3.8 Korrelationsattacken – die Achillesferse der Kombinierer

Sie $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ die Kombinierfunktion eines nichtlinearen Kombinierers. Die Anzahl

$$K_f := \#\{x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n \mid f(x) = x_1\}$$

misst, wie oft der Funktionswert mit dem ersten Argument übereinstimmt. Ist sie $> 2^{n-1}$, so ist die Wahrscheinlichkeit für diese Übereinstimmung

$$p = \frac{1}{2^n} \cdot K_f > \frac{1}{2},$$

also überdurchschnittlich. Die kombinierte Outputfolge „korreliert“ also stärker mit dem Output des ersten linearen Schieberegisters, als zufällig zu erwarten wäre.

Diesen Effekt kann sich der Kryptoanalytiker bei einem Angriff mit bekanntem Klartext zunutze machen: Die (ersten) Schlüsselbits $b_0, \ldots, b_{r-1}$ seien bekannt. Mit einer Exhaustion über die 2^{l_1} Startvektoren des ersten Registers erzeugt man jedesmal die Folge $u_0, \ldots, u_{r-1}$ und zählt die Koinzidenzen. Zu erwarten ist

$$\frac{1}{2^r} \cdot \#\{i \mid u_i = b_i\} \approx \begin{cases} p & \text{beim richtigen Startvektor,} \\ \frac{1}{2} & \text{sonst.} \end{cases}$$

Falls r groß genug ist, kann man also den echten Startvektor des ersten Registers mit einem Aufwand $\sim 2^{l_1}$ bestimmen. Macht man dann mit den anderen Registern genauso weiter, gelingt die Identifikation des gesamten Schlüssels mit einem Aufwand $\sim 2^{l_1} + \cdots + 2^{l_n}$. Das ist zwar exponentiell, aber wesentlich geringer als der Aufwand $\sim 2^{l_1} \cdots 2^{l_n}$ für die naive vollständige Schlüsselsuche.

In der Sprache von Kapitel II haben wir hier die lineare Relation (T_1, T) für f ausgenutzt. Klar ist, dass man analog jede lineare Relation ausnutzen kann, um die Komplexität der vollständigen Schlüsselsuche zu reduzieren.

Beispiel: Geffe-Generator. Hier werden die Korrelationen durch die folgende Tabelle beschrieben:

x	0	0	0	0	1	1	1	1
t	0	0	1	1	0	0	1	1
y	0	1	0	1	0	1	0	1
$f(x,t,y)$	0	0	0	1	1	1	0	1

Als Wahrscheinlichkeit für die Übereinstimmung erhält man also

$$p = \begin{cases} \frac{3}{4} & \text{für das Register 1,} \\ \frac{1}{2} & \text{für das Register 2 (Steuerung),} \\ \frac{3}{4} & \text{für das Register 3.} \end{cases}$$

Daher lassen sich bei einer Korrelationsattacke die Startwerte für die Register 1 und 3 – die Batterieregister – leicht schon aus kurzen Outputfolgen bestimmen; den Startwert für Register 2, das Steuerungsregister, findet man dann auch leicht durch Exhaustion seiner Startvektoren.

Aus der bisherigen Diskussion lassen sich als Design-Kriterien für nichtlineare Kombinierer herleiten:

- Die einzelnen Batterieregister müssen möglichst lang sein; genaueres siehe unten.
- Die Kombinierfunktion f muss balanciert sein
- ... und soll eine möglichst hohe Nichtlinearität, d. h., ein möglichst geringes lineares Potenzial haben.

Zum letzten Punkt: Wegen der nötigen Balanciertheit sind krumme Funktionen trotz ihrer „Korrelationsimmunität“ nicht geeignet. Die nächst beste Eigenschaft ist „resilient“ = unter den balancierten maximal nichtlinear.

Wie lang sollen die Batterieregister sein? Es gibt verschiedene Ansätze zu „schnellen“ Korrelationsattacken, z. B. mit Hilfe der Walsh-Transformation, besonders gegen dünn besetzte Rückkopplungspolynome. Diese reduzieren zwar nicht die Komplexitätsklasse des Angriffs, aber der Aufwand wird um einen beträchtlichen Proportionalitätsfaktor verringert. Auf diese Weise werden Register angreifbar, die bis zu 100 Koeffizienten 1 im Rückkopplungspolynom haben. Folgerung:

- Die einzelnen linearen Schieberegister sollten mindestens 200 Bits lang sein und eine „dicht besetzte“ Rückkopplung besitzen.

Ein eleganter Ausweg, der die Korrelationsattacke zusammenbrechen lässt, wurde von Rueppel vorgeschlagen: eine zeitabhängige Kombinierfunktion, also eine Familie $(f_t)_{t\in\mathbb{N}}$ zu verwenden.

Als Fazit kann man festhalten: Mit linearen Schieberegistern und nichtlinearen Kombinierern lassen sich sich ziemlich effiziente Pseudozufallsgeneratoren aufbauen. Für deren kryptologische Sicherheit gibt es zwar keine umfassende befriedigende Theorie, aber durchaus eine Absicherung, die – ähnlich wie bei Bitblock-Chiffren – auf der Theorie der Nichtlinearität Boolescher Funktionen beruht.