1 Klassische Zufallsgeneratoren: Kongruenzgeneratoren und Schieberegister

1.1 Verschiedene Stufen der Bitstrom-Verschlüsselung

Stufe 1: Periodischer Schlüssel

Hier wird eine mehr oder weniger lange Bitfolge als Schlüssel periodisch wiederholt. Technisch handelt es sich um eine BELASO-Chiffre über dem Alphabet $\mathbb{F}_2$, und gebrochen wird sie wie andere periodische polyalphabetische Chiffren auch durch Periodenanalyse oder Finden eines wahrscheinlichen Wortes.

Stufe 2: Lauftext

Hier wird eine vorhandene Bitfolge als Schlüssel verwendet, z. B. der Inhalt einer CD ab einer bestimmten Stelle. Die Analyse verläuft nach den Methoden aus Kapitel I.5. Außerdem ist, sobald die Quelle der Bits, etwa die CD, dem Gegner bekannt ist, der Schlüsselraum viel zu klein – das lineare Durchprobieren von 700 MB Daten ist wenig aufwendig.

Stufe 4: One Time Pad

Das Extrem auf der sicheren Seite. Wegen der aufwendigen Schlüsselverteilung ist es allerdings für eine Massenanwendung nicht geeignet.

Stufe 3: Pseudozufallsfolgen

Der realistische Mittelweg. Hier wird versucht, die idealen Eigenschaften des One Time Pad zu approximieren, indem man statt einer „echten“ Zufallsfolge eine von einem Algorithmus („Zufallsgenerator“) aus einem „effektiven Schlüssel“ (= kurzen Startwert) erzeugte „pseudozufällige“ Bitfolge verwendet. Sogar bei mäßiger Qualität des Zufallsgenerators ist der Geheimtext dann resistent gegen statistische Analysen. Es bleibt das Problem, die Sicherheit gegen einen Angriff mit bekanntem Klartext in den Griff zu bekommen. Diesem Problem ist der Rest des Kapitels gewidmet.

1.2 Allgemeine Diskussion der Bitstrom-Verschlüsselung

Vorteile

- Der Verschlüsselungsalgorithmus und der Entschlüsselungsalgorithmus sind identisch, ...
- ... extrem einfach ...
- ... und sehr schnell – vorausgesetzt die Schlüsselfolge ist schon vorhanden. Für hohe Datenübertragungsraten kann man evtl. den Schlüsselbitstrom auf beiden Seiten vorherberechnen.
- Bei gut gewählter Schlüsselerzeugung ist sehr hohe Sicherheit möglich.

Nachteile

- Das Verfahren ist anfällig gegen Klartextraten; jedes erratene Klartextbit ergibt ein Schlüsselbit.
- Die Qualität der Schlüsselfolge ist sehr kritisch.
- Es gibt keine Diffusion – bei Blockchiffren war das ein sehr wichtiges Kriterium.
- Der Angreifer kann bei bekanntem Klartextstück das entsprechende Schlüsselstück ermitteln und dann den Klartext beliebig austauschen – z. B. „ich liebe dich“ durch „ich hasse dich“ ersetzen oder einen Geldbetrag von 1000 auf 9999 ändern.

Im Zusammenhang mit dem ersten Punkt hat der gewöhnliche Zeichensatz für Texte eine systematische Schwachstelle: Die Kleinbuchstaben `a..z` beginnen im 8-Bit-Code alle mit `011`, die Großbuchstaben `A..Z` alle mit `010`. Eine vermutete Folge von sechs Kleinbuchstaben enthüllt 6*3 = 18 Schlüsselbits.

[Das Auftreten vieler Nullen in den Leitbits der Bytes ist übrigens ein sehr wichtiges Erkennungsmerkmal für natürlichsprachigen Text in europäischen Sprachen.]

Kryptographische Sicherheit von Zufallsgeneratoren

Die entscheidende Frage an eine Pseudozufallsfolge bzw. an den sie erzeugenden Zufallsgenerator ist:

> Kann man aus einem bekannten (auch fragmentierten) Stück der Folge weitere Bits – vorwärts oder rückwärts – bestimmen?

Die Antwort für die „klassischen“, in statistischen Anwendungen und Simulationen verwendeten Zufallsgeneratoren wird JA sein. Wir werden aber auch Zufallsgeneratoren kennen lernen, die in diesem Sinne – vermutlich – kryptographisch sicher sind.

1.3 Lineare Kongruenzgeneratoren

Die erste wichtige Klasse von elementaren – „klassischen“ – Zufallsgeneratoren sind diejenigen einstufig rekurrenten, die lineare Kongruenzen verwenden. Sie haben zunächst den Vorteil, dass sie sehr schnell sind. Sie erzeugen aber auch lange Perioden, und ihre Zufallsqualitäten lassen sich wegen ihrer einfachen Bauart leicht theoretisch absichern.

Die Zufallserzeugung mit linearen Kongruenzen geht so:

$$x_n = s(x_{n-1}) \text{ mit}$$

$$s : \mathbb{Z}/m\mathbb{Z} \longrightarrow \mathbb{Z}/m\mathbb{Z}, \quad s(x) = ax + b \bmod m.$$

Die Folge hängt also von vier ganzzahligen Parametern ab; diese sind

- der **Modul** m mit $m \geq 2$,
- der **Multiplikator** $a \in [0 \ldots m-1]$,
- das **Inkrement** $b \in [0 \ldots m-1]$,
- der **Startwert** $x_0 \in [0 \ldots m-1]$.

Ein solcher Zufallsgenerator heißt **linearer Kongruenzgenerator**, wobei man im Fall $b = 0$ auch von einem **multiplikativen Generator**, im Falle $b \neq 0$ von einem **gemischten Kongruenzgenerator** spricht. Ein solcher Generator ist sehr einfach zu programmieren, selbst in Assembler, und ist sehr schnell. Gut ist er, *wenn die Parameter m, a, b geeignet gewählt sind.* Der Startwert ist dagegen völlig problemlos frei wählbar. Auch das ist wichtig, um bei Bedarf die erzeugten Zufallszahlen genügend variieren zu können.

Bei der Anwendung für die Bitstrom-Verschlüsselung wird der Startwert x_0 oder aber der ganze Parametersatz (m, a, b, x_0) als effektiver Schlüssel betrachtet, d. h., geheim gehalten.

Bemerkungen

1. Da nur endlich viele Werte x_n möglich sind, ist die Folge periodisch mit einer Periodenlänge $\leq m$; dabei kann auch am Anfang eine Vorperiode auftreten.

2. Die Wahl von $a = 0$ ist offensichtlich unsinnig. Aber auch $a = 1$ erzeugt eine Folge, nämlich $x_0, x_0 + b, x_0 + 2b, x_0 + 3b, \ldots$, die nicht brauchbar ist, weil sie auch mod$m$ immer wieder lange regelmäßige Stücke enthält.

3. Für $m = 13$, $a = 6$, $b = 0$, $x_0 = 1$ wird die Folge

$$6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11, 1$$

der Periodenlänge 12 erzeugt, die der Vorstellung einer zufälligen Permutation der Zahlen 1 bis 12 schon recht nahe kommt (trotz des sehr kleinen Moduls).

4. Nimmt man statt dessen den Multiplikator 7, so entsteht die deutlich weniger sympathische Folge

$$7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1.$$

5. Ist a zu m teilerfremd, so ist die Folge rein-periodisch (d. h., es gibt keine Vorperiode). Es ist nämlich $a \bmod m$ invertierbar, also $ac \equiv 1 \pmod{m}$ für ein c. Daher ist stets $x_{n-1} = cx_n - cb \bmod m$. Ist nun $x_{\mu+\lambda} = x_\mu$ mit $\mu \geq 1$, so auch $x_{\mu+\lambda-1} = x_{\mu-1}$ usw., schließlich $x_\lambda = x_0$.

6. Durch Induktion beweist man sofort

$$x_k = a^k x_0 + (1 + a + \cdots + a^{k-1}) \cdot b \bmod m$$

für alle k – ein krasser Hinweis darauf, wie wenig zufällig die Folge in Wirklichkeit ist: Sogar der direkte Zugriff auf ein beliebiges Folgenglied ist möglich, denn der Koeffizient von b ist $(a^k - 1)/(a - 1)$, wobei die Division mod m vorzunehmen ist.

7. Sei $m = 2^e$ und a gerade. Dann ist

$$x_k = (1 + a + \cdots + a^{e-1}) \cdot b \bmod m$$

für alle $k \geq e$, die Periode also 1. Allgemein verkürzen gemeinsame Teiler von a und m die Periode und sind daher zu vermeiden.

8. Sei d ein Teiler des Moduls m. Die Folge $y_n = x_n \bmod d$ ist dann die entsprechende Kongruenzfolge zum Modul d, also $y_n = ay_{n-1} + b \bmod d$. Die Folge (x_n) hat $\bmod d$ also eine Periode $\leq d$, die eventuell sehr kurz ist.

9. Besonders drastisch ist dieser Effekt im Fall einer Zweierpotenz, $m = 2^e$, zu sehen: Das niedrigste Bit von x_n hat dann bestenfalls die Periode 2, ist also abwechselnd 0 und 1, wenn es nicht überhaupt konstant ist. Die k niedrigsten Bits zusammen haben höchstens die Periode 2^k.

10. Ein Modul m mit vielen Teilern, insbesondere eine Zweierpotenz, ist also gegenüber einem Primzahlmodul bei der Zufallserzeugung gehandikapt. Die Qualität ist aber oft doch noch ausreichend, wenn man die erzeugten Zahlen durch m dividiert, also als Zufallszahlen im reellen Intervall $[0, 1[$ ansieht, und am rechten Ende großzügig rundet. Für kryptographische Anwendungen sind solche Moduln aber sicher nicht geeignet.

11. Im Beispiel $m = 2^{32}$, $a = 4095 = 2^{12} - 1$, $b = 12794$ sind die Parameter nicht geeignet gewählt: Aus $x_0 = 253$ ergibt sich $x_1 = 1048829$ und $x_2 = 253 = x_0$.

    Beliebte Moduln sind

    - $m = 2^{32}$, weil er den 32-Bit-Bereich ausschöpft und außerdem sehr effizient handhabbar ist,
    - $m = 2^{31} - 1$, weil dies oft die maximale darstellbare Ganzzahl ist und weil man damit fast so effizient rechnen kann wie mit einer Zweierpotenz. Ein weiterer Vorteil: Diese Zahl ist eine Primzahl (von MERSENNE 1644 behauptet, von EULER 1772 bewiesen), und das hat gute Auswirkungen auf die Qualität der erzeugten Zufallsfolge. Allgemeiner sind FERMAT-Primzahlen $2^k + 1$ und MERSENNE-Primzahlen $2^k - 1$ ähnlich gut geeignet; die nächste solche Zahl ist $2^{61} - 1$.

Die ersten 100 Glieder einer Folge, die mit dem Modul $m = 2^{31} - 1 = 2147483647$, dem Multiplikator $a = 397204094$, dem Inkrement $b = 0$ und dem Startwert $x_0 = 58854338$ erzeugt wurde, zeigt Tabelle 1. In Abbildung 1 ist diese Information visuell umgesetzt. Man sieht daran schon eine deutliche Regellosigkeit der Folge. Es wird aber auch klar, dass ein solcher visueller Eindruck wohl kaum ausreicht, um die Qualität einer Zufallsfolge zu beurteilen.

Abbildung 1: Eine lineare Kongruenzfolge. Waagerechte Achse: Zähler von 0 bis 100, senkrechte Achse: Größe des Folgenglieds von 0 bis $2^{31} - 1$.

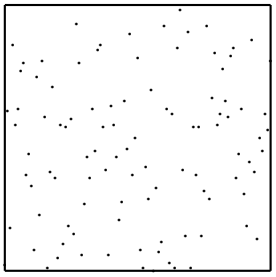


Tabelle 1: 100 Glieder einer linearen Kongruenzfolge

1292048469	319941267	173739233	1992841820
345565651	2011011872	31344917	592918912
1827933824	1691830787	857231706	1416540893
1184833417	145217588	589958351	1776690121
1330128247	558009026	1479515830	1197548384
1627901332	929586843	19840670	1268974074
1682548197	760357405	666131673	1642023821
787305132	1314353697	167412640	1377012759
963849348	971229179	247170576	1250747100
703109068	1791051358	1978610456	1746992541
177131972	1844679385	1328403386	1811091691
1586500120	1175539757	74957396	753264023
468643347	821920620	1269873360	963348259
1698955999	139484430	30476960	1327705603
1266305157	1337811914	1808105128	640050202
37935526	1185470453	2111728842	380228478
808553600	934194915	824017077	881361640
1492263703	414709486	298916786	1883338449
771128019	558671080	1935988732	798347213
120356246	1378842534	37149011	272238278
1190345324	1006355270	1161592162	1079789655
220609946	1918105148	791775291	979447727
1160648370	779600833	1170336930	1271974642
375813045	1089009771	280197098	1144249742
1236647368	1729816359	650188387	1714906064

1.4 Die maximale Periode

Wann hat ein linearer Kongruenzgenerator zum Modul m die maximal mögliche Periode m? Für einen multiplikativen Generator ist das nicht möglich, weil man vom Folgenglied 0 nie mehr wegkommt. Für diese Frage sind also nur gemischte Kongruenzgeneratoren von Interesse. Der triviale Generator mit erzeugender Funktion $s(x) = x + 1 \bmod m$ zeigt, dass dann die Periodenlänge m möglich ist; er zeigt natürlich auch, dass die maximale Periodenlänge noch lange nicht hinreicht, um die Qualität eines Zufallsgenerators nachzuweisen. Das allgemeine Ergebnis ist leicht formuliert:

Satz 1 (Hull/Dobell **1962,** Knuth**)** *Der lineare Kongruenzgenerator mit erzeugender Funktion $s(x) = ax + b \bmod m$ hat genau dann die Periode m, wenn folgende drei Bedingungen erfüllt sind:*

(i) *b und m sind teilerfremd.*

(ii) *Jeder Primteiler p von m teilt auch $a - 1$.*

(iii) *Ist m durch 4 teilbar, so auch $a - 1$.*

Die erste Bedingung bedeutet insbesondere $b \neq 0$, so dass also wirklich ein gemischter Kongruenzgenerator vorliegt. Dem Beweis wird ein Hilfssatz vorangestellt (und es werden zwei weitere Hilfssätze aus Kapitel III, Anhang A.1 verwendet).

Hilfssatz 1 *Sei $m = m_1 m_2$ mit teilerfremden natürlichen Zahlen m_1 und m_2. Seien λ, λ_1 und λ_2 die Perioden der Kongruenzgeneratoren $x_n = s(x_{n-1}) \bmod m$ bzw. $\bmod m_1$ bzw. $\bmod m_2$ zum Startwert x_0. Dann ist λ das kleinste gemeinsame Vielfache von λ_1 und λ_2.*

Beweis. Seien $x_n^{(1)}$ und $x_n^{(2)}$ die entsprechenden Folgeglieder für m_1 bzw. m_2. Dann ist $x_n^{(i)} = x_n \bmod m_i$. Da $x_{n+\lambda} = x_n$ für alle genügend großen n, folgt sofort, dass λ ein Vielfaches von λ_1 und λ_2 ist. Umgekehrt folgt aus $m \mid t \iff m_1, m_2 \mid t$, dass

$$x_n = x_k \iff x_n^{(i)} = x_k^{(i)} \quad \text{für } k = 1 \text{ und } 2.$$

Also ist λ höchstens gleich dem kleinsten gemeinsamen Vielfachen von λ_1 und λ_2. $\diamond$

Beweis des Satzes. Für beide Beweisrichtungen kann man nach dem Hilfssatz 1 o. B. d. A. $m = p^e$ mit einer Primzahl p annehmen.

„$\Longrightarrow$“: Da jede Zahl in $[0 \ldots m-1]$ genau einmal vorkommt, darf man o. B. d. A. $x_0 = 0$ annehmen. Dann ist

$$x_n = (1 + a + \cdots + a^{n-1}) \cdot b \bmod m \quad \text{für alle } n.$$

Da x_n auch den Wert 1 annimmt, muss schon mal b zu m teilerfremd sein. Da $x_m = 0$, folgt nun $m \mid 1 + a + \cdots + a^{m-1}$, also

$$p \mid m \mid a^m - 1 = (a-1)(1 + a + \cdots + a^{m-1}).$$

Nach dem kleinen Satz von FERMAT ist $a^p \equiv a \pmod{p}$, also $a^m = a^{p^e} \equiv a^{p^{e-1}} \equiv \ldots \equiv a \pmod{p}$, also $p \mid a - 1$. Die Aussage (iii) ist der Fall $p = 2$ mit $e \geq 2$. Wegen der Aussage (ii) muss a schon mal ungerade sein. Wäre nun $a \equiv 3 \pmod{4}$, so nach Hilfssatz 1 in III.A.1 bereits $x_{m/2} = 0$. Also muss $a \equiv 1 \pmod{4}$ sein.

„$\Longleftarrow$“: Auch hier kann man wieder o. B. d. A. $x_0 = 0$ annehmen. Dann ist

$$x_n = 0 \Longleftrightarrow m \mid 1 + a + \cdots + a^{n-1}.$$

Insbesondere ist der Fall $a = 1$ trivial. Sei also o. B. d. A. $a \geq 2$. Dann ist weiter

$$x_n = 0 \Longleftrightarrow m \mid \frac{a^n - 1}{a - 1}.$$

Zu zeigen ist:

- $m \mid \frac{a^m-1}{a-1}$ – dann ist $\lambda | m$;
- m kein Teiler von $\frac{a^{m/p}-1}{a-1}$ – da m eine p-Potenz ist, folgt dann $\lambda \geq m$.

Sei p^h die maximale Potenz, die in $a-1$ aufgeht. Nach Hilfssatz 2 in III.A.1 ist dann

$$a^p \equiv 1 \pmod{p^{h+1}},\ a^p \not\equiv 1 \pmod{p^{h+2}}$$

und sukzessive

$$a^{p^k} \equiv 1 \pmod{p^{h+k}},\ a^{p^k} \not\equiv 1 \pmod{p^{h+k+1}}$$

für alle k. Insbesondere folgt $p^{h+e} \mid a^m - 1$. Da in $a-1$ höchstens p^h aufgeht, folgt $m = p^e \mid \frac{a^m-1}{a-1}$. Wäre $p^e \mid \frac{a^{m/p}-1}{a-1}$, so $p^{e+h} \mid a^{p^{e-1}} - 1$, Widerspruch. $\diamondsuit$

Dieser Satz ist vor allem für Zweierpotenz-Moduln von Interesse; für Primzahl-Moduln dagegen ergibt er kein brauchbares Ergebnis.

Korollar 1 (GREENBERGER 1961) *Ist $m = 2^e$ mit $e \geq 2$, so wird die Periode m genau dann erreicht, wenn gilt:*

(i) *b ist ungerade.*

(ii) *$a \equiv 1 \pmod{4}$.*

Korollar 2 *Ist m eine Primzahl, so wird die Periode m genau dann erreicht, wenn b zu m teilerfremd und $a = 1$ ist.*

Dieses (traurige) Ergebnis lässt sich etwas allgemeiner fassen – auch für beliebige quadratfreie Moduln m gibt es keine brauchbaren linearen Kongruenzgeneratoren der Periode m:

Korollar 3 *Ist m quadratfrei, so wird die Periode m genau dann erreicht, wenn b zu m teilerfremd und $a = 1$ ist.*

Wir haben nun mit Satz 1 die überhaupt größtmögliche Periode erreicht und mit Korollar 1 auch einen brauchbaren Spezialfall gefunden.

1.5 Die maximale Periode multiplikativer Generatoren

Multiplikative Generatoren $x_n = ax_{n-1} \bmod m$ können nie die Periode m erreichen, da das Folgenglied 0 nie mehr verlassen wird. Was können sie bestenfalls? – λ ist im folgenden Satz die CARMICHAEL-Funktion und wurde genau in diesem Zusammenhang erstmals eingeführt.

Satz 2 (CARMICHAEL 1910) *Die maximale Periode eines multiplikativen Generators mit erzeugender Funktion $s(x) = ax \bmod m$ ist $\lambda(m)$. Sie wird insbesondere dann erreicht, wenn gilt:*

(i) *a ist primitiv $\bmod m$.*

(ii) *x_0 ist teilerfremd zu m.*

Beweis. Es ist $x_n = a^n x_0 \bmod m$. Ist $k = \mathrm{Ord}_m a$ die Ordnung von a, so $x_k = x_0$, also die Periode $\leq k \leq \lambda(m)$. Sei nun a primitiv $\bmod m$, also $1, a, \ldots, a^{\lambda(m)-1} \bmod m$ verschieden. Da x_0 zu m teilerfremd ist, folgt, dass die Periode $\lambda(m)$ ist. $\diamond$

Korollar 1 *Ist $m = p$ eine Primzahl, so wird die maximale Periode $\lambda(p) = p - 1$ genau dann erreicht, wenn gilt:*
(i) *a ist primitiv $\bmod p$.*
(ii) *$x_0 \neq 0$.*

Für Primzahlmoduln ist die Situation bei den multiplikativen Generatoren also sehr gut: Die Periode ist nur um 1 kleiner als überhaupt mit einstufiger Rekursion möglich und jeder Startwert außer 0 ist geeignet.

Dieses Ergebnis wird in Abschnitt 1.9 weitgehend verallgemeinert.

1.6 Lineare Schieberegister

Neben den bisher behandelten linearen Kongruenzgeneratoren gibt es eine andere klassische und weitverbreitete Methode zur Erzeugung von Pseudozufallsfolgen: die Schieberegister-Methode. Diese Methode wurde von GOLOMB 1955 erstmals vorgeschlagen, wird aber meist nach TAUSWORTHE benannt, der die Idee 1965 in einer Arbeit aufgriff. Sie ist besonders leicht in Hardware zu realisieren. Für die theoretische Beschreibung fasst man Blöcke von jeweils l Bits als Elemente des Vektorraums $\mathbb{F}_2^l$ über dem Körper $\mathbb{F}_2$ aus zwei Elementen auf.

Eine lineare Abbildung

$$A\colon \mathbb{F}_2^l \longrightarrow \mathbb{F}_2$$

ist nichts anderes als eine Vorschrift, aus einem l-Bit-Block eine Teilsumme zu bilden:

$$Au = \sum_{i=1}^{l} a_i u_i,$$

wobei alle Koeffizienten a_i ja 0 oder 1 sind. Als potenzielle Zufallsfolge wird die Folge von Bits betrachtet, die nach der Vorschrift

$$u_n = a_1 u_{n-1} + \cdots + a_l u_{n-l}$$

entsteht. Man braucht als Parameter des Verfahrens

- die **Registerlänge** l mit $l \geq 2$,
- eine **Rückkopplungsvorschrift** A, die eine Folge $(a_1, \ldots, a_l) \in \mathbb{F}_2^l$ ist, und daher auch durch eine Teilmenge $I \subseteq \{1, \ldots, l\}$ beschrieben werden kann.
- einen **Startwert** $u = (u_{l-1} \ldots u_0)$ aus l Bits.

Die Iterationsformal lässt sich damit auch in der Form

$$u_n = \sum_{j \in I} u_{n-j}$$

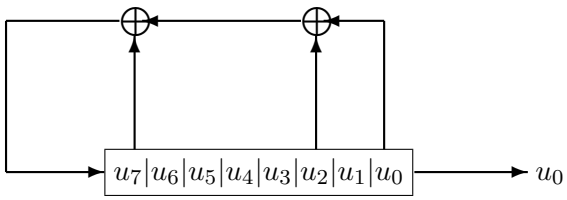
schreiben.

Die Hardware-Realisierung stellt man sich so vor, dass das rechte Bit des Schieberegisters ausgegeben wird, die übrigen $l - 1$ Bits nach rechts nachrücken und auf der linken Seite als „Rückkopplung“ die Summe der durch I angegebenen Bits nachgeschoben wird, siehe Abbildung 2.

Bei der Anwendung für die Bitstrom-Verschlüsselung wird der Startwert u oder aber alle drei Parameter l, I, u als Schlüssel betrachtet, d. h., geheim gehalten.

Bei geschickter Wahl der Parameter, die hier nicht weiter behandelt wird, hat die Folge eine Periode nahe 2^l und ist durch statistische Tests praktisch nicht von einer gleichverteilten Zufallsfolge zu unterscheiden, siehe 1.9 und 1.10.

Abbildung 2: Ein lineares Schieberegister



1.7 Mehrstufige Generatoren

Die gemeinsame Verallgemeinerung von linearen Kongruenzgeneratoren und linearen Schieberegister-Generatoren sind die **mehrstufigen linearen Rekurrenzgeneratoren**. Sie lassen sich bequem im Rahmen eines endlichen Rings R (kommutativ mit 1) behandeln; damit sind nicht nur die Ringe $\mathbb{Z}/m\mathbb{Z}$ erfasst, sondern auch die endlichen Körper zusätzlich zu den Primkörpern $\mathbb{F}_p$, die ebenfalls zur Zufallserzeugung benützt werden können. Bei einem r-stufigen linearen Rekurrenzgenerator wird eine Folge (x_n) in R nach der Vorschrift

$$x_n = a_1 x_{n-1} + \cdots + a_r x_{n-r} + b$$

erzeugt. Als Parameter braucht man

- die **Rekursionstiefe** r (o. B. d. A. $a_r \neq 0$),
- die **Koeffizientenfolge** $a = (a_1, \ldots, a_r) \in R^r$,
- das **Inkrement** $b \in R$,
- einen **Startvektor** $(x_0, \ldots, x_{r-1}) \in R^r$.

Der lineare Rekurrenzgenerator heißt **homogen** oder **inhomogen**, je nachdem, ob $b = 0$ ist oder nicht.

Die Funktionsweise eines linearen Rekurrenzgenerators kann man ähnlich einem Schieberegister veranschaulichen, siehe Abbildung 3.

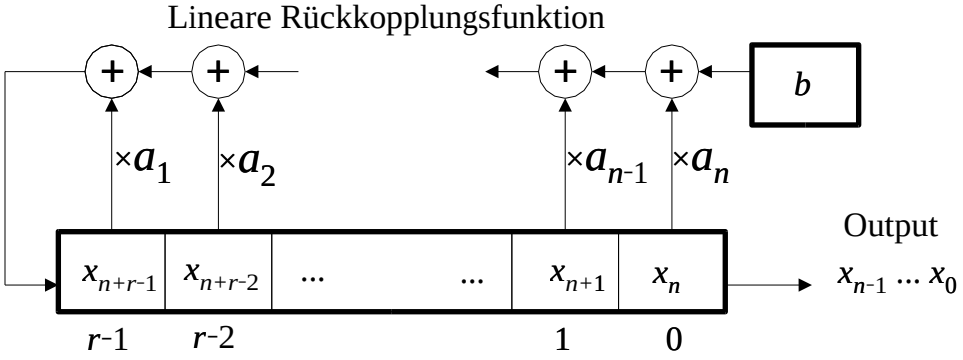


Abbildung 3: Ein linearer Rekurrenzgenerator

Inhomogene lineare Rekurrenzgeneratoren kann man leicht auf homogene reduzieren, wobei man allerdings eine Rekursionsstufe zusätzlich in Kauf nehmen muss: Aus den beiden Gleichungen

$$\begin{aligned} x_{n+1} &= a_1 x_n + \cdots + a_r x_{n-r+1} + b, \\ x_n &= a_1 x_{n-1} + \cdots + a_r x_{n-r} + b, \end{aligned}$$

folgt nämlich durch Subtraktion

$$x_{n+1} = (a_1 + 1)x_n + (a_2 - a_1)x_{n-1} \cdots + (-a_r)x_{n-r}.$$

Im Falle $r = 1$, $x_n = ax_{n-1} + b$, wird diese Formel zu

$$x_n = (a + 1)x_{n-1} - ax_{n-2}.$$

Daher wird der inhomogene Fall im folgenden vernachlässigt.

Im homogenen Fall kann man unter Verwendung der **Zustandsvektoren** $x_{(n)} = (x_n, \ldots, x_{n+r-1})^t$ schreiben

$$x_{(n)} = Ax_{(n-1)} \quad \text{für } n \geq 1,$$

mit der **Begleitmatrix**

$$A = \begin{pmatrix} 0 & 1 & \ldots & 0 \\ & \ddots & \ddots & \\ & & & 1 \\ a_r & a_{r-1} & \ldots & a_1 \end{pmatrix}.$$

Die nächste Stufe der Verallgemeinerung ist also ein **Matrixgenerator**. Parameter sind:

- eine $r \times r$-Matrix $A \in M_r(R)$,
- ein Startvektor $x_0 \in R^r$.

Die Folge wird gebildet nach der Formel

$$x_n = Ax_{n-1} \in R^r.$$

1.8 Allgemeine lineare Generatoren

Noch allgemeiner (und begrifflich einfacher) ist die abstrakt-algebraische Version, der **allgemeine lineare Generator**. Gegeben sind:

- ein Ring R (kommutativ und mit Einselement),
- ein R-Modul M,
- eine R-lineare Abbildung $A : M \longrightarrow M$,
- ein Startwert $x_0 \in M$.

Daraus wird eine Folge $(x_n)_{n\in\mathbb{N}}$ gebildet nach der Formel

$$x_n = Ax_{n-1} \quad \text{für } n \geq 1.$$

Beispiele

1. Für einen homogenen linearen Kongruenzgenerator ist

$$R = \mathbb{Z}/m\mathbb{Z}, \quad M = R \quad (r = 1), \quad A = (a).$$

2. Für einen inhomogenen linearen Kongruenzgenerator ist

$$R = \mathbb{Z}/m\mathbb{Z}, \quad M = R^2 \quad (r = 2), \quad A = \begin{pmatrix} 0 & 1 \\ -a & a+1 \end{pmatrix}.$$

3. Für ein lineares Schieberegister ist

$$R = \mathbb{F}_2, \quad M = \mathbb{F}_2^l \quad (r = l), \quad A = \text{die Begleitmatrix},$$

   die nur aus Nullen und Einsen besteht.

Falls M endlich ist, kann die Rekursion nur endlich viele verschiedene Werte annehmen, muss also nach einer eventuellen Vorperiode periodisch werden.

Satz 3 *Sei M ein endlicher R-Modul und $A : M \longrightarrow M$ linear. Genau dann, wenn A bijektiv ist, sind alle vom zugehörigen allgemeinen linearen Generator erzeugten Folgen rein-periodisch.*

Beweis. Sei A bijektiv und x_0 ein Startvektor. Sei t der kleinste Index, so dass x_t einen bereits vorher durchlaufenen Wert annimmt, und sei s der kleinste Index mit $x_t = x_s$. Wäre $s \geq 1$, so $x_s = Ax_{s-1}$ und $x_t = Ax_{t-1}$, also

$$x_{t-1} = A^{-1}x_t = A^{-1}x_s = x_{s-1},$$

im Widerspruch zur Minimalität von t.

Sei umgekehrt A nicht bijektiv; da M endlich ist, ist A dann auch nicht surjektiv. Man kann also $x_0 \in M - A(M)$ wählen. Dann kann niemals $x_0 = Ax_t$ sein, die Folge ist also nicht reinperiodisch. $\diamond$

Dieses Ergebnis lässt sich über die Begleitmatrix auf homogene mehrstufige Kongruenzgeneratoren, insbesondere auf lineare Schieberegister anwenden:

Korollar 1 *Ein homogener linearer Kongruenzgerator der Rekursionstiefe r erzeugt stets rein-periodische Folgen, wenn der Koeffizient a_r in $\mathbb{Z}/m\mathbb{Z}$ invertierbar ist. Ein lineares Schieberegister der Länge l erzeugt rein-periodische Folgen, wenn der Rückkopplungskoeffizient $a_l \neq 0$ ist.*

Die erste Aussage gilt auch im nicht-homogenen Fall, da die Formel

$$x_{n-r} = a_r^{-1}(x_n - a_1 x_{n-1} - \cdots - a_{r-1} x_{n-r+1} - b)$$

für die Rückwärtsberechnung der Folge sorgt.

1.9 Matrixgeneratoren über endlichen Körpern

Ein Matrix-Generator über einem Körper K wird durch eine $r \times r$-Matrix

$$A \in M_r(K)$$

vollständig beschrieben (bis auf die Wahl des Startvektors $x_0 \in K^r$). Das Ziel dieses Abschnitts ist die Charakterisierung der Folgen mit maximaler Periodenlänge.

Im Polynomring $K[T]$ in einer Unbestimmten T bildet die Menge

$$\{\rho \in K[T] \mid \rho(A) = 0\}$$

ein Ideal. Da $K[T]$ Hauptidealring ist (sogar euklidischer Ring), wird dieses Ideal von einem eindeutig bestimmten normierten Polynom μ erzeugt; dieses heißt das **Minimalpolynom** von A. Da A auch Nullstelle seines charakteristischen Polynoms χ ist, gilt also $\mu|\chi$. Ist A invertierbar, so ist das absolute Glied von μ nicht 0; denn sonst hätte μ die Nullstelle 0 und A den Eigenwert 0.

Hilfssatz 2 *Sei K ein Körper, $A \in GL_r(K)$ von endlicher Ordnung t, μ das Minimalpolynom von A, $s = \operatorname{Grad} \mu$, $R := K[T]/\mu K[T]$ und $a \in R$ die Restklasse von T. Dann gilt:*

$$a^k = 1 \iff \mu | T^k - 1 \iff A^k = \mathbf{1}.$$

Insbesondere ist $a \in R^\times$, t auch die Ordnung von a und $\mu | T^t - 1$.

Beweis. R ist eine K-Algebra der Dimension s. Ist $\mu = b_s T^s + \cdots + b_0$ (wobei $b_s = 1$), so

$$\mu - b_0 = T \cdot (b_s T^{s-1} + \cdots + b_1);$$

da $b_0 \neq 0$, ist also T mod μ invertierbar, also $a \in R^\times$. Da a^k die Restklasse von T^k ist, folgt die behauptete Äquivalenzkette. $\diamond$

Korollar 1 *Ist K ein endlicher Körper mit q Elementen, so ist*

$$t \leq \#R^\times \leq q^s - 1 \leq q^r - 1.$$

Sei von jetzt an K ein endlicher Körper mit q Elementen. Dann ist auch die Gruppe $GL_r(K)$ der invertierbaren $r \times r$-Matrizen endlich. Der Vektorraum K^r besteht aus q^r Vektoren. Wir wissen bereits, dass jede Folge, die von dem Matrixgenerator zu A erzeugt wird, rein-periodisch ist. Eine volle Periode wird immer vom Nullvektor $0 \in K^r$ alleine gebildet. Alle übrigen Vektoren werden im allgemeinen auf mehrere Perioden aufgeteilt sein. Ist s die Länge einer solchen Periode und x_0 der entsprechende Startvektor, so

ist $x_0 = x_s = A^s x_0$. Also hat A^s den Eigenwert 1 und folglich A eine s-te Einheitswurzel als Eigenwert.

Denkbar ist aber auch, dass alle Vektoren $\neq 0$ zusammen eine Periode der maximal möglichen Länge $q^r - 1$ bilden. In diesem Fall gilt $A^s x = x$ für alle Vektoren $x \in K^r$ mit $s = q^r - 1$, aber für keinen kleineren Exponenten > 0. Also ist $t = q^r - 1$ die Ordnung von A. Damit ist gezeigt:

Korollar 2 *Ist K endlich mit q Elementen, so gilt:*

(i) *Erzeugt der Matrixgenerator zu A für einen Startvektor $\neq 0$ eine Folge der Periode s, so hat A eine s-te Einheitswurzel als Eigenwert.*
(ii) *Gibt es eine Periode der Länge $q^r - 1$, so ist $t = q^r - 1$ die Ordnung von A.*

Hilfssatz 3 *Sei K ein endlicher Körper mit q Elementen und $\varphi \in K[T]$ ein irreduzibles Polynom vom Grad d. Dann gilt $\varphi | T^{q^d - 1} - 1$.*

Beweis. Der Restklassenring $R = k[T]/\varphi K[T]$ ist ein Erweiterungskörper vom Grad $d = \mathrm{Dim}_K R$, hat also $h := q^d$ Elemente und enthält mindestens eine Nullstelle a von φ, nämlich die Restklasse von T. Da jedes $x \in R^\times$ die Gleichung $x^{h-1} = 1$ erfüllt, ist insbesondere a auch Nullstelle von $T^{h-1} - 1$. Also ist $\mathrm{ggT}(\varphi, T^{h-1} - 1)$ nicht konstant. Da φ irreduzibel ist, folgt $\varphi | T^{h-1} - 1$. $\diamond$

Definition. Ein Polynom $\varphi \in K[T]$ vom Grad d über dem endlichen Körper K mit q Elementen heißt **primitiv**, wenn φ irreduzibel und kein Teiler von $T^k - 1$ ist für $1 \leq k < q^d - 1$.

Hauptsatz 1 *Sei K ein endlicher Körper mit q Elementen und $A \in GL_r(K)$. Dann sind folgende Aussagen äquivalent:*

(i) *Der Matrixgenerator zu A erzeugt eine Folge der Periode $q^r - 1$.*
(ii) *A hat die Ordnung $q^r - 1$.*
(iii) *Das charakteristische Polynom χ von A ist primitiv.*

Beweis. „(i) $\Longrightarrow$ (ii)“: Siehe Korollar 2 (ii).

„(ii) $\Longrightarrow$ (iii)“: In Korollar 1 ist $t = q^r - 1$. Also ist $\#R^\times = q^s - 1$, also R ein Körper und daher μ irreduzibel. Ferner ist $s = r$, also $\mu = \chi$, und μ nach Hilfssatz 2 kein Teiler von $T^k - 1$ für $1 \leq k < q^r - 1$, also μ primitiv.

„(iii) $\Longrightarrow$ (i)“: Da χ irreduzibel ist, ist $\chi = \mu$. Die Restklasse a von T ist Nullstelle von μ und hat nach der Definition von „primitiv“ die multiplikative Ordnung $q^r - 1$. Da das Potenzieren mit q ein Automorphismus des Körpers R ist, der K elementweise festlässt, sind auch die r Potenzen a^{q^k} für $0 \leq k <$

r Nullstellen von μ, und zwar alle verschieden. Dies müssen daher sämtliche Nullstellen sein, und alle haben die multiplikative Ordnung $q^r - 1$. Daher hat A keinen Eigenwert von geringerer Ordnung und daher gibt es nach Korollar 2 (i) auch keine kürzere Periode. $\diamond$

Für ein lineares Schieberegister ist A die Begleitmatrix wie in 1.7. Das charakterische Polynom ist also $T^l - a_1 T^{l-1} - \cdots - a_l$.

Korollar 1 *Ein lineares Schieberegister der Länge l erzeugt genau dann eine Folge der maximal möglichen Periode $2^l - 1$, wenn sein charakteristisches Polynom primitiv und der Startwert $\neq 0$ ist.*

Die Konstruktion von linearen Schieberegistern, die Folgen maximaler Periode erzeugen, ist also auf die Konstruktion primitiver Polynome über dem Körper $\mathbb{F}_2$ zurückgeführt.

Im eindimensionalen Fall $r = 1$ erhalten wir speziell den multiplikativen Generator mit der Rekursionsvorschrift $x_n = ax_{n-1}$ über dem endlichen Körper K mit q Elementen. Die zugehörige Matrix $A = (a)$ bewirkt die Multiplikation mit a, also ist a der einzige Eigenwert und $\chi = T - a \in K[T]$ das charakteristische Polynom. Dieses ist, da linear, in jedem Fall irreduzibel. Da

$$\chi | T^k - 1 \Longleftrightarrow a \text{ Nullstelle von } T^k - 1 \Longleftrightarrow a^k = 1,$$

ist χ also genau dann primitiv, wenn a erzeugendes Element der multiplikativen Gruppe $K^\times$, also primitives Element ist. Damit ist die folgende leichte Verallgemeinerung des Korollars zu Satz 2 gezeigt:

Korollar 2 *Ein multiplikativer Generator über K mit Multiplikator a erzeugt genau dann eine Folge der Periode $q - 1$, wenn a primitives Element und der Startwert $x_0 \neq 0$ ist.*

1.10 Statistische Eigenschaften von linearen Schieberegistern

Die statistischen Eigenschaften von Schieberegisterfolgen der maximalen Periode $2^l - 1$, wobei l die Länge des Schieberegisters ist, wurden bereits von GOLOMB ausführlich untersucht.

Referenz:

> Solomon E. GOLOMB: **Shift Register Sequences.** Revised Edition, Aegean Park Press, Laguna Hills 1982. ISBN 0-89412-048-4

Hier einige Aussagen dazu:

Bemerkungen

1. In jeder vollen Periode kommen genau 2^{l-1} Einsen und $2^{l-1} - 1$ Nullen vor.

   Beweis. Es werden alle 2^l Zustandsvektoren $\in \mathbb{F}_2^l$ außer 0 jeweils genau einmal angenommen; das entspricht den ganzen Zahlen im Intervall $[1 \dots 2^l - 1]$. Davon sind 2^{l-1} ungerade, der Rest gerade, und ihre Paritäten bilden genau die Output-Folge des Schieberegisters.

2. Ein **Run** in einer Folge ist ein maximales konstantes Stück.

   Beispiel: $\dots 0111110 \dots$ ist ein Einser-Run der Länge 5.

   Bedenkt man, dass die Stücke der Länge l der Schieberegister-Folge genau die verschiedenen Zustandsvektoren $\neq 0$ sind, so ist klar, dass in der vollen Periode folgendes vorkommt:

   - Kein Run der Länge $> l$.
   - Genau ein Einser-Run und kein Nuller-Run der Länge l – denn sonst käme der Nuller-Zustand vor bzw. der Einser-Zustand öfter als einmal vor.
   - Jeweils genau ein Einser- und Nuller-Run der Länge $l - 1$.
   - Allgemein jeweils genau 2^{k-1} Einser- und Nuller-Runs der Länge $l - k$ für $1 \leq k \leq l - 1$.
   - Insbesondere genau 2^{l-1} Runs der Länge 1, davon jeweils genau die Hälfte Nullen und Einsen.

3. Für eine periodische Folge $x = (x_n)_{n \in \mathbb{N}}$ in $\mathbb{F}_2$ der Periode s ist die **Autokorrelation** zur Verschiebung um t definiert durch

$$\begin{aligned} \kappa_x(t) &= \frac{1}{s} \cdot [\#\{n \mid x_{n+t} = x_n\} - \#\{n \mid x_{n+t} \neq x_n\}] \\ &= \frac{1}{s} \cdot \sum_{n=0}^{s-1} (-1)^{x_{n+t} + x_n} \end{aligned}$$

(analog wie in Kapitel II für BOOLEsche Funktionen). Wird nun x von einem Schieberegister der Länge l erzeugt,

$$x_n = a_1 x_{n-1} + \cdots + a_l x_{n-l} \quad \text{für } n \geq l,$$

so kann man die Differenzenfolge $y_n = x_{n+t} - x_n$ bilden. Diese wird offensichtlich von dem gleichen Schieberegister erzeugt. Sind die Startwerte $y_0, \ldots, y_{l-1}$ sämtlich 0, so ist der Zustandsvektor $x_{(t)} = x_{(0)}$, also t ein Vielfaches der Periode und $\kappa_x(t) = 1$. Andernfalls – und falls x die maximal mögliche Periode $s = 2^l - 1$ hat – durchläuft y in einer Periode nach Bemerkung 1 genau 2^{l-1} Einsen und $2^{l-1} - 1$ Nullen. Daher ist

$$\kappa_x(t) = \begin{cases} 1, & \text{wenn } s|t, \\ -\frac{1}{s}, & \text{sonst.} \end{cases}$$

Die Autokorrelation ist also – außer bei Verschiebungen um Vielfache der Periode – *gleichmäßig klein.*

Diese Aussagen bedeuten, dass die Folge sehr gleichmäßig verteilt ist, und wurden von GOLOMB als die drei Zufälligkeits-Postulate bezeichnet. Wegen dieser Eigenschaften werden solche Folgen, also insbesondere Schieberegisterfolgen maximaler Periode, in der Elektrotechnik auch als „Rauschen“ bezeichnet (PN-sequences = Pseudo Noise Sequences).

Hier eine Implementation von Schieberegistern in der leicht verständlichen Sprache von Mathematica – für eine Anwendung mit hohem Effizienzbedarf würde man natürlich eine Implementation in C vorziehen.

```
linShRep[n_Integer] :=
  Module[{y, outlist = {}},
    For[i = 0, i < n, i++,
      outlist = Append[outlist, Last[x]];
      y = Mod[a.x, 2];
      x = RotateRight[x];
      x[[1]] = y
      ];
    Return[outlist]
    ]
linShRep::usage =
"Generate a linear feedback shift register sequence.\n
1. Set up the coefficient array a consisting of 0s and 1s.\n
2. Set up the initial state of the shift register as an array
   x of the same length consisting of 0s and 1s.\n
3. Call linShRep with the desired number n of output Bits."
```

Ein exemplarischer Aufruf dieser Funktion mit einem Schieberegister der Länge 16, aus dem 1024 Bits erzeugt werden sollen, sieht so aus:

```
a = {0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}
x = {0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1}
z = linShRep[1024]
```

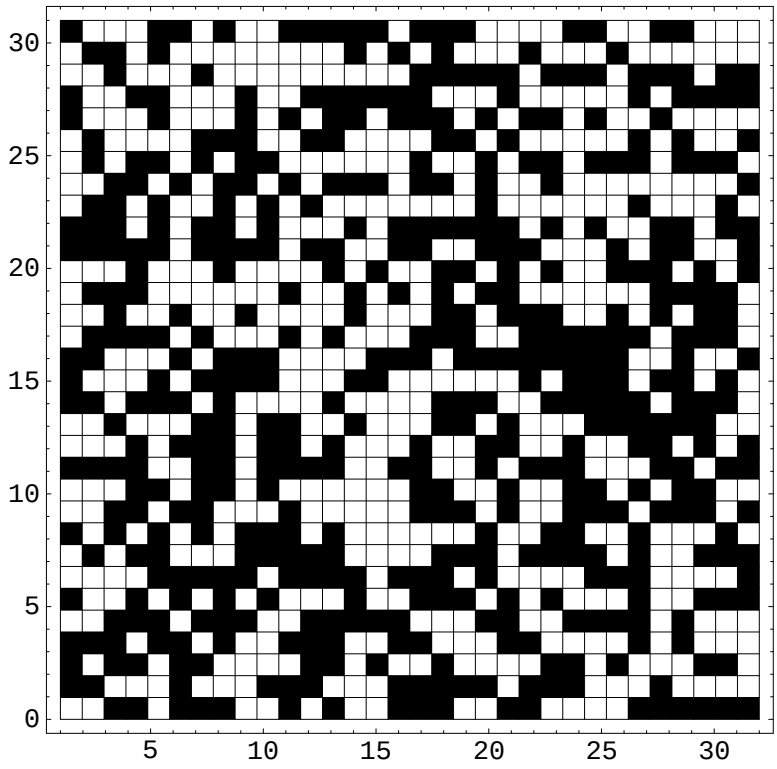
und ergibt den Output (ohne Klammern und Kommata wiedergegeben):

```
11001000110101100011001111000000
00111011100011100000100011101111
01001001111001011011110010111001
00010010110001100111001111010111
11000100011000001110011000010111
01101010101110110001010111011000
11110000010000100010111100011110
10100111000001111000100001011000
01010101000101111110110011011101
11001001110111110001011000100010
11100100101111110011011001010011
00001100100001100110100011100100
11101000100101110110011011001010
11011100100110111001011100000011
00100010111101111000110000010001
01110100001110011111101000100101
00111010001111000100000000110110
10000101110101110001100000010001
11011011011110111001000110101001
10001111110110101010011111100001
11101110111101011001010110001010
00000100001001100110001110100110
00010100101110100000010101100100
10010110101011111110111111011101
11001010010100010010110111111110
10100101001111110110100100010001
10111100011001111001011111010110
01110111010100100010100101101111
01100111011000000111011111010000
11011101111111110000010001000100
10010111111110101011101110111111
01110010110000010001111001100111
```

Eine Visualiserung, die mit dem Mathematica-Kommando

```
DensityPlot[z[[32*i + j]], {j, 1, 32}, {i, 0, 31},
   PlotPoints -> 32]
```

erzeugt wurde, zeigt, dass zumindest der äußere Eindruck der einer ziemlich zufälligen Bitfolge ist:



Das im Beispiel verwendete Schieberegister erzeugt übrigens eine Folge der maximalen Periode $2^{16} - 1 = 65535$, da sein charakteristisches Polynom

$$T^{16} + T^{14} + T^{13} + T^{11} + 1 \in \mathbb{F}_2$$

primitiv ist.