Kryptografie verständlich

Ein Fachbuch für
Studierende und Anwender

von
Christof Paar und Jan Pelzl

www.crypto-textbook.com

Kapitel 2

Stromchiffren

(Version: 1. Dezember 2016)

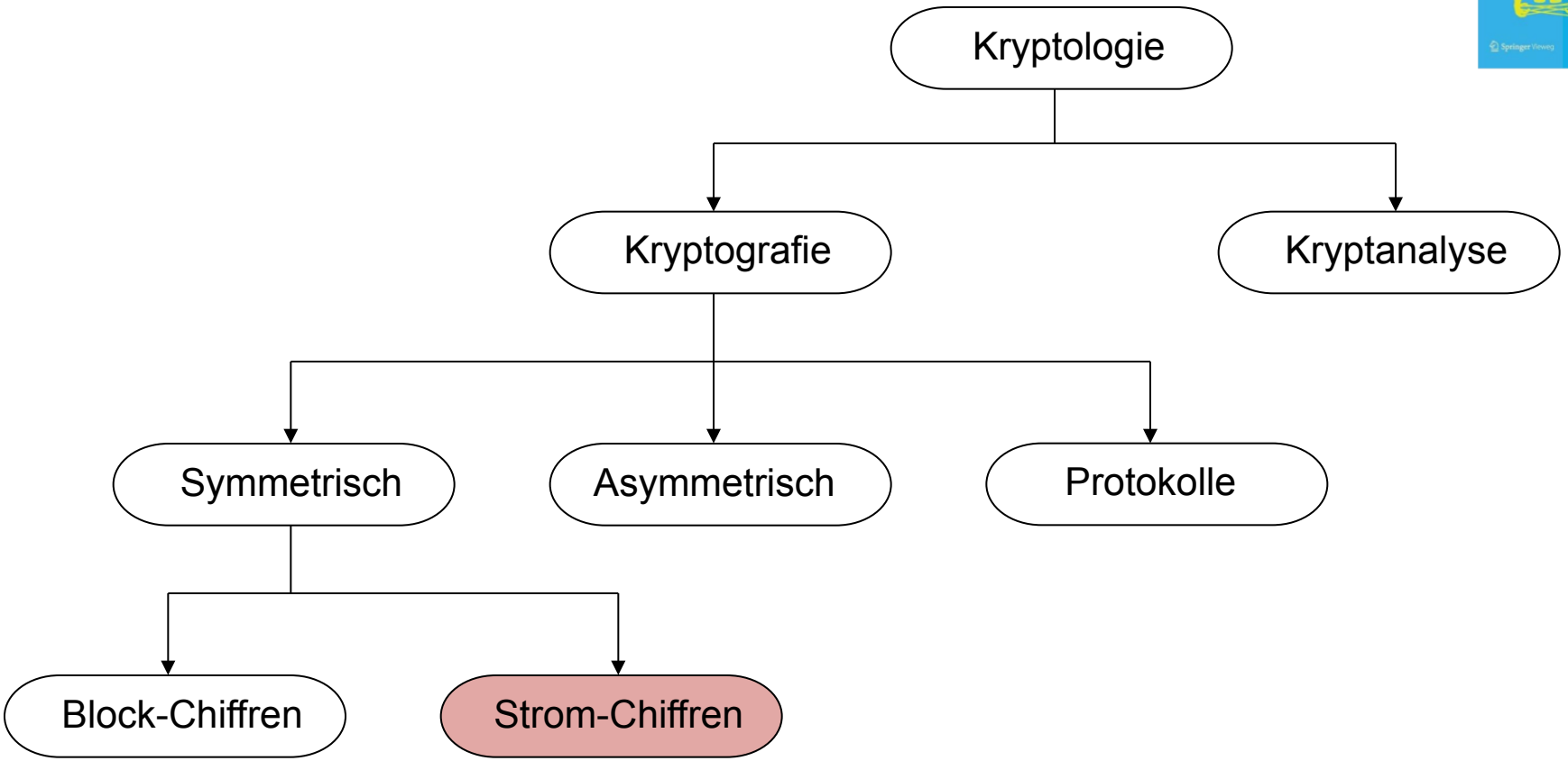
Übersicht

- Grundlagen von Stromchiffren
- Zufallszahlengeneratoren (RNGs)
- Der One-Time Pad (OTP)
- Linear Feedback Shift Register (LFSR)
- Trivium: Eine moderne Stromchiffre

Übersicht

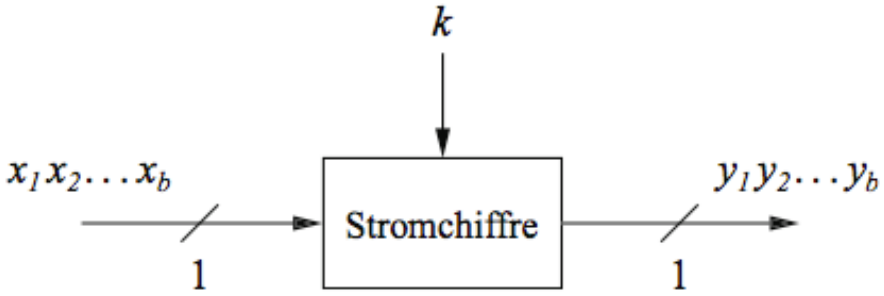
- **Grundlagen von Stromchiffren**
- Zufallszahlengeneratoren (RNGs)
- Der One-Time Pad (OTP)
- Linear Feedback Shift Register (LFSR)
- Trivium: Eine moderne Stromchiffre

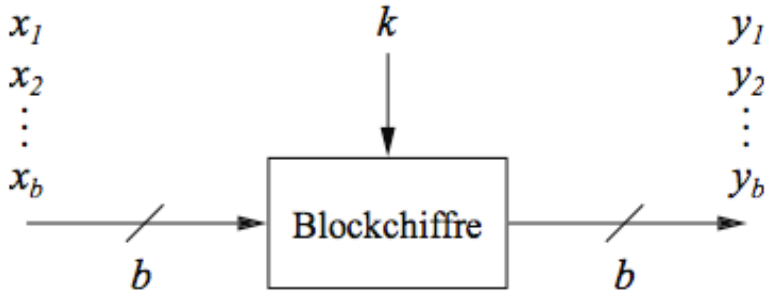
Stromchiffren in der Kryptologie



Stromchiffren wurden 1917 von Gilbert Vernam erfunden

Stromchiffren vs. Blockchiffren



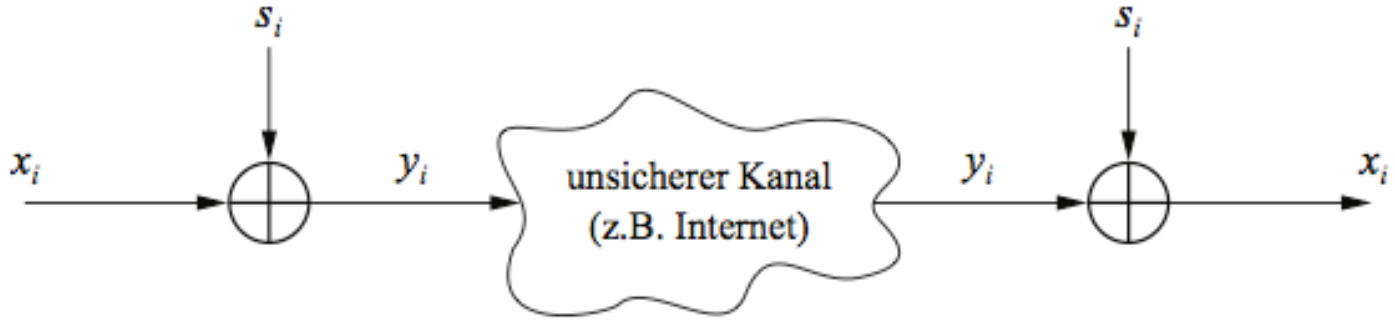


- **Stromchiffre**
  - Verschlüsselt bitweise
  - I.d.R. klein und schnell → Verwendung in eingeschränkten Umgebungen (z.B. A5/1 im GSM Standard)

- **Blockchiffre:**
  - Verschlüsselt immer einen ganzen Block (zahlreiche Bit)
  - Verwendung in klassischer ICT (z.B. Internet)

Ver- und Entschlüsselung mit Stromchiffren

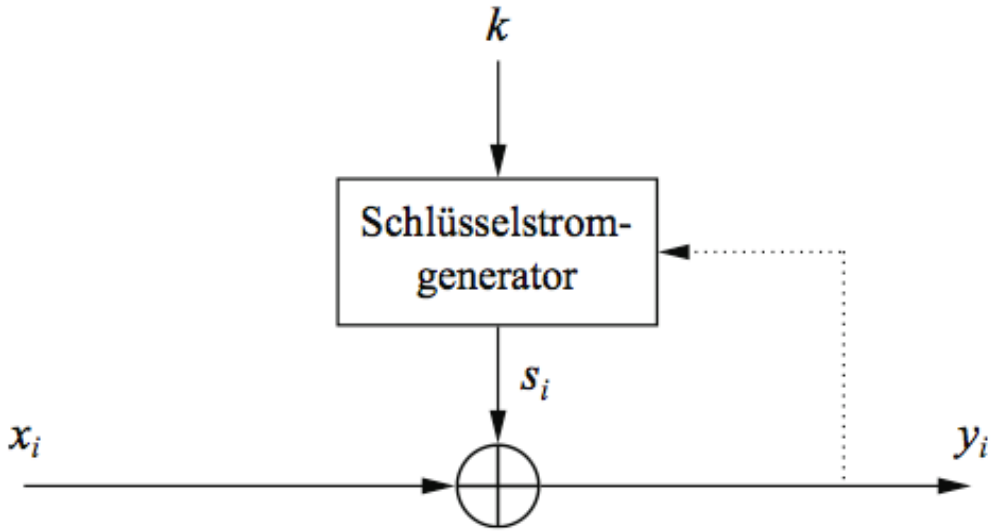
Klartext x_i, Chiffrat y_i und Schlüsselstrom s_i
besteht aus einzelnen Bit



- Ver- und Entschlüsselung sind einfache Additionen modulo 2 (aka XOR)
- Ver- und Entschlüsselung sind die gleichen Funktionen

- **Verschlüsselung:** $y_i = e_{si}(x_i) = x_i + s_i \bmod 2 \quad x_i, y_i, s_i \in \{0,1\}$
- **Entschlüsselung:** $x_i = e_{si}(y_i) = y_i + s_i \bmod 2$

Synchrone und Asynchrone Stromchiffren



- Sicherheit abhängig vom Schlüsselstrom s_i:
  - sollte zufällig sein, d.h., $Pr(s_i = 0) = Pr(s_i = 1) = 0.5$
  - muss von Sender und Empfänger reproduzierbar sein
- Synchrone Stromchiffre: Schlüsselstrom hängt nur vom Schlüssel ab (und ggf. von einem IV)
- Asynchrone Stromchiffre: Schlüsselstrom hängt auch von dem Chiffrat ab (sh. gestrichelte Rückkopplung)

Warum die Modulo 2 Addition eine gute Funktion zur Verschlüsselung ist

- Modulo 2 Addition ist äquivalent zu einer XOR Operation

- Für einen perfekten Schlüsselstrom s_i , hat jedes Chiffrat eine 50%ige Chance, 0 oder 1 zu sein

  → Gute statistische Eigenschaft für das Chiffrat

- Die Umkehrung eines XOR ist wiederum ein XOR

x_i	s_i	y_i
0	0	0
0	1	1
1	0	1
1	1	0

Durchsatz von Stromchiffren

Einige Zahlen zum Durchsatz von symmetrischen Verfahren (Pentium4):

Chiffre	Schlüssellänge	Mbit/s
DES	56	36.95
3DES	112	13.32
AES	128	51.19
RC4 (Stromchiffre)	(wählbar)	211.34

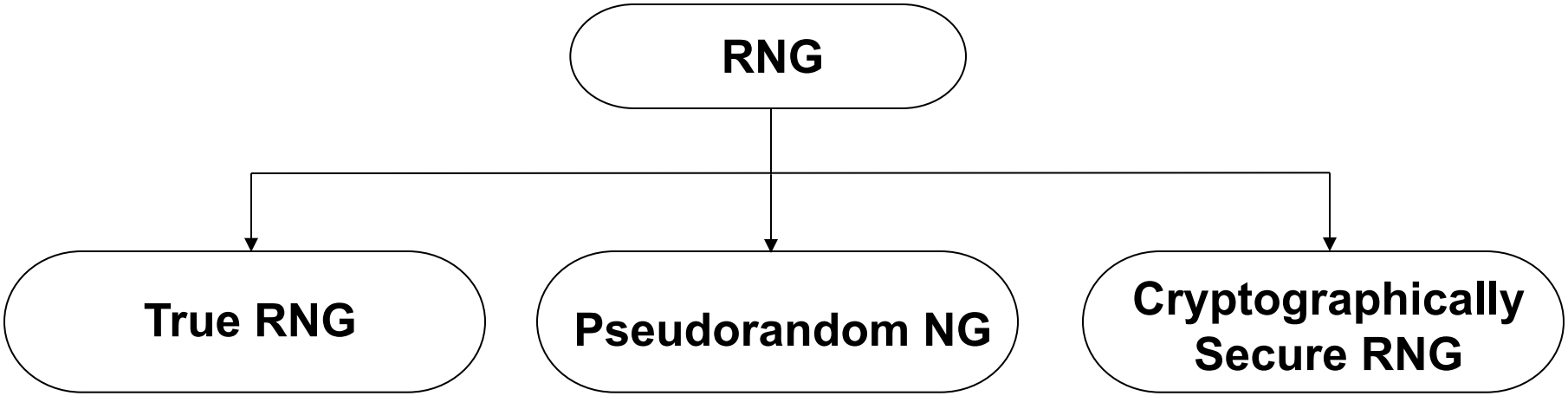
Quelle: Zhao et al., Anatomy and Performance of SSL Processing, ISPASS 2005

Übersicht

- Grundlagen von Stromchiffren
- **Zufallszahlengeneratoren (RNGs)**
- Der One-Time Pad (OTP)
- Linear Feedback Shift Register (LFSR)
- Trivium: Eine moderne Stromchiffre

Zufallszahlengeneratoren (RNGs)

Übersicht

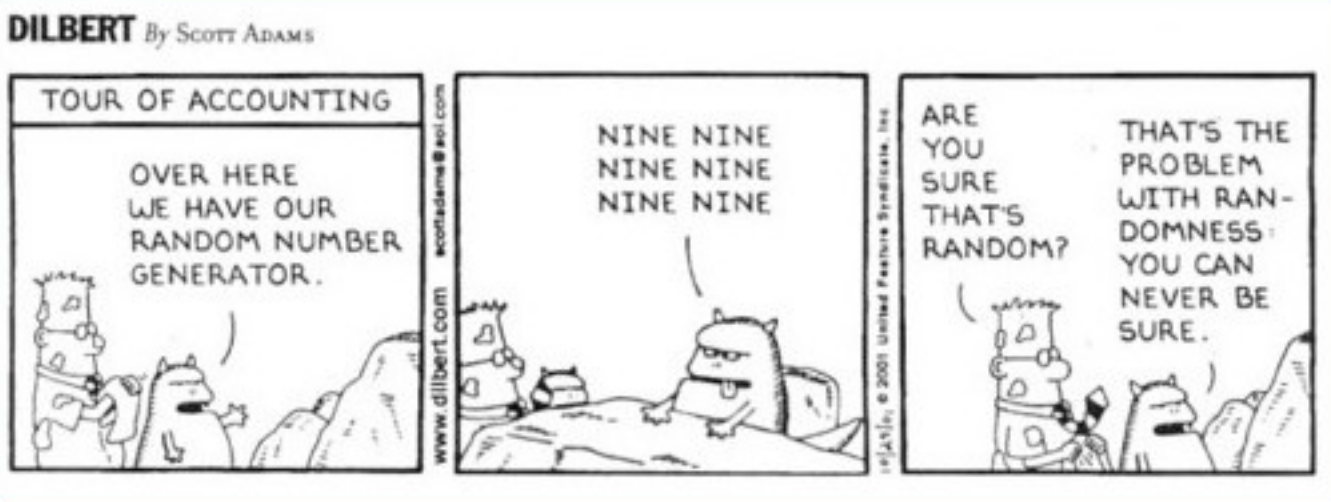


Zufallszahlengeneratoren

Echte Zufallszahlengeneratoren (TRNGs)

- Basieren auf physikalischen Zufallsprozessen: Münzwurf, Würfeln, Halbleiterrauschen, radioaktiver Zerfall, Mausbewegung, Clock Jitter digitaler Schaltungen
- Ausgabestrom s_i sollte gute statistische Eigenschaften haben: $Pr(s_i = 0) = Pr(s_i = 1) = 50\%$ (oft durch Nachverarbeitung erreicht)
- Ausgabe ist weder vorhersehbar noch reproduzierbar

Verwendung: Schlüsselerzeugung, Nonces (Numbers used only-once) und einige weitere Anwendungen



Zufallszahlengeneratoren

Pseudozufallszahlengeneratoren (PRNG)

- Erzeugen pseudozufällige Sequenz aus Seed
- Üblicherweise Ausgabe mit guten statistischen Eigenschaften
- Ausgabe kann reproduziert werden und ist vorhersehbar
- Berechnung oft rekursiv: $s_0 = seed$

$$s_{i+1} = f(s_i, s_{i-1}, \ldots, s_{i-t})$$

- Beispiel: *rand()* Funktion in ANSI C:

$$s_0 = 12345$$

$$s_{i+1} = 1103515245 s_i + 12345 \bmod 2^{31}$$

- Die meisten PRNGs haben **schlechte kryptografische Eigenschaften**

Zufallszahlengeneratoren
Kryptanalyse

Beispiel eines einfachen PRNG: **Linearer Kongruenz-Generator**

$$S_0 = seed$$
$$S_{i+1} = AS_i + B \bmod m$$

Annahme

- A, B und S_0 als Schlüssel sind unbekannt
- Größe von A, B und S_i sind 100 Bit
- 300 Bit der Ausgabe sind bekannt, d.h. S_1, S_2 und S_3

Lösung von

$$S_2 = AS_1 + B \bmod m$$
$$S_3 = AS_2 + B \bmod m$$

ergibt A und B. Alle S_i können nun leicht berechnet werden!

Die Linearität in den meisten PRNGs führt zu schlechten kryptografischen Eigenschaften

Zufallszahlengeneratoren
Cryptographically Secure Pseudorandom Number Generator (CSPRNG)

- Spezieller PRNG mit zusätzlicher Eigenschaft:
  - **Die Ausgabe muss unvorhersehbar sein**

  Genauer: Bei gegebenen n aufeinanderfolgenden Ausgabebits s_i sind die nachfolgenden Bit s_{n+1} nicht vorhersagbar (in polynomieller Zeit).

- Verwendung in der Kryptografie, insbesondere bei Stromchiffren

- Anmerkung: Fast keine Anwendung benötigt Nichtvorhersagbarkeit, wobei viele (technische) Systeme PRNGs benötigen

Übersicht

- Grundlagen von Stromchiffren
- Zufallszahlengeneratoren (RNGs)
- **Der One-Time Pad (OTP)**
- Linear Feedback Shift Register (LFSR)
- Trivium: Eine moderne Stromchiffre

Der One-Time Pad (OTP)

Uneingeschränkte Sicherheit:

- Ein Kryptosystem ist uneingeschränkt sicher, wenn es selbst mit *unendlich* viel Rechenleistung nicht gebrochen werden kann

One-Time Pad

- Von Mauborgne entwickeltes Kryptosystem, basierend auf Vernams Stromchiffre:
- Eigenschaften:

  Seien Klartext, Chiffrat und Schlüssel gegeben als einzelne Bit $x_i, y_i, k_i \in \{0,1\}$.

Verschlüsselung: $e_{k_i}(x_i) = x_i \oplus k_i$.
Entschlüsselung: $d_{k_i}(y_i) = y_i \oplus k_i$

Der OTP ist nur dann uneingeschränkt sicher, wenn der Schlüssel k_i nur einziges mal verwendet wird!

Der One-Time Pad (OTP)

Uneingeschränkte Sicherheit

Jede Gleichung ist eine lineare Gleichung mit zwei Unbekannten

$$y_0 = x_0 \oplus k_0$$
$$y_1 = x_1 \oplus k_1$$
$$:$$

für jedes y_i sind $x_i = 0$ und $x_i = 1$ gleich wahrscheinlich!

⇒ Notwendige Bedingung: $k_0, k_1, \ldots$ sind voneinander unabhängig (d.h. echt zufällig)

⇒ Dann kann gezeigt werden, dass das System *beweisbar* sicher ist

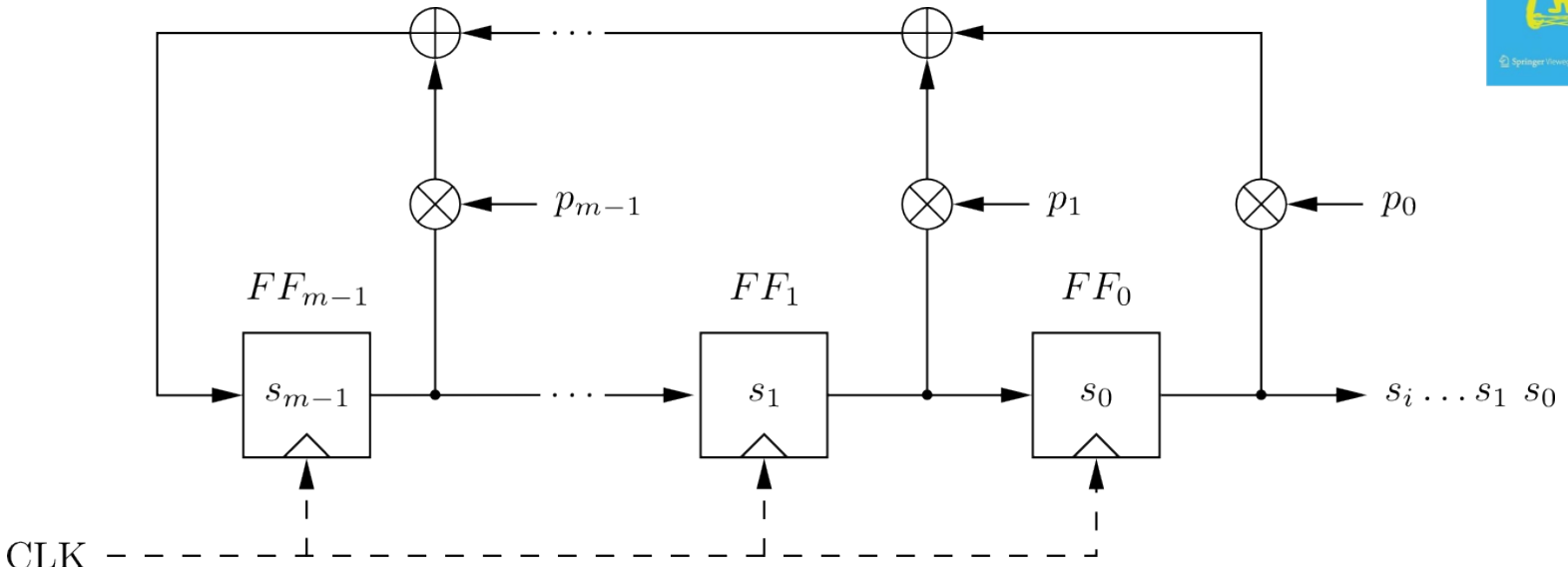
Nachteil: Der OTP ist **unpraktisch**, da der Schlüssel genauso lang wie die Nachricht sein muss!

Übersicht

- Grundlagen von Stromchiffren
- Zufallszahlengeneratoren (RNGs)
- Der One-Time Pad (OTP)
- **Linear Feedback Shift Register (LFSR)**
- Trivium: Eine moderne Stromchiffre

Linear Feedback Shift Register (LFSR)

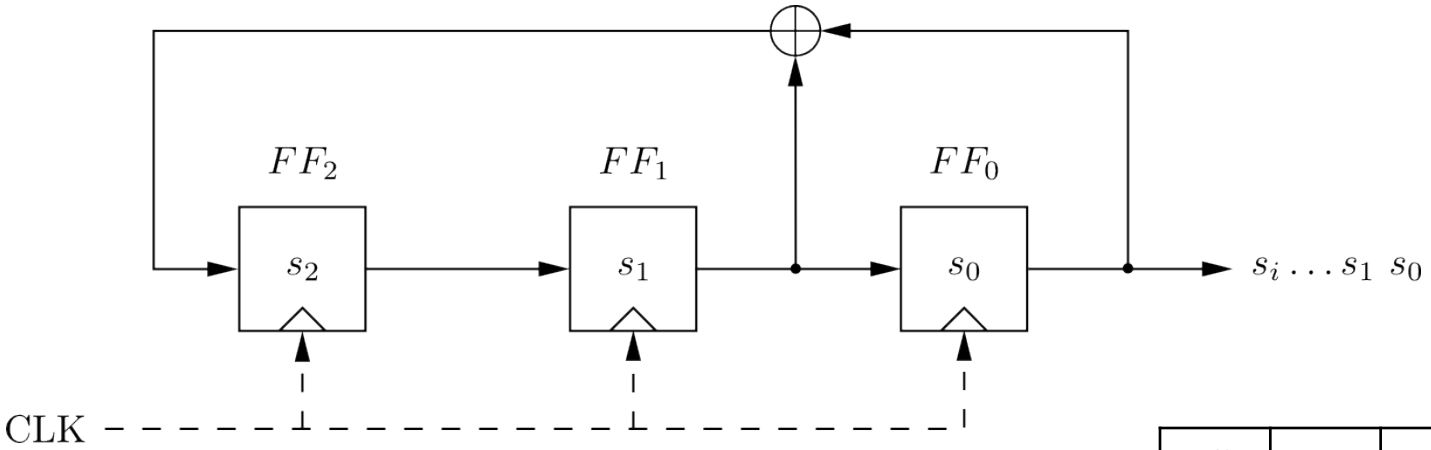
Grundlagen



- Verkettete *Flip-Flops (FF)* mit Rückkopplungspfad
- Rückkopplung berechnet neue Eingabe durch XOR-Verknüpfung bestimmter Status-Bits
- *Grad m* gegeben durch die Anzahl der Speicher-Elemente
- Wenn $p_i = 1$, ist eine Rückkopplung aktiv (“geschlossener Schalter”), ansonsten keine Rückkopplung durch das Flip-Flop (“offener Schalter”)
- Ausgabesequenz wiederholt sich periodisch
- Maximale Sequenzlänge: 2^m-1

Linear Feedback Shift Register (LFSR)

Beispiel mit m=3



- Ausgabe des LFSRs durch folgende rekursive Gleichung darstellbar:

$$s_{i+3} = s_{i+1} + s_i \bmod 2$$

- Maximale Sequenzlänge (von 2^3-1=7) nur durch bestimmte Rückkopplungskoeffizienten möglich, z.B.

clk	FF_2	FF_1	$FF_0=s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

Linear Feedback Shift Register (LFSR)

Sicherheit

Beschreibung von LFSRs durch Polynome:

$$P(x) = x^m + p_{l-1}x^{m-1} + \ldots + p_1 x + p_0$$

- Einfache LFSRs erzeugen vorhersagbare Ausgabe
- Wenn 2*m* Bit der Ausgabe eines LFSR vom Grad *m* bekannt sind, können die Rückkopplungskoeffizienten p_i des LFSRs durch Berechnung eines Linearen Gleichungssystems berechnet werden*

- Aus diesem Grund verwenden viele Stromchiffren **Kombinationen** von LFSRs

*Siehe Kapitel 2 von *Understanding Cryptography* für weitere Details.

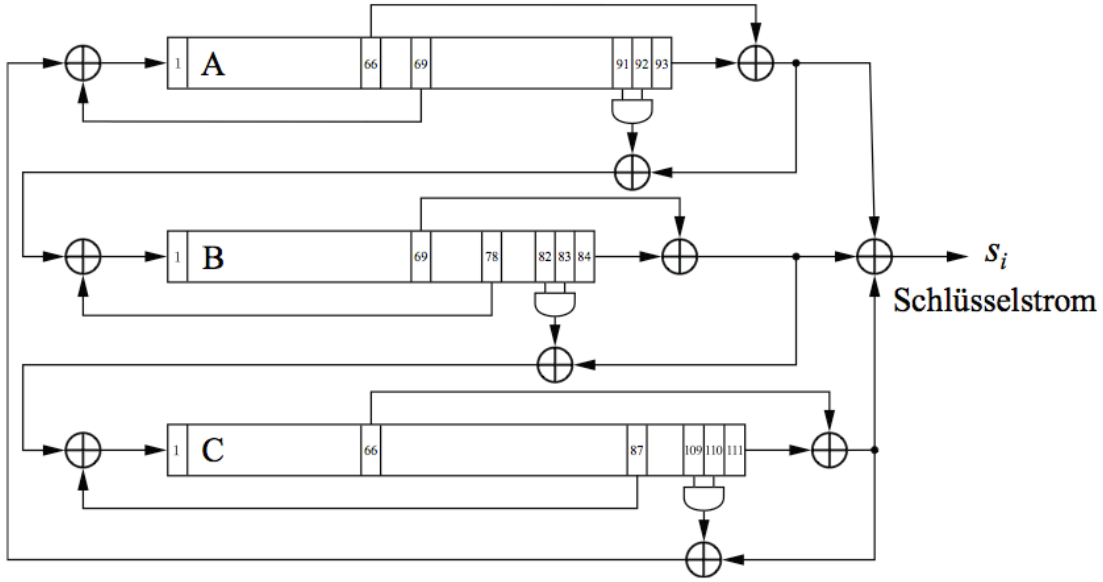
Übersicht

- Grundlagen von Stromchiffren
- Zufallszahlengeneratoren (RNGs)
- Der One-Time Pad (OTP)
- Linear Feedback Shift Register (LFSR)
- **Trivium: Eine moderne Stromchiffre**

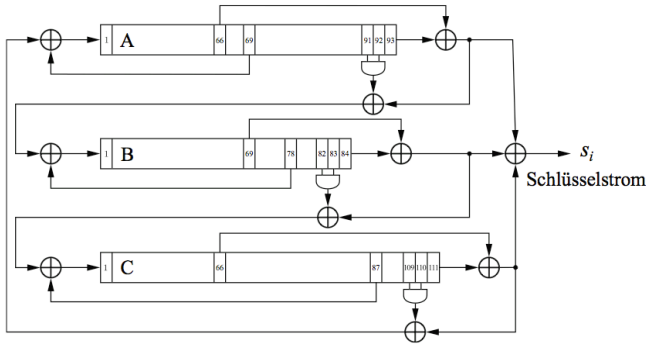
Trivium

Eine moderne Stromchiffre

- Drei *nichtlineare* LFSR (NLFSR) der Länge 93, 84, 111
- XOR-Summe der Ausgaben der drei NLFSR erzeugen Schlüsselstrom s_i
- Klein in Hardware:
  - Anzahl Register insgesamt: 288
  - Nichtlinearität: 3 AND-Gatter
  - 7 XOR-Gatter (4 mit 3 Eingängen)



Trivium
Übersicht



Initialisierung:

- Lade 80-bit IV in Register A
- Lade 80-bit Schlüssel in Register B
- Setze c_{109} , c_{110} , c_{111} = 1, alle anderen Bit = 0

Warm-Up:

- Takte Chiffre 4 x 288 = 1152 mal, ohne Ausgabe zu erzeugen

Ver- / Entschlüsselung:

- XOR-Summe der drei NLFSR Ausgänge erzeugen Schlüsselstrom s_i

Design kann parallelisiert werden, um 64 Ausgabebit pro Takt zu erzeugen

	Registerlänge	Feedback Bit	Feedforward Bit	AND Input
A	93	69	66	91, 92
B	84	78	69	82, 83
C	111	87	66	109, 110

Lessons Learned

- In den meisten Anwendungen sind Stromchiffren weniger verbreitet als Blockchiffren, mit Ausnahme einiger weniger Chiffren wie beispielsweise RC4.

- Umsetzungen von Stromchiffren benötigen manchmal weniger Ressourcen (Codegröße, Chipfläche) als Blockchiffren und sind daher für eingeschränkte Umgebungen wie RFIDs interessant.

- Die Anforderungen für einen kryptografisch sicheren Zufallszahlengenerator sind wesentlich aufwendiger als solche an Pseudozufallszahlengeneratoren

- Der One-Time Pad ist eine beweisbar sichere symmetrische Chiffre. Der OTP ist jedoch für die meisten Anwendungen ungeeignet, da u.a. die Schlüssellänge so groß wie die Nachrichtenlänge sein muss.

- Einfache LFSR sind trotz ihrer guten statistischen Eigenschaften schlechte Stromchiffren. Nur durch eine sorgfältige Kombination zahlreicher LFSR kann eine starke Chiffre erzeugt werden.