Einführung in die Kryptographie am Beispiel des A5 Algorithmus

Christoph Kurek

Geschichte

Seit der Antike: Verbreiteter, aber unsystematischer Einsatz kryptographischer Methoden (z.B. durch Caesar).

Ende 19. Jhdt.: Systematisierung und Formalisierung der Kryptographie.

2. Weltkrieg: Polen, Briten und Amerikaner knacken „sehr“ starke deutsche Chiffrierer (u.a. *Enigma*). Erstmalig Einsatz von Rechenmaschinen zum Code-Knacken.

Neuere Geschichte

70er Jahre:

- Data Encryption Standard (DES)
- Public-Key Kryptographie (Diffie-Hellman)

90er Jahre:

- Massenhafte Verbreitung der Kryptographie (Geldautomaten, Internet, Mobilfunk, Pay-TV, Signaturgesetz, . . .)

Betätigungsfelder

- Klassische Kryptographie

  diente der Geheimhaltung von Nachrichten und wurde hauptsächlich von Militärs, Geheimdienstlern und Diplomaten genutzt.
- Moderne Kryptographie (etwa seit 1975)

  beschäftigt sich mit erheblich weitreichenderen Kommunikations- und Sicherheitsproblemen.

„Cryptography is about communication in the presence of adversaries.“

Die Kryptographie beschäftigt sich mit Kommunikationsproblemen in der Anwesenheit von Widersachern.

(Zit.: Ron Rivest)

Einsatzfelder der Kryptographie

- Geheimhaltung von Daten (klassische Kryptographie)

  „*Nur wir können diesen Text lesen.*“
- Authentizität und Integrität von Daten

  „*Du hast diesen Brief geschrieben, und niemand hat am Text etwas geändert.*“
- Authentizität von Kommunikations-partnern

  „*Ach, du bist es!*“
- Anonyme Kommunikation (elektronisches Geld, . . .).

Grundlagen

Eine Chiffre wird def. durch drei Mengen:

1. P : Klartexte (Nachrichten),
2. C: Chiffretexte (Kryptogramme),
3. K: Schlüssel

und zwei (bzw. drei) Algorithmen

1. E : K x P => C (Verschlüsseln),
2. D : K x C => P (Entschlüsseln),

(3. G... => K (Schlüssel erzeugen)).

Die Caesar Chiffre

Julius Caesar verschlüsselte seine Nachrichten, indem er Klartext-Buchstaben „a“, . . . , „z“ auf die folgende Weise auf Chiffretext-Buchstaben „A“, . . . , „Z“ abbildete:

a => D
b => E
...
w => Z
x => A
y => B
z => C

Beispiel: "caesar" FDHVDU

Die Caesar Chiffre

Ist das nun eine Chiffre?

Klartextmenge P = {„a“, . . . , „z“}.

Chiffretextmenge C = {„A“, . . . „Z“}.

(Großbuchstaben wg. Übersicht).

Ver- und Entschlüsselungsalgorithmus

E und D vorhanden ➠ klar!

Schlüsselmenge???

Die Caesar Chiffre

Man ordne den Buchstaben eine Zahl aus der Menge $Z_{26} = \{0, \ldots, 25\}$ zu:

a bzw. A $\Leftrightarrow$ 0

b bzw. B $\Leftrightarrow$ 1

c bzw. C $\Leftrightarrow$ 2

y bzw. Y $\Leftrightarrow$ 24

z bzw. Z $\Leftrightarrow$ 25

Mengen $P = C = K = 26$.

Verschlüsseln: $E(k, x) = x + k \bmod 26$.

Entschlüsseln: $D(k, y) = y - k \bmod 26$.

Schlüsselerzeugung G: Trivial.

Wir haben eine Chiffre!!!

Die Caesar Chiffre

Problem:

- **Nur 26 Schlüssel möglich**

Lösung:

- **Substitutionschiffre**

Die Substitutionschiffre

Bei einer Substitutionschiffre ordnet man jedem Klartext-Buchstaben eindeutig einen Chiffretext-Buchstaben zu, z.B.:

a => D

b => R

c => L

. . .

y => X

z => Z

Anzahl Schlüssel : 26**!**

Die Substitutionschiffre

Aber: Auch die Substitutionschiffre ändert die Sprachstatistik nicht. Im Deutschen ist

- etwa 1/6 aller Buchstaben ein „e“,
- etwa 1/3 aller Buchstaben einer der drei häufigsten Buchstaben „e“, „n“ oder „i“ und
- etwa 2/3 aller Buchstaben einer der acht häufigsten Buchstaben.

Die Substitutionschiffre

Kann man genug (z.B. die häufigsten acht) Buchstaben richtig zuordnen, ist der Text fast schon flüssig zu lesen.

=> Ist der Klartext deutsch, englisch, französisch, und wird buchstabenweise verschlüsselt, so ist eine Substitutionschiffre hochgradig verwundbar gegen Angriffe.

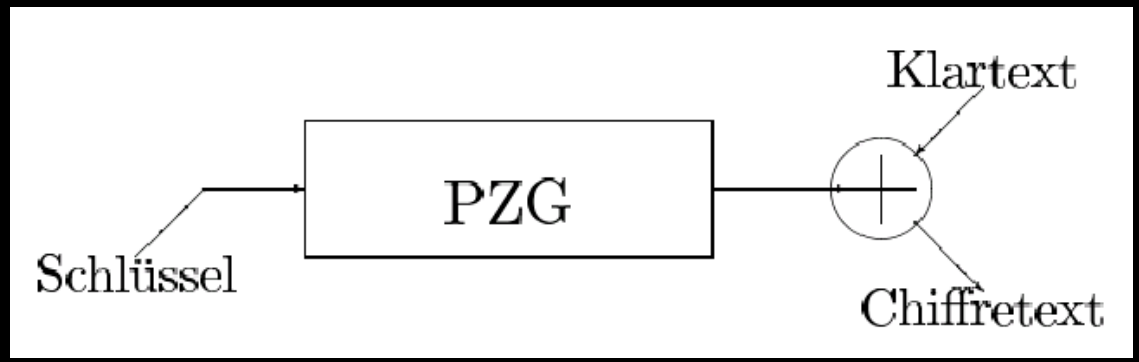
Flußchiffren

Definition: Sei A ein Alphabet mit q Elementen und E_e eine einfach Substitutionschiffre mit Blocklänge 1 mit $e \in K$. Sei $m_1m_2m_3...$ ein Klartext und $e_1e_2e_3...$ Schlüsselstrom(-fluß) aus K. Eine Flußchiffre nimmt den Klartext und produziert einen Chiffretext $c_1c_2c_3...$ $c_i = E_{e_i}(m_i)$. Falls d_i das Inverse zu e_i ist, dann „entschlüsselt“ $D_{d_i}(c_i) = m_i$ den Chiffretext.

PZG (Pseudozufallsgenerator)

Binäre additive Flußchiffre.

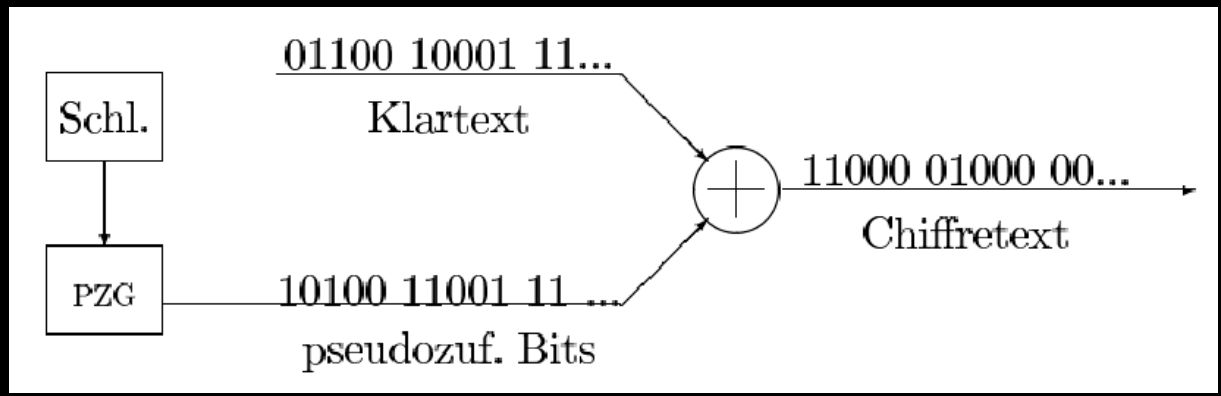
Pseudozufälliger Bitstrom, erzeugt mit Hilfe eines Pseudozufallsbitgenerators (PZG):



PZG (Pseudozufallsgenerator)

Der mit dem PZG erzeugte Schlüsselstrom wird zum Verschlüsseln bitweise zum Klartext addiert, zum Entschlüsseln bitweise vom Chiffretext subtrahiert. (Operation: „XOR“ für bitweises addieren bzw. subtrahieren.)

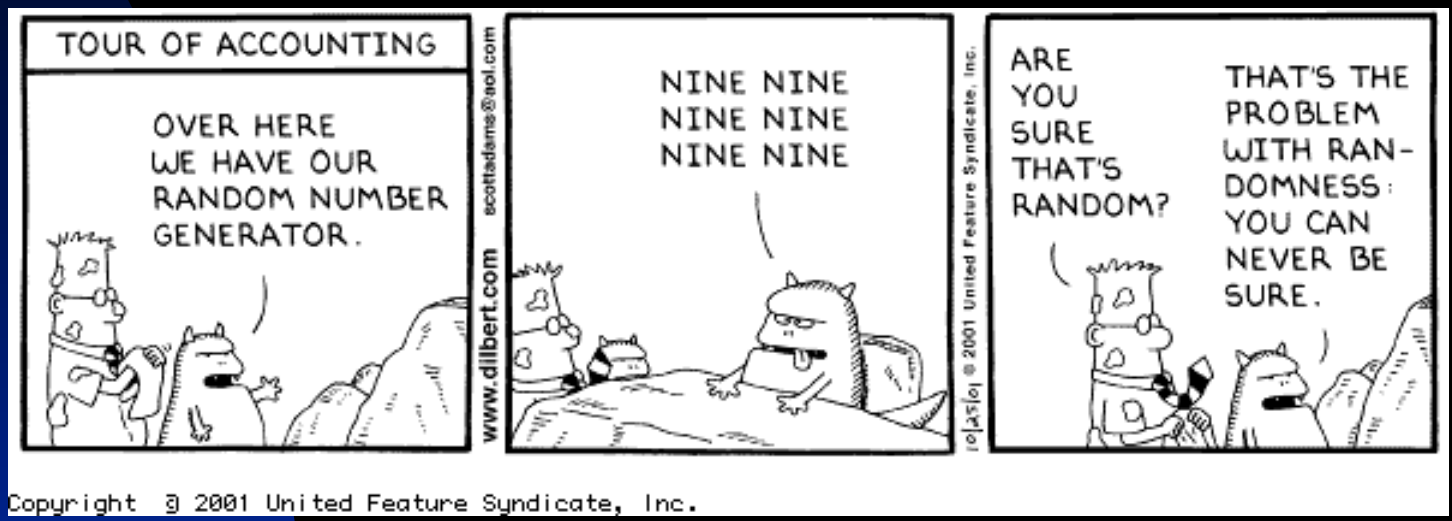
PZG (Pseudozufallsgenerator)



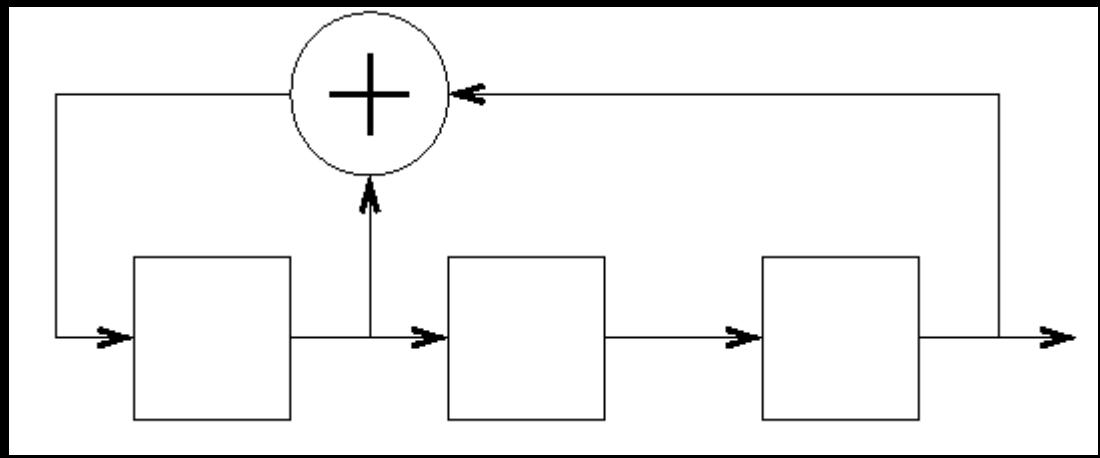
Ein PZG ist dann kryptographisch sicher, wenn man ihn ohne Kenntnis des Schlüssels nicht von einem zufälligen Bit-Strom („Werfen einer fairen Münze“) unterscheiden kann.

PZG kryptographisch sicher => Binäre additive Flußchiffre sicher.

PZG (Pseudozufallsgenerator)

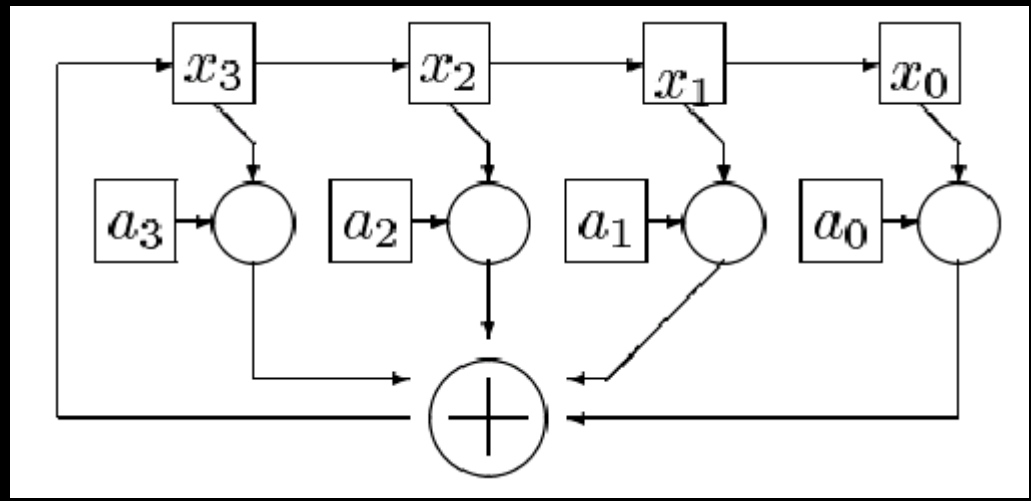


Schieberegister



Ist die Rückkopplungsfunktion linear, dann sprechen wir von einem „linearen, rückgekoppelten Schieberegister“ oder einem (engl.) „Linear Feedback Shift Register“ (LFSR).

Allgemeine LFSR



Funktion $f(x) = a_{n-1}\, x_{n-1} + \ldots + a_1\, x_1 + a_0\, x_0$

(„Feedback-Polynom“).

Allgemeine LFSR

Ein LFSR bildet einen sehr schlechten PZG!

Trotzdem: LFSR werden gerne als Bausteine für PZGs genutzt, in Verbindung mit nichtlinearen Bausteinen.

Einordnung des A5

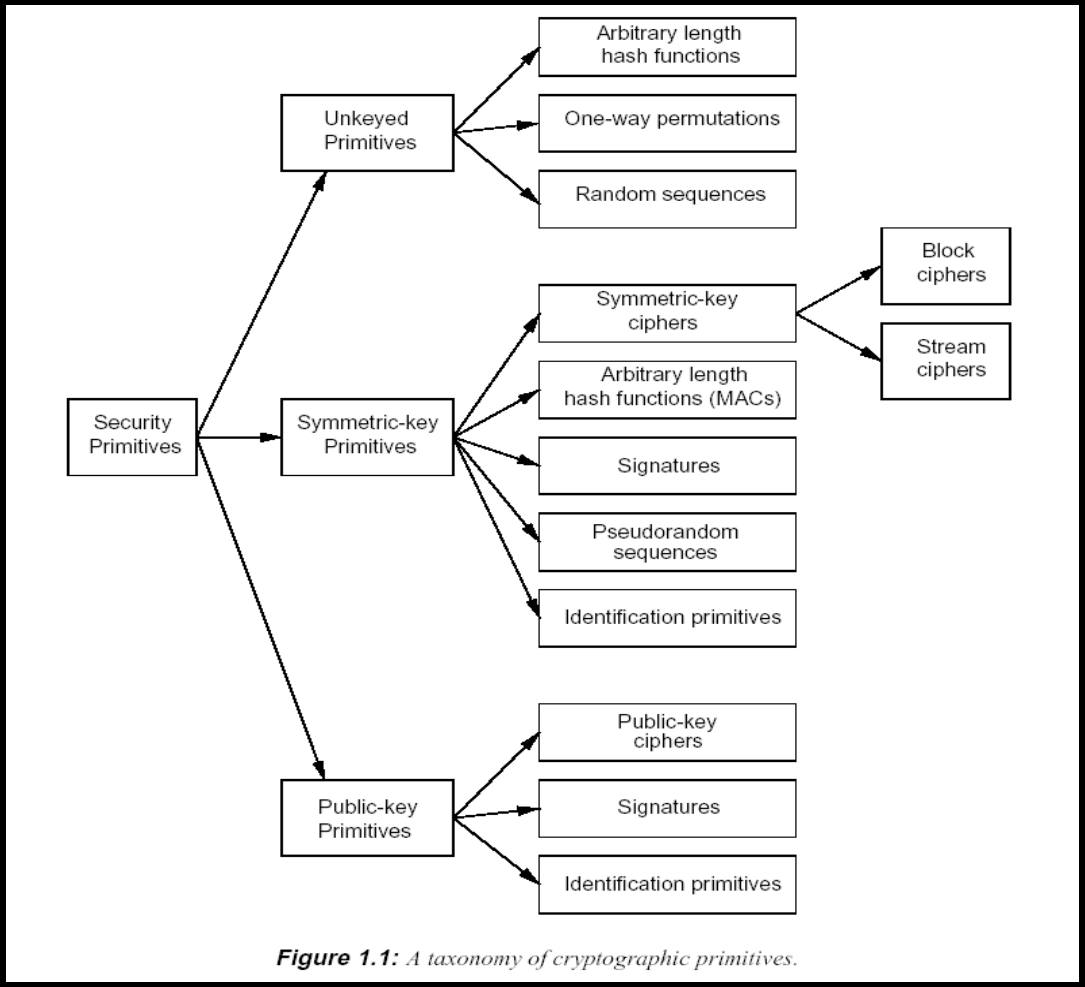
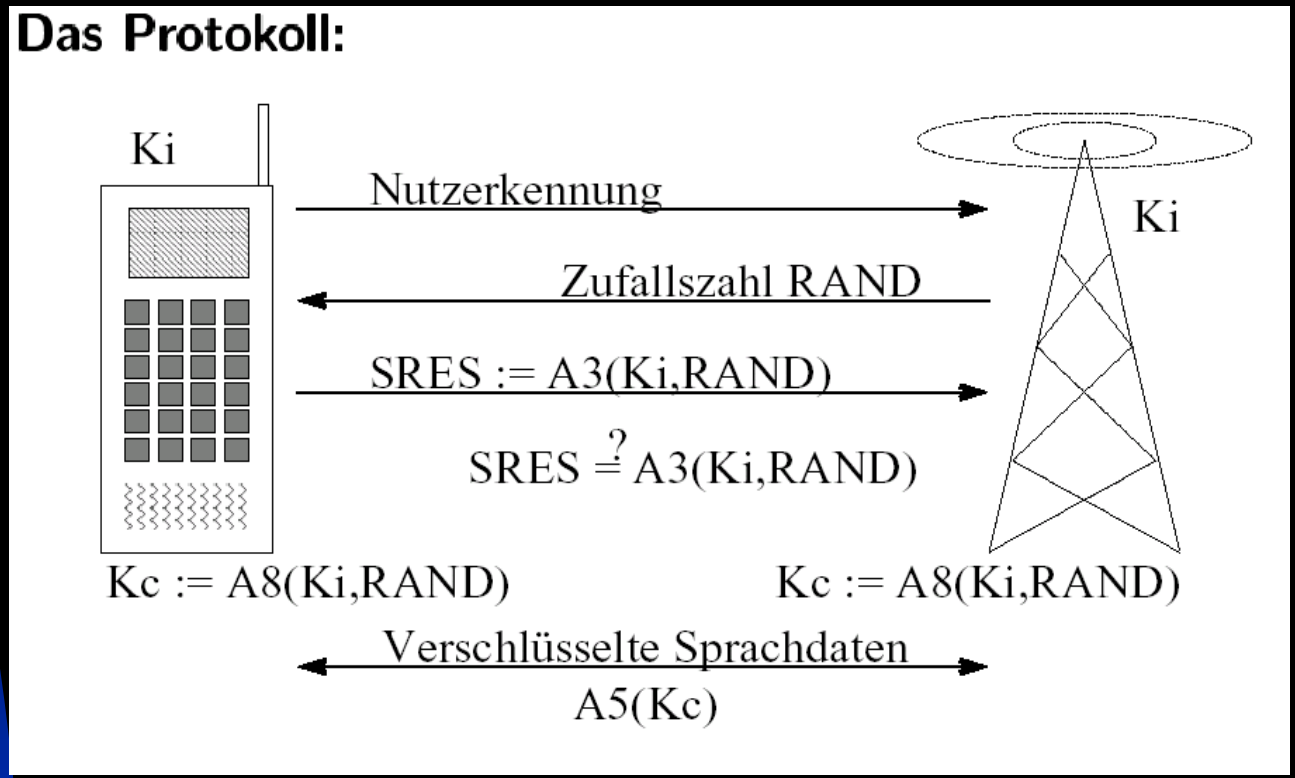


Figure 1.1: *A taxonomy of cryptographic primitives.*

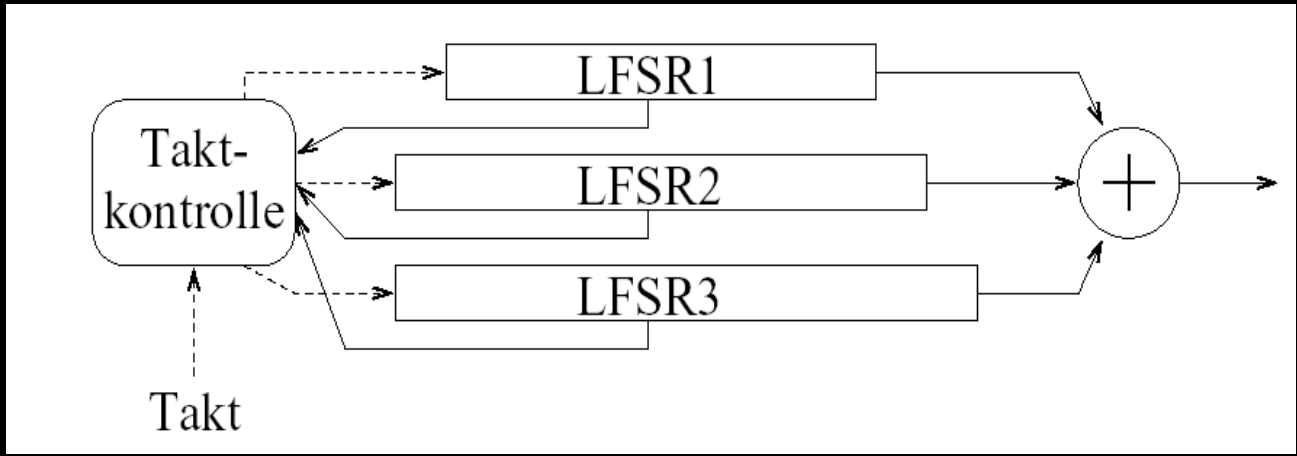
Allgemeines zu A5

- Ein Verfahren zur Verschlüsselung des Informationsflusses im GSM Mobilfunk.
- Eine geheime Entwicklung. Sie wurde nie offiziell veröffentlicht.
- Es existieren 2 Varianten mit unterschiedlichem Sicherheitsniveau.

A5 Funktionsweise



A5 PZG



LSFR1: 19 Bit,

LFSR2: 22 Bit,

LFSR3: 23 Bit,

gesamt: 64 Bit

A5 PZG

- Die Feedback-Polynome der drei LFSR sind bekannt:

  $C_1(x) = x^{19} + x^5 + x^2 + x + 1$
  $C_2(x) = x^{22} + x + 1$
  $C_3(x) = x^{23} + x^{15} + x^2 + x + 1$

- Die „mittleren“ Bits der LFSR dienen als Input für die Taktkontrollfunktion.
- Es werden in jeden Schritt mindestens 2 Register getaktet

A5 Arbeitsweise

- Einsatz des A5 zur Verschlüsselung digitalisierter Sprachdaten.
- GSM sendet in kurzen Abständen Datenblöcke („Frames“).
- Ein Frame enthält bis zu 228 Datenbits (114 für jede Kommunikationsrichtung bei „full duplex“ Arbeitsweise).
- Zu jedem Frame gehört eine (öffentlich bekannte) Frame-Nummer (22 Bit).

A5 Arbeitsweise

- Resynchronisation vor jedem Frame: Setze A5 auf Initialzustand (=Schlüssel)
- Generiere (auf nichtlineare Weise) aus Initialzustand und Frame-Nummer den Startzustand für den Frame.

A5 Arbeitsweise

- Die ersten 100 Outputbits nach der Initialisierung werden verworfen.
- Die nachfolgenden 228 Bits werden zur Verschlüsselung benutzt.

A5 Angriff

- Neuster Angriff (vollständiges Knacken von A5) von Biryukov/Shamir/Wagner (1999) benötigt Vorberechnungen mit einem Zeitaufwand zwischen $O(2^{38})$ und $O(2^{48})$ und Speicherbedarf von ca. 140 GB bis 280 GB.
- Dies ist heutzutage mit einen handelsüblichen PC durchführbar!