k)}$$

mit $\lambda(k) > k$.

Intention: Nimm einen "kurzen" k-bit Schlüssel als Input für f, um einen "langen" $\lambda(k)$-bit Schlüsselstrom zu erzeugen. In der Regel ist $\lambda(k) \gg k$.

PZBG (Angreifer)

Ein **Angreifer auf einen PZBG** ist ein effizienter Algorithmus, der einen $\lambda(k)$-bit Schlüsselstrom als Eingabe hat und ein Bit ausgibt.

Sei $x_0 \in \{0,1\}^{\lambda(k)}$ ein mit dem PZBG unter einem zufälligen Schlüssel erzeugter Schlüsselstrom, $x_1 \in \{0,1\}^{\lambda(k)}$ sei das Ergebnis von $\lambda(k)$ unabhängigen Würfen mit einer fairen Münze.

Der **Vorteil** eines Angreifers (auf einen PZBG) ist

$$\Big|\Pr\left[\text{A gibt “0” aus}|x_0\right] - \Pr\left[\text{A gibt “0” aus}|x_1\right]\Big|.$$

PZBG (Definition der Sicherheit)

Ein PZBG ist **sicher**, wenn es keinen effizienten Angreifer gibt, der einen signifikanten Vorteil erreicht.

Intention: Der Vorteil gibt an, ob man zwischen einem pseudozufälligen und einem zufälligen Schlüsselstrom unterscheiden kann. Bei einem sicheren PZBG soll dies eben praktisch unmöglich sein.

"Allgemeine" PZBGs aus "einfacheren" PZGBs

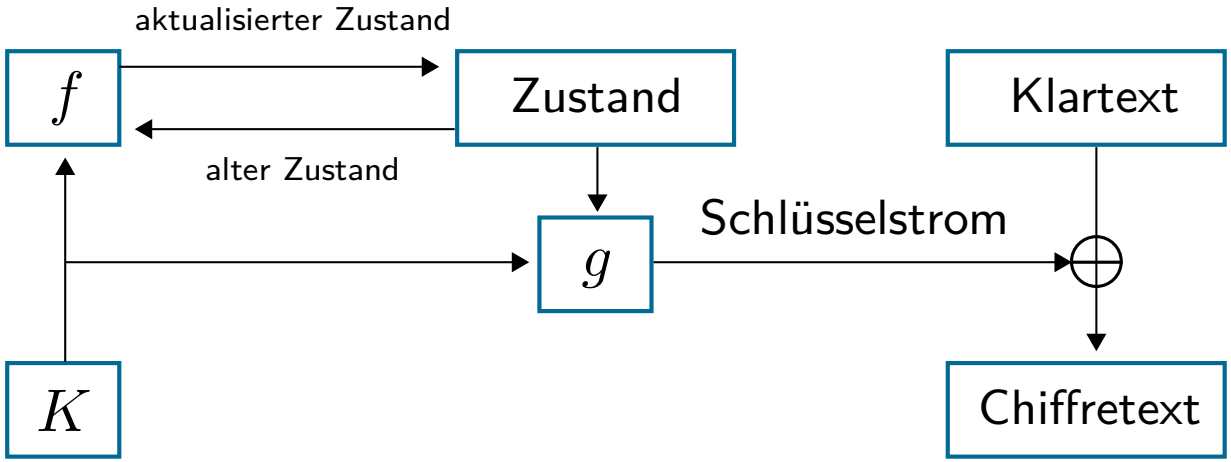
Familie $\{F_k\}_{k\in\mathbb{N}}$ von Funktionen $F_k : \{0,1\}^k \to \{0,1\}^{k+1}$;
Aufspalten der $k+1$-bit Ausgabe in einen k-bit und einen 1-bit Anteil:

$$F_k(x_1,\ldots,x_k) = (\overbrace{f_k(x_1,\ldots,x_k)}^{k \text{ bit}}, \overbrace{g_k(x_1,\ldots,x_k)}^{1 \text{ bit}}).$$

Für $\lambda \geq 1$: neue Familie $\{F_k^\lambda\}_{k\in\mathbb{N}}$ von Funktionen $F_k^\lambda : \{0,1\}^k \to \{0,1\}^\lambda$:

$$\begin{aligned} F_k^1(x_1,\ldots,x_k) &= \overbrace{g_k(x_1,\ldots,x_k)}^{1 \text{ bit}} \\ F_k^\lambda(x_1,\ldots,x_k) &= (\overbrace{f_k^{\lambda-1}(f_k(x_1,\ldots,x_k))}^{\lambda-1 \text{ bit}}, \overbrace{g_k(x_1,\ldots x_k)}^{1 \text{ bit}}). \end{aligned}$$

Zur Erinnerung: Abstrakte PZBGs



$$F_k^1(x_1,\ldots,x_k) = \overbrace{g_k(x_1,\ldots,x_k)}^{1 \text{ bit}}$$

$$F_k^\lambda(x_1,\ldots,x_k) = (\overbrace{f_k^{\lambda-1}(f_k(x_1,\ldots,x_k))}^{\lambda-1 \text{ bit}}, \overbrace{g_k(x_1,\ldots x_k)}^{1 \text{ bit}}).$$

Die Sicherheit dieser “allgemeinen” PZBGs

Satz 6 (Yao, 1982)

Sei $\lambda = \lambda(k) \geq 0$ durch ein Polynom in k beschränkt. Dann gilt:

a) *Wenn $\{F_k\}_{k\in\mathbb{N}}$ effizient berechenbar ist, dann ist auch $\{F_k^\lambda\}_{k\in\mathbb{N}}$ effizient berechenbar.*

b) *Wenn $\{F_k\}_{k\in\mathbb{N}}$ sicher ist, dann ist auch $\{F_k^\lambda\}_{k\in\mathbb{N}}$ sicher.*

2.3: Schlussbemerkungen

Sie sollten

- den Zusammenhang zwischen einem PZBG und einer synchronen Stromchiffren verstanden haben
- und wissen, warum es *theoretisch* genügt, einen sicheren und effizienten PZBG zu haben, der aus k zufälligen Schlüsselbits $k + 1$ pseudozufällige Ausgabebits generiert.