FSE 2011, February 14 -16, Lyngby, Denmark

Fast correlation attacks on certain stream ciphers

Willi Meier

FHNW

Switzerland

Overview

- A decoding problem
- LFSR-based stream ciphers
- Correlation attacks
- Fast correlation attacks
- Towards correlation immunity
- Combiners with memory
- Linear attacks (correlations everywhere?)
- Conclusions

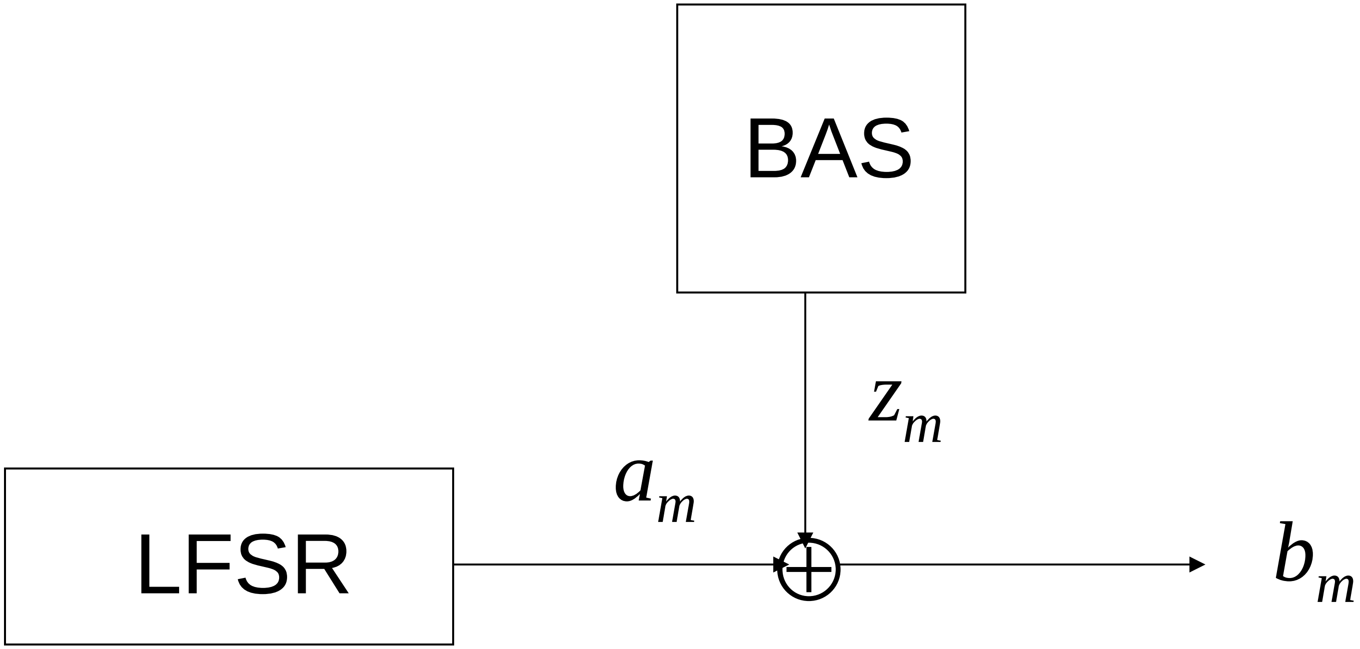
A decoding problem

Given: A noisy version of the output sequence of length L of a LFSR with known length n and known feedback connection.

Problem: Find the initial state of the LFSR

Solution: Decoding of a linear $[n,L]$-code.

Statistical Model:



BAS: Binary asymmetric source,

$Prob(z_m = 0) = p > 0.5$

For given L digits of $\underline{b}$ and structure of the LFSR of length n:

Find correct output sequence $\underline{a}$ of LFSR

Known solution: By exhaustive search over all initial states of LFSR find $\underline{a}$ such that

$$T = \# \{ j \mid b_j = a_j, 1 \le j \le L \}$$

is maximum. Complexity: $O(2^n)$
Feasible for n up to about 50.

Efficient solution of this problem of interest in:

- Satellite communications
- Correlation attacks on LFSR-based stream ciphers
- TCHo: An efficient trapdoor stream cipher (M. Finiasz, S. Vaudenay, 2006)
- Digital watermarking (D. Wang. P. Lu, 2006)
- ε-Biased Generators in NC^0 (E. Mossel, A. Shpilka, L. Trevisan, 2007)

LFSR-based stream ciphers

Output sequences of linear feedback shift registers (LFSR's):

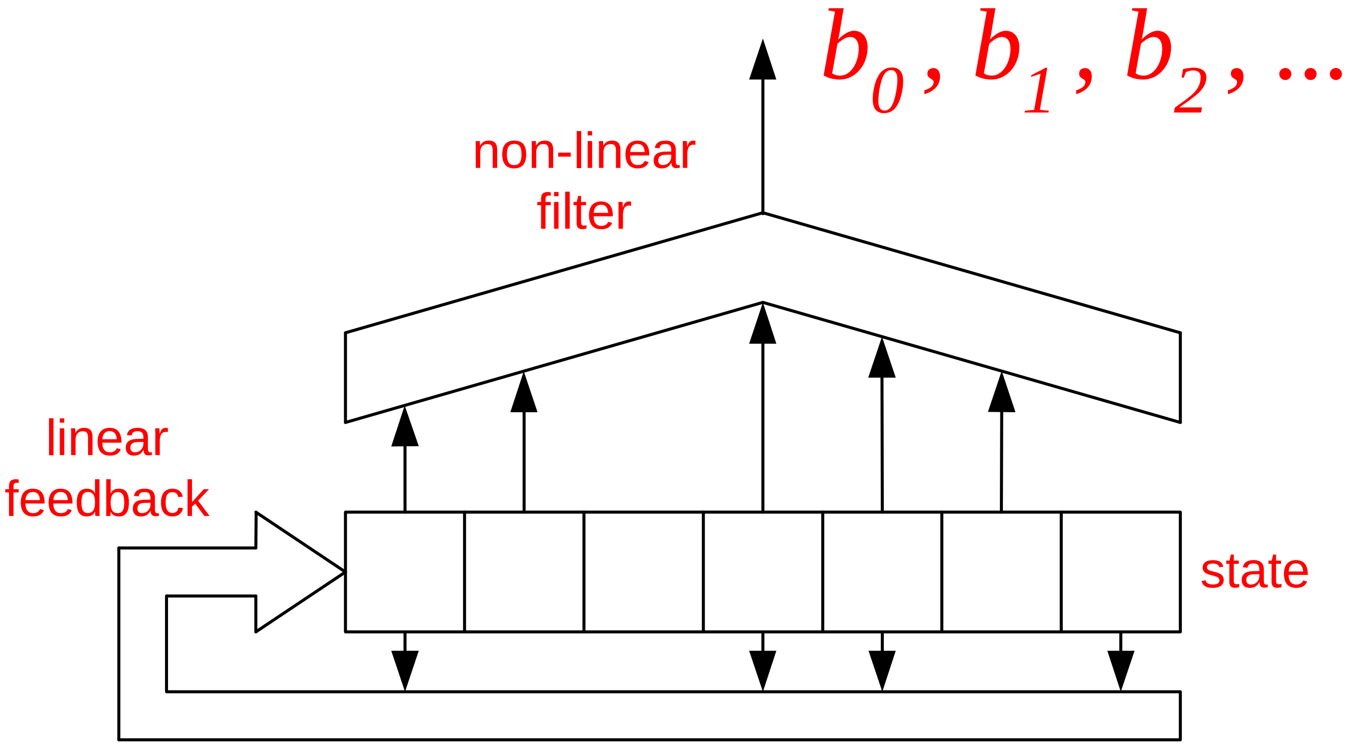
Have desirable statistical properties and large period.

Readily analyzable using algebraic techniques, via feedback polynomial.

For cryptographic properties, their linearity has to be destroyed.

Nonlinear filter generator

Generate keystream bits b_0, b_1, b_2 ,..., as some nonlinear function f of the stages of a single LFSR.



Nonlinear combiner generator

The outputs a_{m} of s LFSR‘s are used as input of a Boolean function f to produce keystream bits b_{m} , for $m = 1, 2, \ldots,$

$$f(a_{1\mathrm{m}},\ldots,a_{s\mathrm{m}}) = b_{\mathrm{m}}$$

Correlation Attacks

Case of combination generator:

Boolean function f produces keystream bits b_{m} , for $m = 1, 2, \ldots,$

$$f(a_{1\mathrm{m}},\ldots,a_{s\mathrm{m}}) = b_{\mathrm{m}}$$

Suppose there exist correlations,

$Prob(b_{\mathrm{m}} = a_{\mathrm{im}}) = p$, $p \neq 0.5$, to the output a_{im} of the i-th LFSR.

Example

$s = 3$ inputs

combination function f: majority function

$$f(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 = \mathrm{y}$$

$p(y = x_\mathrm{i}) = 0.75$ for $i = 1, 2, 3$.

Assume:
There exists correlation of the output sequence *b* of a stream cipher to one or several of its component LFSR-sequences *a*:

Use decoding technique to determine state of the LFSR in divide-and-conquer manner.

Correlation attacks first considered systematically by Thomas Siegenthaler (1984).

Fast correlation attacks

Fast correlation attack: Significantly faster than exhaustive search over all initial states of target LFSR.

Based on using certain parity check equations created from feedback polynomial of LFSR.

Joint work with Othmar Staffelbach (1988).

Two phases

- Search for suitable parity check equations
- Equations are used in fast decoding algorithm to recover initial state of LFSR.

Algorithms most efficient if feedback connection has only few taps.

Closely related: Linear syndrome decoding, has been applied for fast correlation attacks (Zheng-Yang, Crypto 1988)

Algorithm description:

Example: $n = 3$. Recursion: $x_j = x_{j-1} + x_{j-3}$ mod 2

Squaring: Recursion $x_j = x_{j-2} + x_{j-6}$ mod 2 does also hold.

$$a_{j-3} + a_{j-1} + a_j = 0$$
$$a_{j-2} + a_j + a_{j+1} = 0$$
$$a_j + a_{j+2} + a_{j+3} = 0$$

A fixed digit a_j of the LFSR sequence $\underline{a}$ satisfies a certain number m of linear relations (involving a fixed number t of other digits of $\underline{a}$), obtained by shifting and iterated squaring of LFSR-relation.

Substitute the digits of the known output sequence $\underline{b}$ in these linear relations.

Some relations will hold; some others not.

Observation:

The more relations are satisfied for a digit b_j, the higher is the (conditional) probability that $b_j = a_j$

Compute probability p^* for $b_j = a_j$, conditioned on the number of relations satisfied.

Digit contained in one relation:

Assume a fixed digit $a^{(0)} = a_{\mathrm{j}}$ satisfies a linear relation involving t other digits of the LFSR-sequence $\underline{a}$,

$$a^{(0)} + a^{(1)} + a^{(2)} + \ldots + a^{(t)} = 0$$

Denote by $b^{(0)}, b^{(1)}, b^{(2)}, \ldots, b^{(t)}$ the digits in same positions of the perturbed sequence

$$b^{(0)} = a^{(0)} + z^{(0)}$$

$$b^{(1)} = a^{(1)} + z^{(1)}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$b^{(\mathrm{t})} = a^{(\mathrm{t})} + z^{(\mathrm{t})}$$

$$Prob(z^{(0)} = 0) = \ldots = Prob(z^{(t)} = 0) = p$$

$$s = Prob(\ z^{(1)} + \ldots + z^{(t)} = 0) : s = s(p,t)$$

$$s(p,t) = p * s(p,t\text{-}1) + (1\text{-}\ p)(1\text{-}\ s(p,t\text{-}1))$$

$$s(p,1) = p$$

Digit contained in several relations:

Assume that a fixed digit $a = a_j$ is contained in m relations each involving t other digits.

For a subset S of relations denote by $E(S)$ the event that exactly the relations in S (and no other relations) are satisfied:

$$Prob((b=a) \text{ and } E(S)) = ps^h(1 - s)^{m-h}$$

$$Prob((b \mathrel{!=} a) \text{ and } E(S)) = (1 - p)s^{m-h}(1 - s)^h$$

where $h = |S|$ denotes the number of relations in S.

New probability $p^* = Prob(b = a \mid E(S))$:

$$p^* = \frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h} + (1-p)s^{m-h}(1-s)^h}$$

Probability distributions for number of relations satisfied: **Binomial distributions**

Correct digits: $b = a$

$$p_1(h) = \binom{m}{h} s^h (1-s)^{m-h}$$

Incorrect digits: $b \mathrel{!}= a$

$$p_0(h) = \binom{m}{h} s^{m-h}(1-s)^h$$

Average number m of relations available:

$$m = \log_2\left(\frac{L}{2n}\right)(t+1)$$

Example: $p = 0.75$, $t = 2$, LFSR-length $n = 100$, $L = 5000$ output bits of $\underline{b}$.

Then $m = 12$ (in the average), and $s = 0.75^2 + 0.25^2 = 0.625$.

Example (cont.) Value of p^*, if h relations are satisfied:

h	p^*
12	0.9993
11	0.9980
10	0.9944

Two algorithms, Algorithms A and B, for „fast correlation attacks“ (Eurocrypt 1988 and J. Cryptology, 1989). Much faster than exhaustive search, even for long LFSR‘s (n=1000 or longer). Only efficient for low weight recursions ($t < 10$).

Algorithm A

Take the digits of $\underline{b}$ with highest (conditional) probability $p*$ as a guess of the sequence $\underline{a}$ at the corresponding positions.

Approximately n digits are required to find $\underline{a}$ by solving linear equations.

Computational complexity: $O(2^{cn}), 0 < c < 1$, i.e., complexity is exponential. c is a function of p, t and N/n.

Example: $c = 0.012$ if $t = 2$, $p = 0.75$ and $L/n = 100$.

Algorithm B

1. Assign the correlation probability p to every digit of $\underline{b}$
2. To every digit of $\underline{b}$ assign the new probability p^* . Iterate this step a number of times.
3. Complement those digits of $\underline{b}$ with $p^* < p_{\text{thr}}$ (suitable threshold).
4. Stop, if $\underline{b}$ satisfies the basic relation of the LFSR, else go to 1.

The number of iterations in 2. and the probability threshold in 3. have to be adequately chosen to obtain maximum correction effect.

Algorithm B

is essentially linear in the LFSR-length n

Successful only if $t < 10$.

Previous work: R. G. Gallager, Low-Density Parity Check Codes (1962).

Problem: Fast correlation attacks for arbitrary linear relations, i.e., for arbitrary t ?

First approach: Polynomial multiples

If recursion not of low weight: consider multiples of feedback polynomial that have low weight.

Apply correlation attack to linear recursion of sparse polynomial multiple.

Low weight multiples of feedback polynomial of more general interest:

Useful in numerous distinguishing attacks on LFSR-based stream ciphers.

Enumeration of multiples of primitive polynomials over GF(2) (S. Maitra, K. Gupta A. Venkateswarlu, 2005)

Example

Connection polynomial $g(x)$ over $GF(2)$ of degree 7 and of weight 5:

$$g(x) = x^7 + x^6 + x^4 + x + 1$$

has a polynomial multiple (a trinomial)

$$f(x) = g(x) \cdot m(x) = x^{21} + x^3 + 1$$

over $GF(2)$ with a polynomial $m(x)$ of degree 14.

Basic search for low weight multiples: Birthday paradox.

More elaborate methods, with different time/memory tradeoffs:

- Generalized birthdays (D. Wagner, 2002)

- Syndrome decoding

- Based on discrete logs in $GF(2^n)$ (W. Penzhorn, G. Kühn, 1995)

A feedback polynomial of LFSR of length *n* can have a polynomial multiple of weight 4 and length about $2^{n/3}$.

Decimation attack (Filiol, 2000)

Decimating output of LFSR by constant factor d can simulate characteristics of many other LFSR‘s, which are possibly shorter than original one.

Can apply correlation attack to such decimated sequence, as correlation probability is same as for original LFSR, but complexity of attack is lower for shorter LFSR.

Factor d depends on prime factorization of $2^n - 1$, and may be large.

Long list of results/contributions by …

- J. Golić
- D. MacKay
- A. Canteaut, M. Trabbia
- Ph. Hawkes, G. Rose
- V. Chepizhov, B. Smeets
- Th. Johansson, F. Jönsson
- M. Mihaljević, M. Fossorier, H. Imai
- P. Chose, A. Joux, M. Mitton

… (incomplete)

- Y. Edel, A. Klein

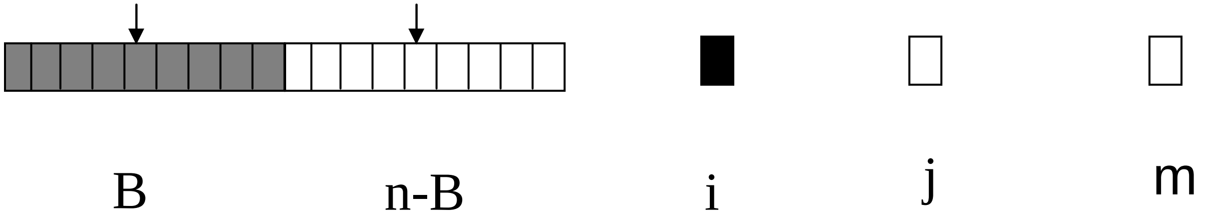
Culminates in achievement:

Fast correlation attacks are feasible for arbitrary linear relations and LFSR-length n up to about 100.

Fast correlation attack for LFSR of arbitrary weight:

- Call target bit a LFSR output bit to be predicted.
- Construct set of parity checks, involving k output bits.
- Evaluate estimators and conduct majority poll among them to recover initial state of LFSR.

Procedure is combined with partial exhaustive search for efficiency:
For a length n LFSR, B bits are guessed through exhaustive search, and n-B bits found using parity checks.



Parity check combines two bits *j* and *m* together with linear combination of guessed bits *B* in order to predict target bit *i*.

For better estimate, target more than n-B bits in the output.

Denote this number $D > n$.

For each of D target bits, evaluate large number of parity checks using noisy values b_t, and count number of parity checks that are satisified.

Number of parity checks satisfied: N_s
Number of parity checks not satisfied: N_u.

If difference $N_s - N_u$ is larger than threshold, predict $x_i = b_i$ if $N_s > N_u$, else $x_i = b_i + 1$.

If difference decisive for at least n-B of the D target bits, can easily recover initial state of LFSR.

Preprocessing: Parity checks found by:

- Collision search using Birthday paradox

- Match-and-sort algorithm

Fast correlation attacks on concrete ciphers:

LILI-128 (Jönsson-Johansson, 2002)

Grain-v0 (Berbain-Gilbert-Maximov, 2006)

Towards correlation immunity

Correlation attacks successful if cipher allows for good approximations of the output function by linear functions in state bits of LFSR‘s involved.

Impact of correlation attacks to design of stream ciphers:

Boolean functions f used should

- be correlation immune
- have high algebraic degree
- have large distance to affine functions

In view of correlation attacks, combining (or filter) function f should be carefully chosen, so that there is no statistical dependence between any small subset of inputs and the output.

Let X_1 X_2,..., X_n be independent binary variables, which are balanced (i.e. each takes values 0 or 1 with prob. ½).

A Boolean function $f(x_1, x_2,..., x_n)$ is m-th order correlation immune if for each subset of m random variables X_{i1}, X_{i2},.., X_{im}, the random variable $Z = f(X_{i1}, X_{i2},.., X_n)$ is statistically independent of the random vector $(X_{i1}, X_{i2},.., X_{im})$.

Tradeoff between correlation immunity and algebraic degree (Th. Siegenthaler, 1984).

Low algebraic degree of combining (or filter) function conflicts with security:

- Berlekamp-Massey LFSR-synthesis

- Algebraic Attacks

Study of Boolean functions with good cryptographic properties an ongoing topic.

Impact of cryptanalysis of DES block cipher (D. Chaum, J.-H. Evertse, 1986) to design of stream ciphers:

Alternative solution of correlation problem: Perfect nonlinear functions (1989, with O. Staffelbach).

Coincide with Bent functions (Rothaus 1976)
These functions are not exactly balanced.

Bent functions have:

Maximum nonlinearity, i.e., largest possible distance to all affine functions, and

Good correlation immunity properties.

Study of S-boxes with similar properties (K. Nyberg)

Important role of this type of S-boxes in design of AES block cipher, to counter differential and linear cryptanalysis

Tradeoff between correlation immunity and algebraic degree can be avoided if the combining function is allowed to have memory
(R. Rueppel, 1985).

Combiners with Memory

A (k,m)-combiner with k inputs and m memory bits is a finite state machine (FSM) which is defined by an output function

$$f : \{0,1\}^m \times \{0,1\}^k \rightarrow \{0,1\}$$

and a memory update function

$$\varphi : \{0,1\}^m \times \{0,1\}^k \rightarrow \{0,1\}^m$$

Given: A stream of (X_1, X_2,..) of inputs, X_i in $\{0,1\}^\mathrm{k}$, and initial assigment Q_1 in $\{0,1\}^\mathrm{m}$ to memory bits.

The output bitstream (z_1, z_2,..) is defined according to

$$z_t = f(Q_t, X_t),$$

and

$$Q_{t+1} = \varphi(Q_{t,} X_t)$$

for all t >0. For keystream generation, stream of inputs (X_1, X_2,..) is produced by output of k driving devices. Initial states determined by secret key. Often, driving devices are LFSR‘s.

Example of combiner with memory: Summation generator with k = 2 inputs.

Write X_{t} as $X_{\mathrm{t}} = (a_{\mathrm{t}}, b_{\mathrm{t}})$. The number of memory bits is m = 1 (i.e. the usual carry of addition of integers in binary representation).

The functions are defined by

$$z_t = f(Q_t, a_t, b_t) = a_t \oplus b_t \oplus Q_t$$

$$Q_{t+1} = \varphi(Q_t, a_t, b_t) = a_t b_t \oplus a_t Q_t \oplus b_t Q_t$$

The function f in this summation generator is 2nd – order correlation immune.

Example: E0 stream cipher used in Bluetooth is combiner with k = 4 inputs and m = 4 bit memory.

Stream of inputs is produced by outputs of 4 LFSR‘s of length 128 in total.

More recent (word-oriented) stream ciphers with memory:

SNOW, SOSEMANUK, ZUC.

Different development related to combiners with memory, leading to cryptanalysis of summation generator:

Feedback with carry shift registers (FCSR‘s)
(A. Klapper, M. Goresky, 1994).

Linear Attacks

Recall:

Correlation attack successful, if linear relations hold with nonnegligible probabilities, between single output bits and a subset of state bits of driving LFSR‘s.

Linear attack: Successful if there are correlations between linear functions of several output bits and linear functions of a subset of the LFSR-bits.

Linear attack more general than correlation attack.

If there are such correlations, get a linear system of equations, each of which does hold with some probability.

Linear system can be solved by methods reminiscent to fast correlation attacks (Golić).

Methods efficient if known keystream is long enough, i.e., if many more equations are available than number of unknowns.

Consider block of M consecutive inputs.

Outputs $Z_t = (z_t, z_{t-1},..., z_{t-M+1})$ as a function of the corresponding block of M consecutive inputs
$X_t=(x_t, x_{t-1},...,x_{t-M+1})$ and
the preceeding memory bits C_{t-M+1}.

X_t denotes bit vector at time t of state bits of driving LFSR‘s, and C_{t-M+1} bit vector of m memory bits at time t-$M+1$.

Assume that X_t and C_{t-M+1} are balanced and mutually independent.

Then, if $M > m$, there must exist linear correlations between the output and input bits, but they may also exist if $M <= m$ (Golić, 1996).

Linear attack on Bluetooth stream cipher E0 (Golić-Bagini-Morgari, 2002).

Exploits a variety of correlations between inputs and outputs. Cryptanalysis inspired by fast correlation attacks.

Linear Correlations unavoidable: Correlations everywhere?

Open Problems

- Optimal algorithmics of fast correlation attacks?
- Fast correlation attacks over extension fields?
- Construction of secure stream ciphers that provably have low correlations ("cryptographic ε-biased sequences")?