Stream cipher cryptanalysis

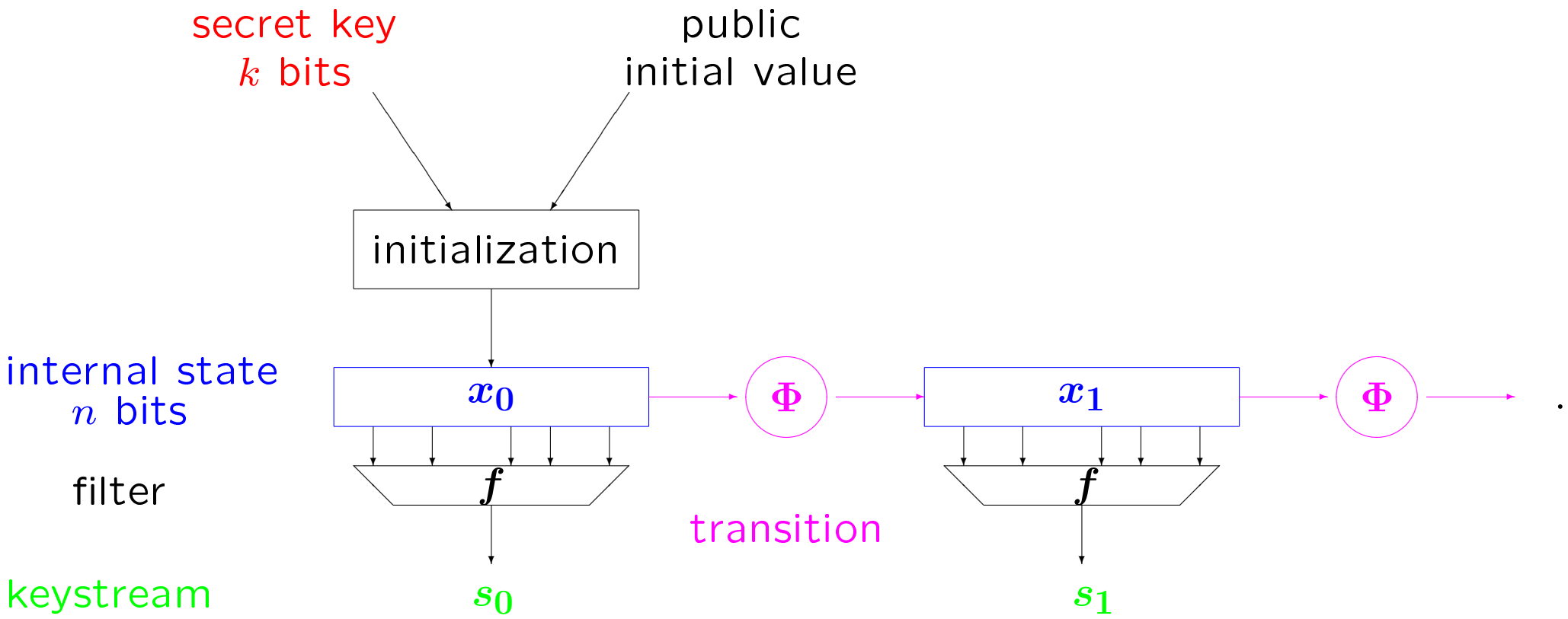
Anne Canteaut

Anne.Canteaut@inria.fr

http://www-rocq.inria.fr/secret/Anne.Canteaut/

Ice Break 2013

Keystream generator



Avoiding generic attacks

- The internal state must be at least twice larger than the key (Time-memory-data trade-off [Golic 95][Babbage 95])
- The generator must pass the statistical tests. In particular, the filtering function f must be balanced.
- At least one function among Φ and f must be nonlinear.
- Φ has no short cycles.

Choosing the transition function Φ

Two strategies:

- Choose a random-looking mapping/permutation operating on a large internal state: the period of $(x_t)_{t \geq 0}$ is expected to be close to $2^{\frac{n}{2}}$. Short cycles exist but are unlikely to occur. Eg: RC4.

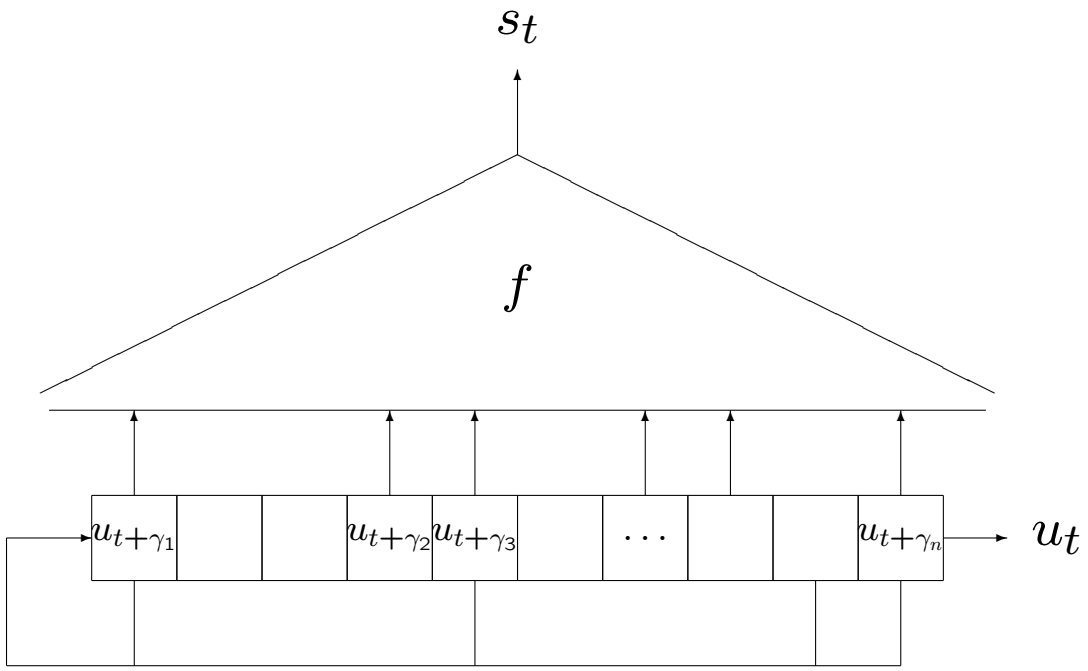
- Choose a permutation with some known mathematical properties operating on a small internal state: the period of $(x_t)_{t \geq 0}$ can be proved to be close to 2^n. Short cycles are avoided. Eg: LFSR (Tor's lecture).

Outline

- Statistical attacks against filtered LFSRs
- Algebraic attacks against filtered LFSRs
- Correlation attacks against combination generators

Statistical attacks against filtered LFSRs

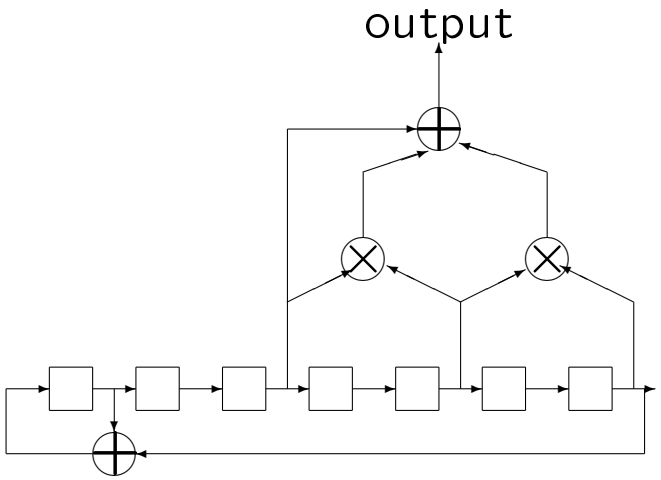
Principle



If f is balanced, then the output sequence contains roughly the same numbers of 0 and 1.

Question: What if we consider pairs of bits? Do we get a uniform distribution?

Example

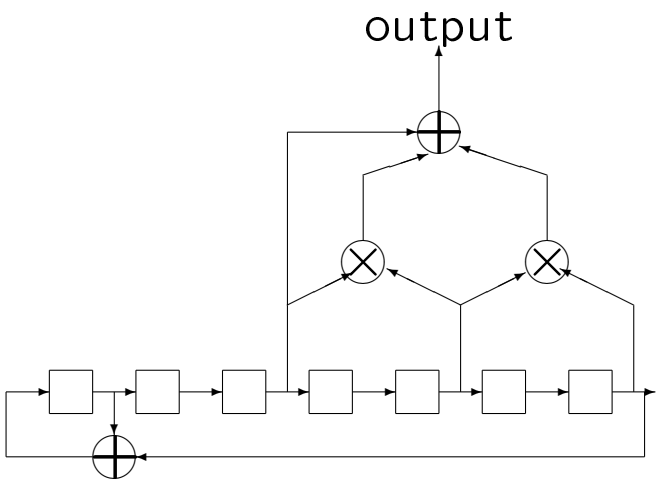


generates the $\mathbf{127}$-bit sequence

0100001100000100001111110111010001001110111011001100011001111
0110111001100110010010100110010111001111010100110110010001110
0110100

$\longrightarrow$ It has $\mathbf{64}$ ones.

Example

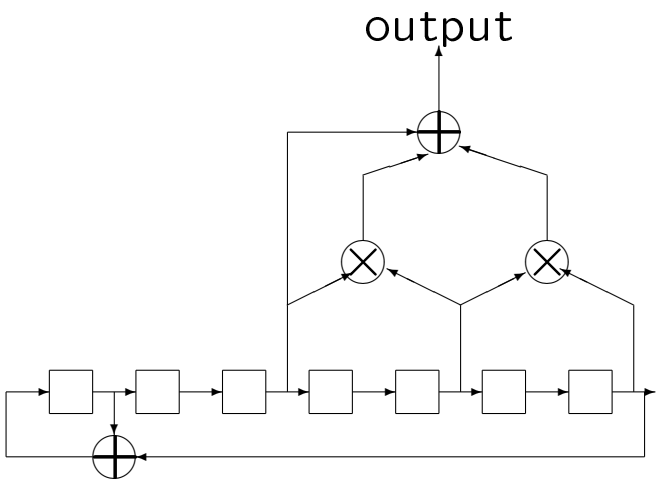


generates the $\mathbf{127}$-bit sequence

0100001100000100001111101110100010011101110110011000110011110110111001100100101001100101110011101010011011001000111000110100

$$(s_t, s_{t+1}) = \begin{cases} (0,0) & \mathbf{30} \text{ times} \\ (0,1) & \mathbf{32} \text{ times} \\ (1,0) & \mathbf{32} \text{ times} \\ (1,1) & \mathbf{32} \text{ times} \end{cases}$$

Example



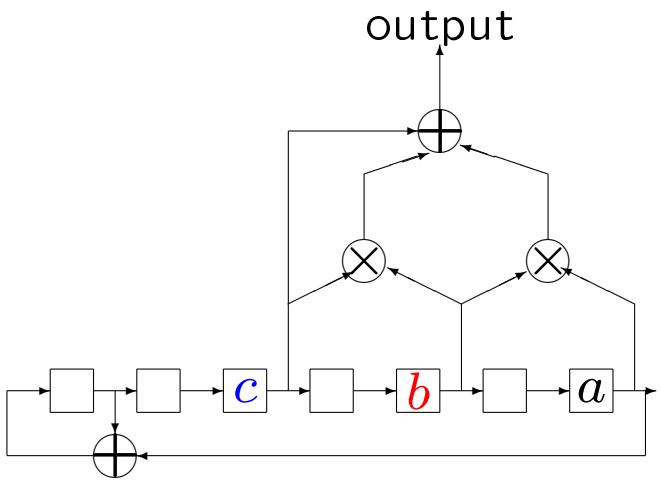
generates the $\mathbf{127}$-bit sequence

0100001100000100001111101110100010011101110110011000110011110110111001100100101001100101110011101010011011001000111000110100

$$(s_t, s_{t+2}) = \begin{cases} (0,0) & \mathbf{22} \text{ times} \\ (0,1) & \mathbf{40} \text{ times} \\ (1,0) & \mathbf{39} \text{ times} \\ (1,1) & \mathbf{24} \text{ times} \end{cases}$$

Example

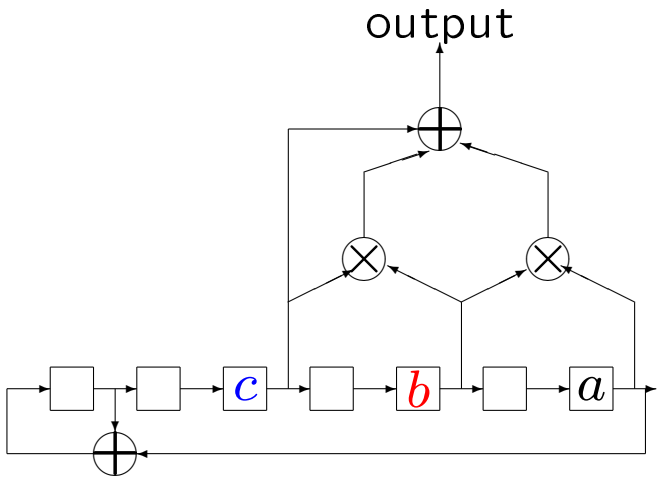
At time t:



$$s_t = f(c, b, a)$$

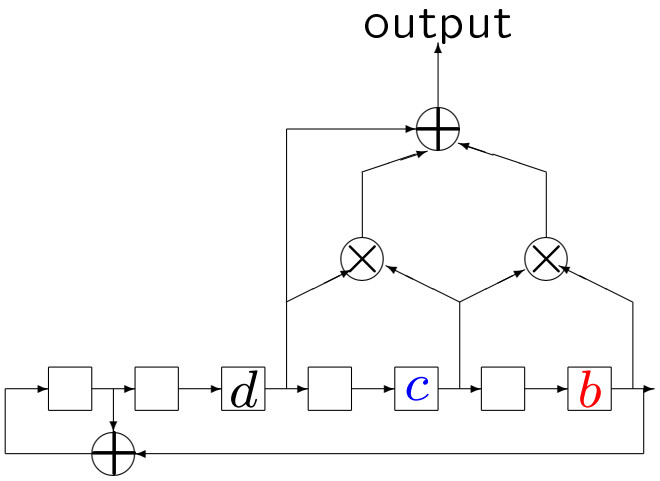
Example

At time t:



$$s_t = f(c, b, a)$$

At time $t + 2$:



$$s_{t+2} = f(d, c, b)$$

Example

The distribution of (s_t, s_{t+2}) is given by the truth table of the function

$$(a, b, c, d) \mapsto (f(c, b, a), f(d, c, b))$$

d	c	b	a	y_1	y_2
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	1

d	c	b	a	y_1	y_2
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	1	1	0

Example

The distribution of (s_t, s_{t+2}) is given by the truth table of the function

$$(a, b, c, d) \mapsto (f(c, b, a), f(d, c, b))$$

d	c	b	a	y_1	y_2
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	1

d	c	b	a	y_1	y_2
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	1	1

$$\rightarrow \Pr[s_t \neq s_{t+2}] = \frac{5}{8} > \frac{1}{2}$$

Related design criteria

Augmented function [Anderson 91]

$$\begin{array}{rcl} \{0,1\}^{\gamma_n-\gamma_1+m} & \longrightarrow & \{0,1\}^m \\ (u_{\gamma_n}, u_{\gamma_n+1}, \ldots, u_{\gamma_1+m-1}) & \longmapsto & \big(f(u_{\gamma_n}, \cdots, u_{\gamma_1}), f(u_{\gamma_n+1}, \cdots, u_{\gamma_1+1}), \ldots \\ & & f(u_{\gamma_n+m-1}, \cdots, u_{\gamma_1+m-1})\big) \end{array}$$

should be balanced for all $m \leq (\gamma_n - \gamma_1)$.

This condition holds for instance if f is linear with respect to its first or to its last input variable.

Choice of the selection cells.

The number of pairs (i, j) with the same $\gamma_j - \gamma_i$ should be as small as possible.

Algebraic attacks against filtered LFSRs

Recovering the initial state for a linear transition function

- The initial state is secret:

$$\text{initial state} = k_0, \ldots, k_{n-1} \in \{0,1\}^n$$

- Linear transition function on $\{0,1\}^n$:

$$x_t = L^t(k_0, \ldots, k_{n-1})$$

- Filtering function from $\{0,1\}^n$ into $\{0,1\}$:

$$s_t = f(x_t) = f\left[L^t(k_0, \ldots, k_{n-1})\right]$$

Problem. Recover the initial state $k_0, \ldots, k_{n-1}$ from the knowledge of N keystream bits $s_0, s_1, \ldots, s_{N-1}$.

Basic algebraic attack

Set up the enciphering equations:

$$\begin{cases} s_0 &= f(k_0, \ldots, k_{n-1}) \\ s_1 &= f \circ L(k_0, \ldots, k_{n-1}) \\ & \\ s_t &= f \circ L^t(k_0, \ldots, k_{n-1}) \end{cases}$$

System of equations with n variables of degree $d = \deg(f)$.

Basic algebraic attack

Set up the enciphering equations:

$$\begin{cases} s_0 &= f(k_0, \ldots, k_{n-1}) \\ s_1 &= f \circ L(k_0, \ldots, k_{n-1}) \\ & \\ s_t &= f \circ L^t(k_0, \ldots, k_{n-1}) \end{cases}$$

System of equations with n variables of degree $d = \deg(f)$.

- **The lazy cryptographer:** ask Martin!

Basic algebraic attack

Set up the enciphering equations:

$$\begin{cases} s_0 &= f(k_0, \ldots, k_{n-1}) \\ s_1 &= f \circ L(k_0, \ldots, k_{n-1}) \\ & \\ s_t &= f \circ L^t(k_0, \ldots, k_{n-1}) \end{cases}$$

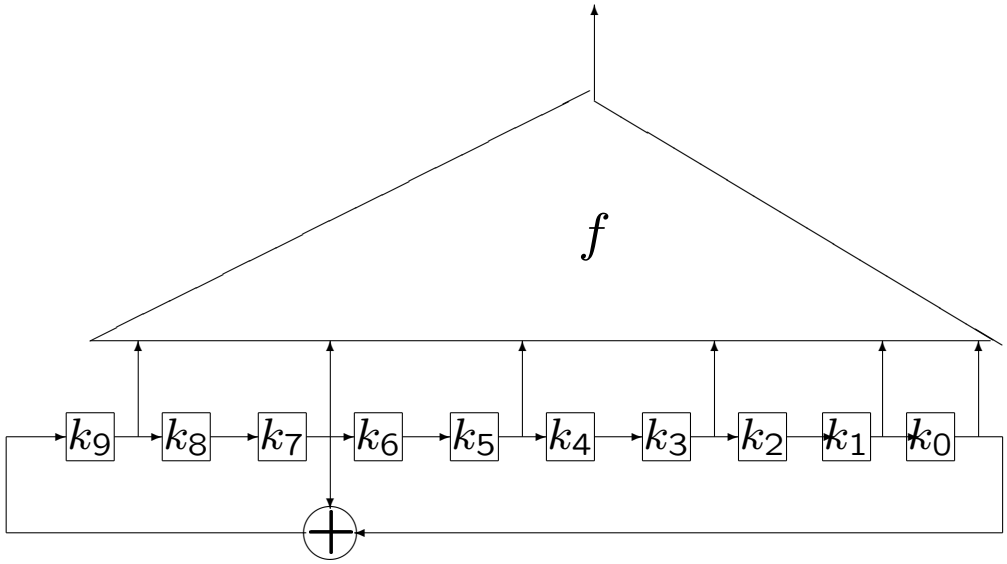
System of equations with n variables of degree $d = \deg(f)$.

- **The stupid cryptographer:** Solve the system by linearization

$$\sum_{i=1}^{d} \binom{n}{i} \simeq \frac{n^d}{d!} \text{ keystream bits}$$

Time complexity: n^{3d} operations .

Example



where $f = x_1 + x_6 + x_2x_4 + x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_3x_4x_5$.

We get a system of degree 5 with 10 unknowns.

Algebraic attacks [Courtois-Meier 03]

Let $AN(f) = \{g,\ g(x)f(x) = 0 \text{ for all } x \in \{0,1\}^n\}$
be the set of annihilators of f.

Let $g \in AN(f)$, i.e., such that $g(x)f(x) = 0$ for all $x \in \{0,1\}^n$.

$$g(x_t)f(x_t) = g(x_t)s_t = 0$$

$$\Longrightarrow g \circ L^t(k_0, \ldots, k_{n-1}) = 0 \text{ if } s_t = 1\ .$$

Algebraic attacks [Courtois-Meier 03]

Let $AN(f) = \{g,\ g(x)f(x) = 0 \text{ for all } x \in \{0,1\}^n\}$
be the set of annihilators of f.

Let $g \in AN(f)$, i.e., such that $g(x)f(x) = 0$ for all $x \in \{0,1\}^n$.

$$g(x_t)f(x_t) = g(x_t)s_t = 0$$

$$\Longrightarrow g \circ L^t(k_0, \ldots, k_{n-1}) = 0 \text{ if } s_t = 1 \ .$$

Let $h \in AN(1+f)$, i.e, such that $h(x)(1+f(x)) = 0$ for all $x \in \{0,1\}^n$.

$$h(x_t)(1+f(x_t)) = h(x_t)(1+s_t) = 0$$

$$\Longrightarrow h \circ L^t(k_0, \ldots, k_{n-1}) = 0 \text{ if } s_t = 0 \ .$$

Algebraic system with n variables of degree

$$d = \min\{\deg(g), g \in AN(f) \cup AN(1+f), g \neq 0\} \ .$$

Example

$$f = x_1 + x_6 + x_2x_4 + x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_3x_4x_5$$

Value vector of f:

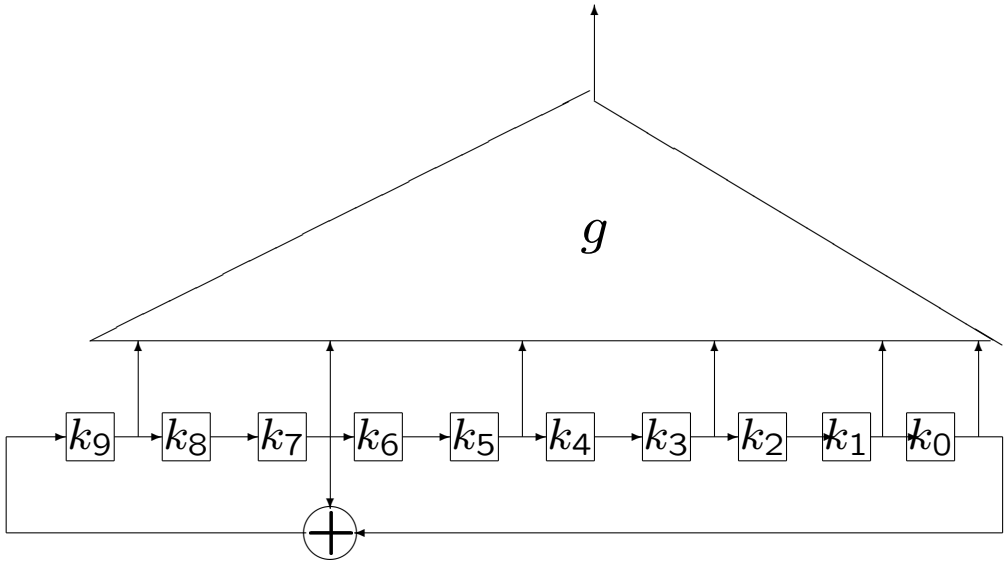
01010101011001100101010110011011101010101001100110101010011001001010101000000000101010100000000010101010000000000101010100000000

$\rightarrow$ The function g with the previous value vector belongs to $AN(f)$.

$$g = 1 + x_1 + x_4 + x_1x_4 + x_6 + x_4x_6$$

In particular, $\deg g = 2$.

Example



If $s_t = 1$, then the output of this generator is 0. Suppose that the keystream is $10110\ldots$.

$$
\begin{aligned}
0 &= 1 + k_9 + k_3 + k_3k_9 + k_0 + k_0k_3 \\
0 &= 1 + k_{11} + k_5k_{11} + k_2 + k_2k_5 \\
&= 1 + k_1 + k_8 + k_1k_5 + k_5k_8 + k_2 + k_2k_5 \\
0 &= 1 + k_2 + k_9 + k_2k_6 + k_6k_9 + k_3 + k_3k_6
\end{aligned}
$$

Similar complexity as the attack for a filtering function of degree 2.

Complexity of the attack

n = size of the internal state

$AI(f)$ = algebraic immunity of the filtering function f

$$AI(f) = \min\{\deg(g),\ g \in AN(f) \cup AN(1+f),\ g \neq 0\}.$$

Number of operations:

$$\left(\sum_{i=1}^{AI(f)} \binom{n}{i} \right)^3 \simeq n^{3AI(f)}\ .$$

Security criterion

$AI(f) =$ algebraic immunity of the filtering function f

$AI(f) = \min\{\deg(g),\ g \in AN(f) \cup AN(1+f),\ g \neq 0\}.$

Comparison with exhaustive search:

If $n = 2k$ where k is the key size, we must have $(2k)^{3AI(f)} \geq 2^k$ i.e.,

$$AI(f) \geq \frac{k}{3(1+\log_2 k)}$$

Example. $k = 128$ bits, $n = 256$ bits.

$$\longrightarrow AI(f) \geq 7\ .$$

Finding annihilators

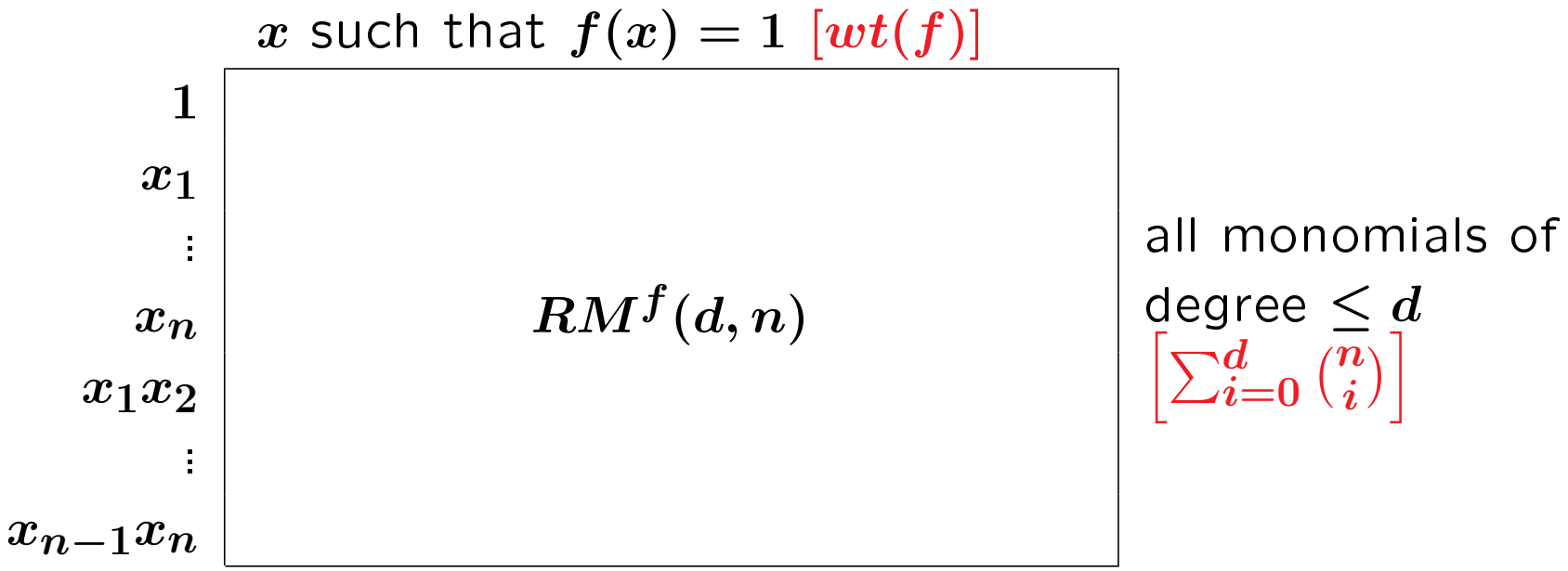
Expensive technique:

Compute the ANF of all functions g such that $g(x) = 0$ for all x such that $f(x) = 1$.

0101010101100110010101011001101110101010100110011010101001100100
?0?0?0?0?00??00??0?0?0?00??00?000?0?0?0?0??00??00?0?0?0??00??0??

If f is a balanced function of n variables, 2^{n-1} different annihilators should be examined.

Finding annihilators



Proposition. There exists $g \neq 0$ in $AN(f)$ with $\deg g \leq d$ when

$$wt(f) < \sum_{i=0}^{d} \binom{n}{i} .$$

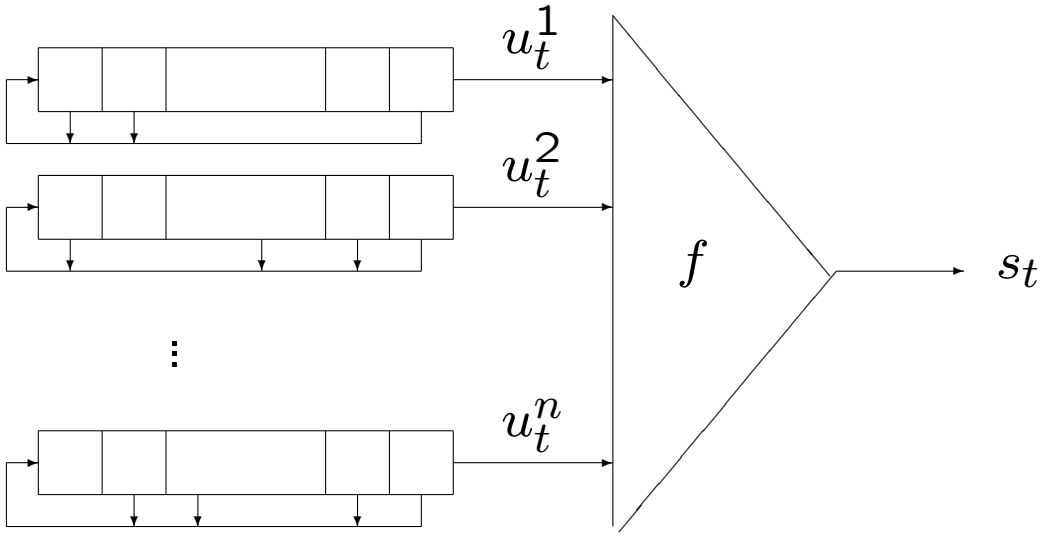
Corollary. For any f of n variables,

$$AI(f) \leq \left\lceil \frac{n}{2} \right\rceil .$$

Correlation attacks

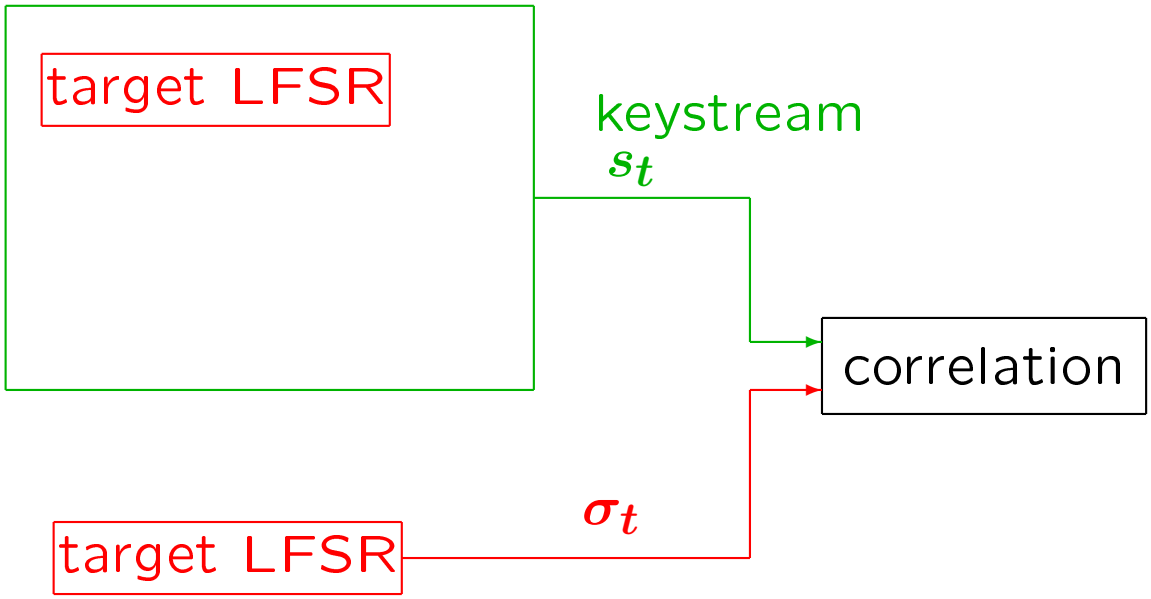
against combination generators

The combination generator



$$s_t = f(u_t^1, u_t^2, \ldots, u_t^n)$$

Correlation attack [Siegenthaler 85]



where $p = \mathbf{Pr}[s_t = \sigma_t] \neq \frac{1}{2}$.

Divide-and-conquer attack:

Recover the initial state of the target LFSR from the knowledge of some keystream bits.

Basic correlation attack [Siegenthaler 85]

Algorithm:

For each possible initialization of the target LFSR:

Compute the sequence σ generated by the target LFSR

Check whether σ is correlated to the keystream s

(or to its bitwise complement) by computing

$$C = \sum_{t=0}^{N-1} (-1)^{s_t + \sigma_t}$$

Example: Geffe generator

$x_1x_2x_3$	000	100	010	110	001	101	011	111
$f(x_1, x_2, x_3)$	0	1	0	0	0	1	1	1
$f(x_1, x_2, x_3) = x_1$?	x	x	x		x	x		x

$$p = \Pr[s_t = \sigma_t] = \Pr[f(x_1, x_2, x_3) = x_1] = \frac{1}{2}\left(1 + \frac{\mathcal{E}(f + x_1)}{2^3}\right) = \frac{3}{4}\,.$$

→Target LFSR: LFSR 1

Statistical test

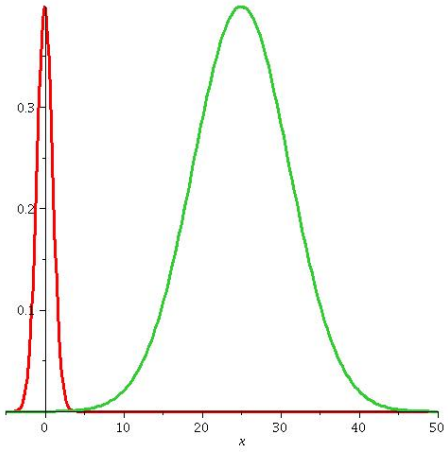
- If σ is independent from s,

$$\mathbf{Pr}[C = x] \simeq \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

- If σ is the output of the target LFSR,

$$\mathbf{Pr}[C = x] \simeq \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x - N(2p-1))^2}{8Np(1-p)}\right)$$

For $p = \frac{3}{4}$ and $N = 50$



Complexity

Data complexity:

For a target LFSR of length L, we choose a false alarm probability $\alpha \simeq 2^{-L}$. Then, we need

$$N \simeq \frac{L}{\left(p - \frac{1}{2}\right)^2}$$

Time complexity:

$$2^L N \simeq \frac{L 2^L}{\left(p - \frac{1}{2}\right)^2}$$

Correlation attacks with a Fast Fourier Transform

Each bit produced by the target LFSR can be written as a linear combination of the bits of the initial state k:

$$\sigma_t = \bigoplus_{i=0}^{L-1} \alpha_{t,i} k_i = \alpha_t \cdot k \ .$$

For the LFSR of length 4 with feedback polynomial $1 + X + X^4$:

$$\begin{aligned}
\sigma_4 &= k_3 \oplus k_0 && \rightarrow \alpha_4 = (1,0,0,1) \\
\sigma_5 &= k_3 \oplus k_1 \oplus k_0 && \rightarrow \alpha_5 = (1,1,0,1) \\
\sigma_6 &= k_3 \oplus k_2 \oplus k_1 \oplus k_0 && \rightarrow \alpha_6 = (1,1,1,1) \\
\sigma_7 &= k_2 \oplus k_1 \oplus k_0 && \rightarrow \alpha_7 = (1,1,1,0)
\end{aligned}$$

We need to compute the 2^L correlations

$$C(k) = \sum_{t=0}^{N-1} (-1)^{s_t + \sigma_t} = \sum_{t=0}^{N-1} (-1)^{s_t} (-1)^{\alpha_t \cdot k} \ .$$

Correlation attacks with a Fast Fourier Transform

$$C(k) = \sum_{t=0}^{N-1} (-1)^{s_t + \sigma_t} = \sum_{t=0}^{N-1} (-1)^{s_t} (-1)^{\alpha_t \cdot k} \ .$$

Discrete Fourier transform:

$$\begin{array}{ccc} \{0,1\}^L & \longrightarrow & \mathbb{Z} \\ k & \longmapsto & \widehat{F}(k) = \sum_{x \in \{0,1\}^L} F(x)(-1)^{x \cdot k} \end{array}$$

Here,

$$C(k) = \widehat{F}(k)$$

with $F(x) = \begin{cases} (-1)^{s_t} & \text{if } x = \alpha_t \\ 0 & \text{if there is no such } t \in \{0, \ldots, N-1\} \end{cases}$

Correlation attacks with a Fast Fourier Transform

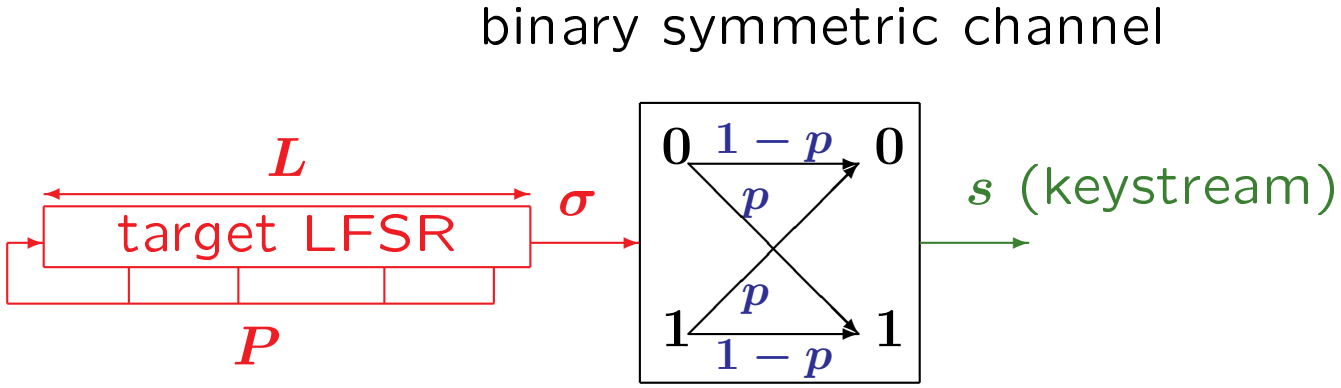
$\alpha_0 = 1000, \alpha_1 = 0100, \alpha_2 = 0010, \alpha_3 = 0001, \alpha_4 = 1001,$
$\alpha_5 = 1101, \alpha_6 = 1111, \alpha_7 = 1110$

Observed keystream: $s_0 \dots s_7 = 00101100$

x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
t		0	1		2			7	3	4		5				6
F	0	1	1	0	-1	0	0	1	1	-1	0	-1	0	0	0	1
	1	0	1	-1	-1	0	0	2	-1	2	1	1	-1	0	0	0
	0	0	1	1	2	0	1	-3	-2	2	1	1	0	2	1	1
	1	1	-1	-1	3	-3	1	3	-1	3	-3	1	1	3	-1	1
$\widehat{F}$	2	0	-2	0	0	6	4	-2	2	-4	-2	-4	4	-2	0	-2

$\longrightarrow$ Time complexity $L2^L$ instead of $N2^L$.

Correlation attack as a decoding problem [Meier-Staffelbach 88]



Error probability:

$$p = Pr[s_t \neq \sigma_t] < \frac{1}{2}$$

$(\sigma_t)_{t<N}$ belongs to the $[N, L]$-linear code defined by P.

LFSR code

Linear code of length N and dimension L :

$$(\sigma_0, \ldots, \sigma_{L-1}) \begin{bmatrix} 1 & 0 & \ldots & 0 & c_L & \ldots & g_{0,t} & \ldots \\ 0 & 1 & \ldots & 0 & c_{L-1} & \ldots & g_{1,t} & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & c_1 & \ldots & g_{L-1,t} & \ldots \end{bmatrix} = (\sigma_0, \ldots \ldots \sigma_{N-1})$$

where the t-th column is given by

$$\sum_{i=0}^{L-1} g_{i,t} X^i = X^t \bmod P^\star(X) \text{ with } P^\star(X) = \sum_{i=0}^{L} c_{L-i} X^i \ .$$

Proposed decoding techniques

- ML-decoding of the original LFSR-code [Siegenthaler 85];
- Convolutionnal code and Viterbi algorithm
  [Johansson-Jönsson 99];
- Turbo-code [Johansson-Jönsson 99];
- Low-Density Parity-Check codes and iterative decoding
  [Meier-Staffelbach 88], [Canteaut-Trabbia 00],
  [Mihaljevic - Fossorier- Imai 00];
- ML-decoding of a derived code with smaller dimension
  [Chepyshov - Johansson - Smeets 00];
- Sudan's algorithm for reconstructing a linear polynomial
  [Johansson-Jönsson 00].

Correlation-immune combining function

Security criterion:

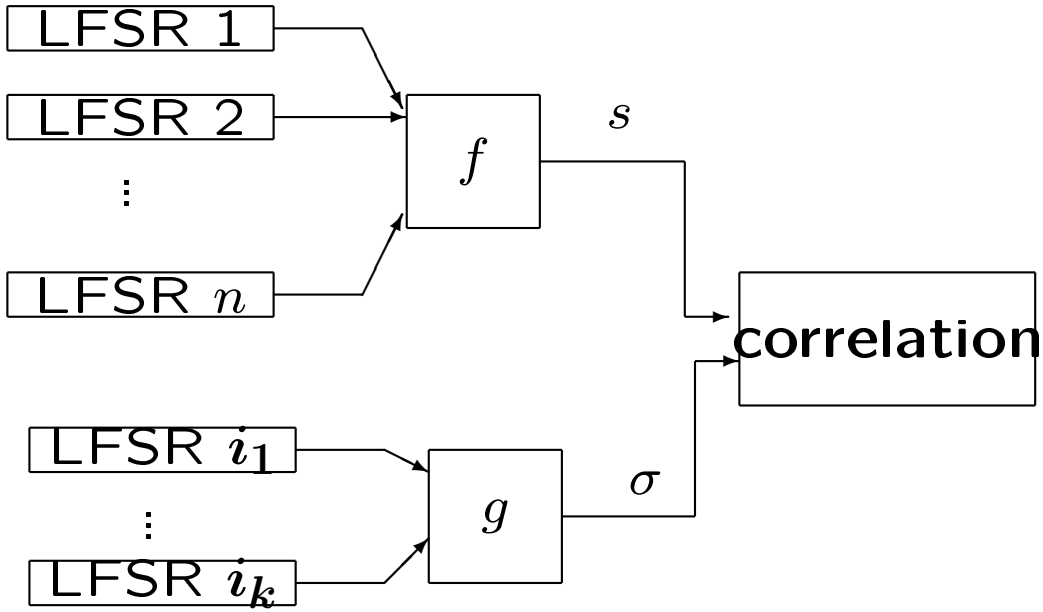
for each input variable x_i, $1 \leq i \leq n$,

$$p = \Pr[f(x_1, \ldots, x_n) = x_i] = \frac{1}{2} .$$

f is said to be correlation-immune.

Equivalently, $\mathcal{E}(f + \varphi_a) = 0$ for all a with $wt(a) = 1$.

Correlation attack involving several LFSRs:



Grain v1 [Hell Johansson Meier]