A survey of algebraic attacks against stream ciphers

Frederik Armknecht

NEC Europe Ltd. Network Laboratories
frederik.armknecht@netlab.nec.de

Special semester on Gröbner bases and related methods,
May 4th, 2006, Linz, Austria

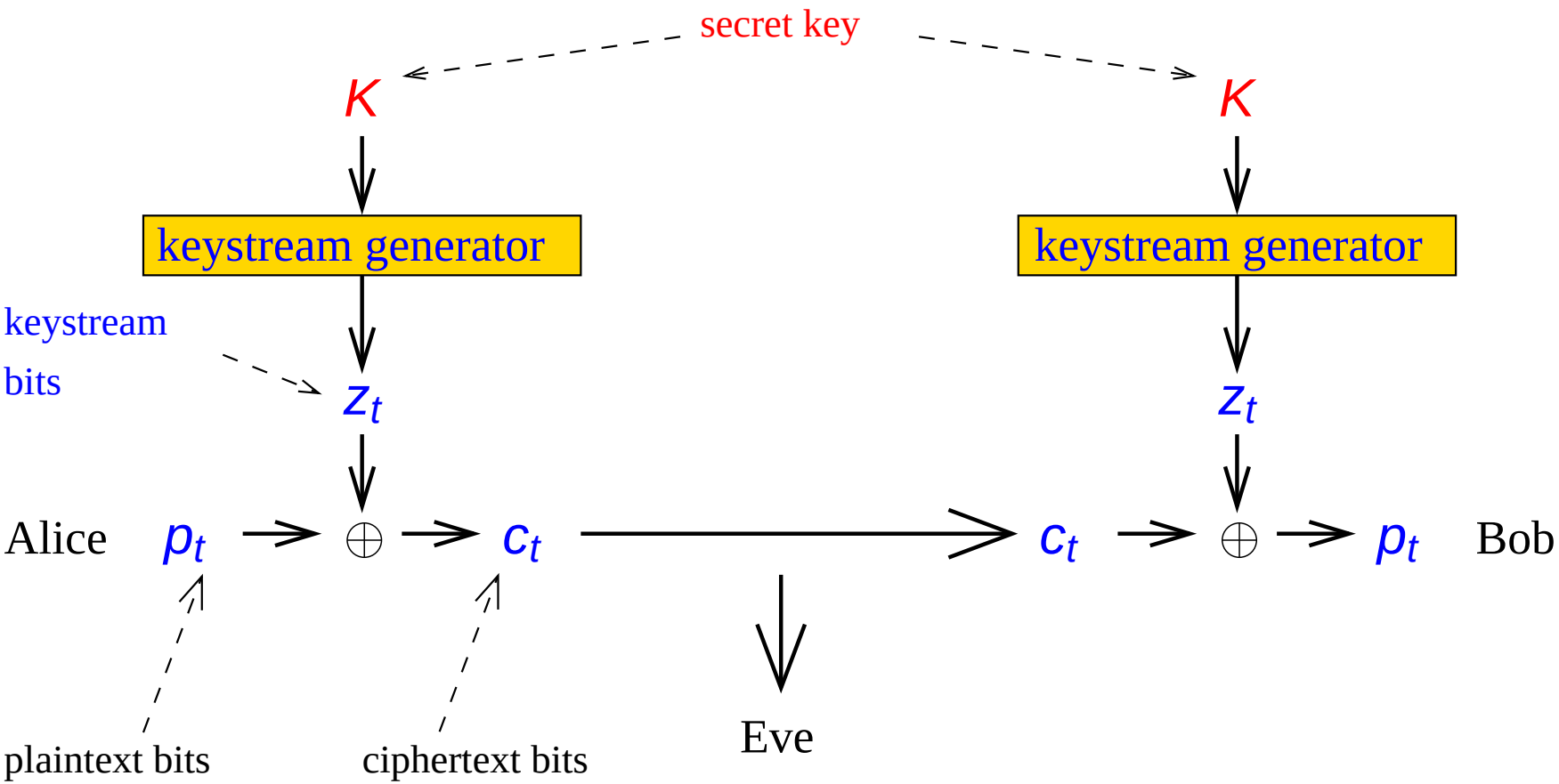
Outline

Outline

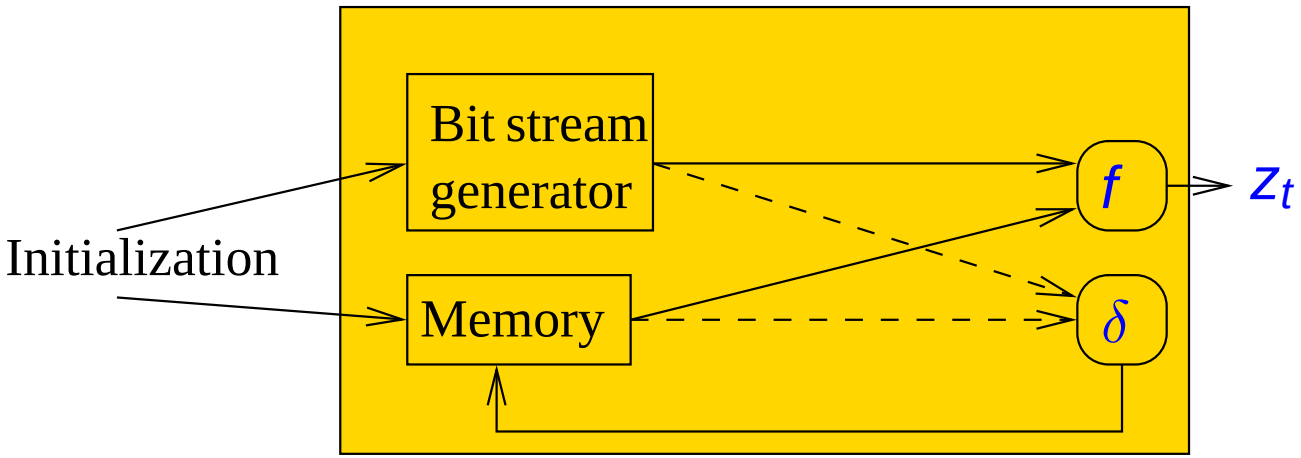
Stream ciphers

- Encrypt data of arbitrary length
- High speed
- Low hardware complexity
- Mostly based on keystream generators

Encryption with keystream generator



Combiners with memory



Bit stream generator should . . .

- . . . be fast
- . . . produce bit stream which ”looks random”

Often used in practice: Linear feedback shift registers (LFSR)

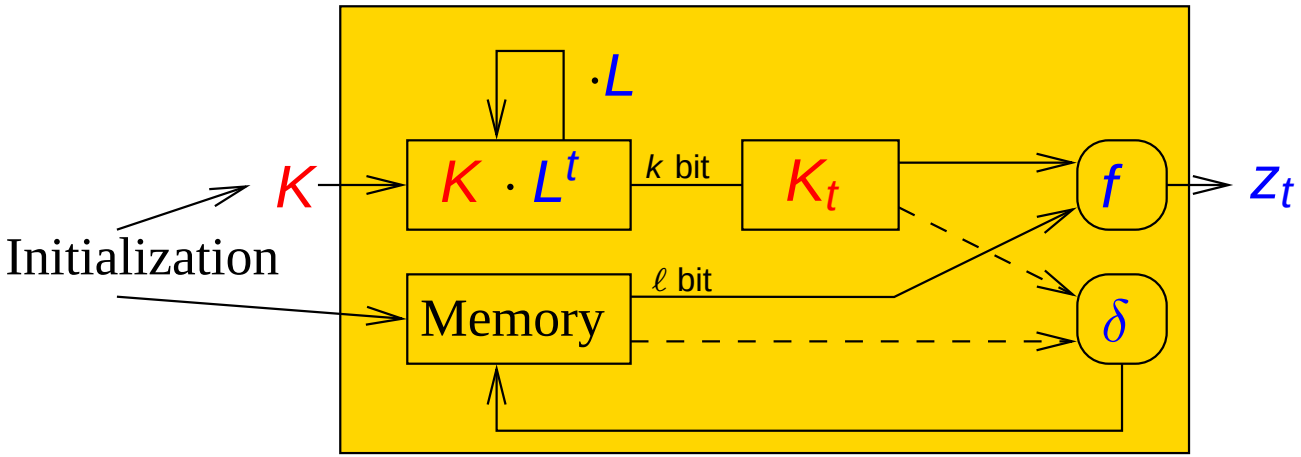
Linear feedback shift register (LFSR)

- Initial state $S_0 = (s_0, \ldots, s_{l-1})$
- State update at clock t:

$$\begin{aligned} S_t &= (s_t, s_{t+1}, \ldots, s_{t+l-1}) \\ &\downarrow \\ S_{t+1} &= (s_{t+1}, \ldots, s_{t+l-1}, \bigoplus_{i=0}^{l} \lambda_i \cdot s_{t+i}) \end{aligned}$$

- Output at clock t: s_t
- Update function is linear: $S_t = S_{t-1} \cdot L = S_0 \cdot L^t$

LFSR-based combiner with memory



Example (E_0)

- Bluetooth standard
- K: 128 bit, K_t : $k = 4$ bit, Memory: $\ell = 4$ bit

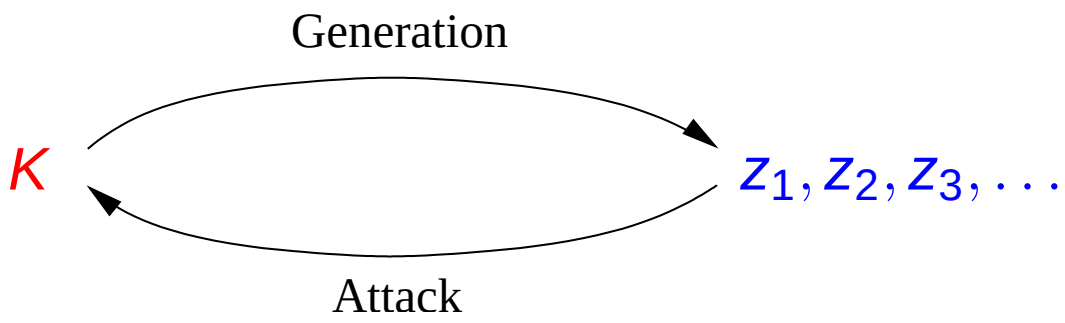
Keystream generator - Security

Kerckhoffs' model:

- Specifications of keystream generator are public
- Attacker knows some (or many) keystream bits z_t
- Security depends only on secrecy of K and memory

Attack:

- Often memory $\ll$ than K
- Recover secret key $K = (k_1, \ldots, k_n)$



Outline

Principles

Given: Specifications of the combiner with memory, the knowledge of several keystream bits z_t

Task: Recover LFSRs initial state $K \in \{0,1\}^n$

1. Set up system of equations in unknown K and known keystream bits z_t.

$$\begin{aligned} f_1(K, z, \ldots) &= 0 \\ &\vdots \\ f_N(K, z, \ldots) &= 0 \end{aligned}$$

2. Compute the solution to get K.

”One equation is enough”

Courtois, Meier; 2002, Krause, Armknecht; 2003

Assume one equation of the following kind is known for one t:

$$0 = F(K_t, \ldots, K_{t+r-1}, z_t, \ldots, z_{t+r-1}).$$

Then, the same equation can be re-used for any clock:

$$\begin{aligned}
0 &= F(K_0, \ldots, K_{r-1}, z_0, \ldots, z_{r-1}) \\
0 &= F(K_1, \ldots, K_r, z_1, \ldots, z_r) \\
0 &= F(K_2, \ldots, K_{r+1}, z_2, \ldots, z_{r+1}) \\
0 &= F(K_3, \ldots, K_{r+2}, z_3, \ldots, z_{r+2}) \\
&\vdots
\end{aligned}$$

This system of equations is not ordinary

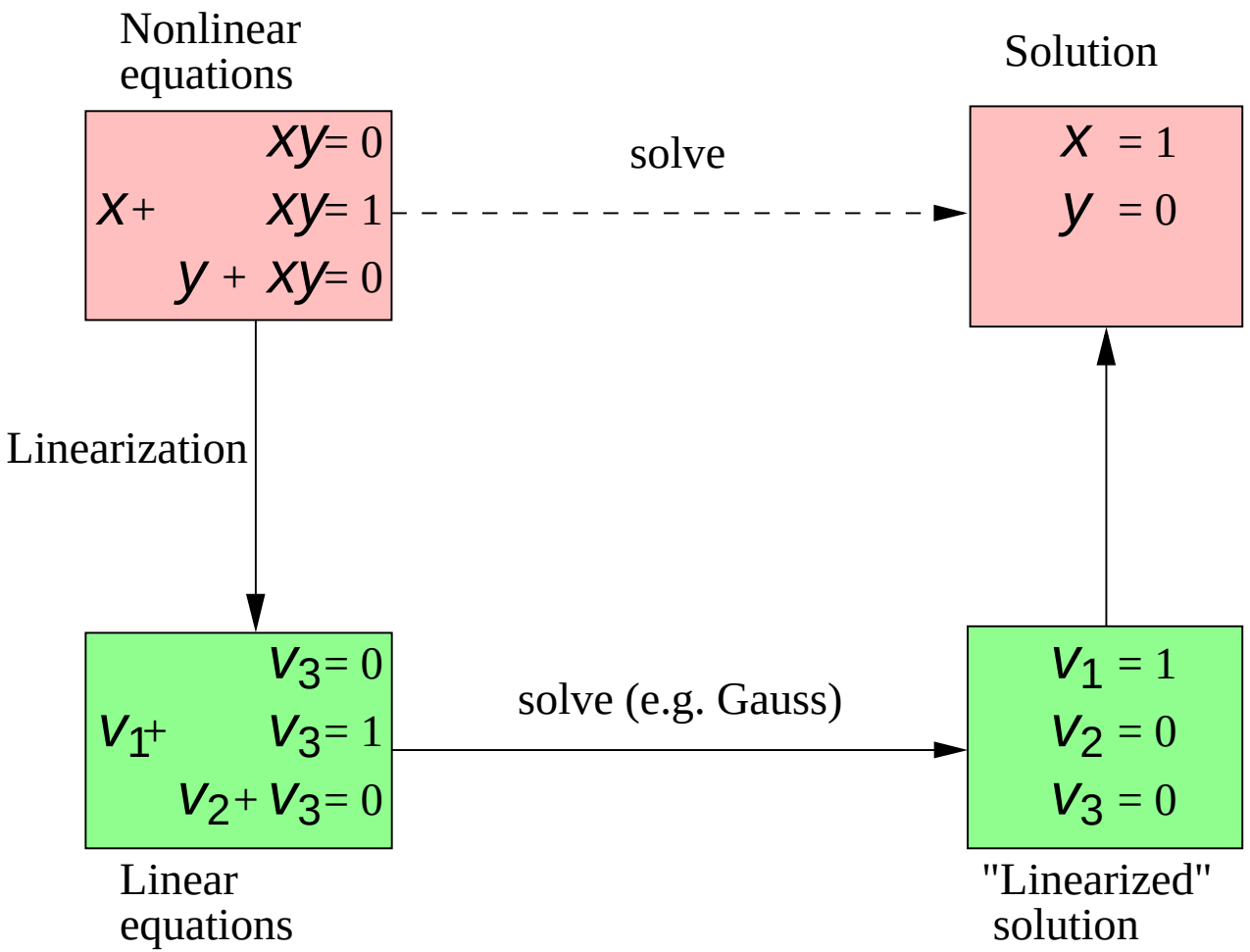
Given system of equations:

$$0 = F(K_t, \ldots, K_{t+r-1}, z_t, \ldots, z_{t+r-1}), \quad t \in T.$$

Observations:

- All equations have similar structure.
- All equations have the same degree d in K.
- The more keystream bits are known, the more equations can be set up.
- If number of linearly independent equations is equal to number of monomials, linearization is possible.

Linearization



Work effort of linearization

- System of equations in $n := |K|$ unknowns
- All equations of degree $\leq d$
- # Monomials $\leq \binom{n}{0} + \ldots + \binom{n}{d} \in \mathcal{O}(n^d)$
- Linearization takes:
  - $\mathcal{O}(n^{3d})$ operations
  - $\mathcal{O}(n^{2d})$ space

Effort is **polynomial** in key size n
but **exponential** in the degree d.

Recent attacks on stand-alone E_0

Attack	# key str.	# ops	memory
Divide and Conquer (Fluhrer, Lucks; 2001)	2^{43} 132	2^{73} 2^{84}	10638 small
BDD-based (Krause; 2002)	128	2^{77}	2^{77}
Algebraic Attack (Krause, Armknecht; 2003) (improved by Courtois; 2003)	$2^{23.07}$ $2^{23.44}$ succ.	$2^{68.48}$ $2^{54.51}$	$2^{46.14}$ $2^{36.84}$

Fast algebraic attack on E_0

- Alg. attack polynomial in $|K|$ but exponential in degree
- Idea: Exploit structure to simplify system of equations in precomputation step

$$z_0, z_1, z_2, \ldots, z_N \overset{\text{Set up}}{\Rightarrow} \left.\begin{array}{l} 0 = F(K_0, \ldots, K_3, z_0, \ldots, z_3) \\ 0 = F(K_1, \ldots, K_4, z_1, \ldots, z_4) \\ \vdots \end{array}\right\} \text{Deg. 4}$$

$$\Downarrow \text{Simplify}$$

$$K \overset{\text{Solve}}{\Leftarrow} \left.\begin{array}{l} 0 = G(K_0, \ldots, K_{T-1}, z_0, \ldots, z_{T-1}) \\ 0 = G(K_1, \ldots, K_T, z_1, \ldots, z_T) \\ \vdots \end{array}\right\} \text{Deg. 3}$$

Fast algebraic attack - Precomputation step

$$\sum_{i=0}^{T} c_i \cdot \underbrace{F(K_{t+i}, \ldots, K_{t+i+3}, z_{t+i}, \ldots, z_{t+i+3})}_{\deg 4} = \underbrace{G(K_t, \ldots, K_{t+T+3}, z_t, \ldots, z_{t+T+3})}_{\deg 3}$$

$$\begin{boxed}
\begin{aligned}
0 &= F(K_0, \ldots, K_3, z_0, \ldots, z_3) \\
0 &= F(K_1, \ldots, K_4, z_1, \ldots, z_4) \\
0 &= F(K_2, \ldots, K_5, z_2, \ldots, z_5) \\
&\vdots \\
0 &= F(K_T, \ldots, K_{T+3}, z_T, \ldots, z_{T+3})
\end{aligned}
\end{boxed}
\begin{aligned}
0 &= F(K_{T+1}, \ldots, K_{T+4}, z_{T+1}, \ldots, z_{T+4}) \\
0 &= F(K_{T+2}, \ldots, K_{T+5}, z_{T+2}, \ldots, z_{T+5}) \\
&\vdots
\end{aligned}
\quad \Rightarrow \quad G(K_0, \ldots, z_{T+3})$$

Fast algebraic attack - Precomputation step

$$\sum_{i=0}^{T} c_i \cdot \underbrace{F(K_{t+i}, \ldots, K_{t+i+3}, z_{t+i}, \ldots, z_{t+i+3})}_{\deg 4} = \underbrace{G(K_t, \ldots, K_{t+T+3}, z_t, \ldots, z_{t+T+3})}_{\deg 3}$$

$$\begin{aligned}
&0 = F(K_0, \ldots, K_3, z_0, \ldots, z_3) \\
&\boxed{\begin{aligned}
&0 = F(K_1, \ldots, K_4, z_1, \ldots, z_4) \\
&0 = F(K_2, \ldots, K_5, z_2, \ldots, z_5) \\
&\quad \vdots \\
&0 = F(K_T, \ldots, K_{T+3}, z_T, \ldots, z_{T+3}) \\
&0 = F(K_{T+1}, \ldots, K_{T+4}, z_{T+1}, \ldots, z_{T+4})
\end{aligned}} \\
&0 = F(K_{T+2}, \ldots, K_{T+5}, z_{T+2}, \ldots, z_{T+5}) \\
&\quad \vdots
\end{aligned}$$

$$G(K_0, \ldots, z_{T+3})$$

$$\Rightarrow \quad G(K_1, \ldots, z_{T+4})$$

Fast algebraic attack - Precomputation step

$$\sum_{i=0}^{T} c_i \cdot \underbrace{F(K_{t+i}, \ldots, K_{t+i+3}, z_{t+i}, \ldots, z_{t+i+3})}_{\deg 4} = \underbrace{G(K_t, \ldots, K_{t+T+3}, z_t, \ldots, z_{t+T+3})}_{\deg 3}$$

$$\begin{aligned}
0 &= F(K_0, \ldots, K_3, z_0, \ldots, z_3) \\
0 &= F(K_1, \ldots, K_4, z_1, \ldots, z_4) \\
0 &= F(K_2, \ldots, K_5, z_2, \ldots, z_5) \\
&\vdots \\
0 &= F(K_T, \ldots, K_{T+3}, z_T, \ldots, z_{T+3}) \\
0 &= F(K_{T+1}, \ldots, K_{T+4}, z_{T+1}, \ldots, z_{T+4}) \\
0 &= F(K_{T+2}, \ldots, K_{T+5}, z_{T+2}, \ldots, z_{T+5}) \\
&\vdots
\end{aligned}$$

$$\begin{aligned}
&G(K_0, \ldots, z_{T+3}) \\
&G(K_1, \ldots, z_{T+4}) \\
\Rightarrow \quad &G(K_2, \ldots, z_{T+5})
\end{aligned}$$

Outline

Task

Given:
Specifications of the combiner with memory
Recall:
Effort of linearization is polynomial in $|K|$ but exponential in degree d
Task:
Find a valid equation in K and z_t which is independent of memory states and has a degree as low as possible.

Some basic thoughts

- Output $z_t, \ldots, z_{t+r-1}$ depends only on $memory_t \in \{0,1\}^\ell$ and $K_t, \ldots, K_{t+r-1} \in \{0,1\}^k$
- If r is big enough, then set of possible values of $K_t, \ldots, K_{t+r-1}$ is a subset of $\{0,1\}^{k \cdot r}$. Let's call this set $S_{z_t,\ldots,z_{t+r-1}}$.
- Then, $f(K_t, \ldots, K_{t+r-1}, z_t, \ldots, z_{t+r-1}) = 0$ if $f(X_1, \ldots, X_r, z_t, \ldots, z_{t+r-1}) \in \mathbb{F}_2[X_1, \ldots, X_r]$ is zero on $S_{z_t,\ldots,z_{t+r-1}}$.

Annihilators

Definition (Annihilator)

f is annihilator of $S \subseteq \{0,1\}^m \Leftrightarrow f(x) = 0$ for all $x \in S$.

Theorem (Meier, Pasalic, Carlet; '04, Krause, Armknecht; '03)

Valid equation $\Leftrightarrow$ annihilators of certain sets.

The annihilator approach

- Fix $r \geq 1$
- Compute the sets S_Z for all $Z \in \{0,1\}^r$
- Compute annihilators of S_Z of **minimum degree**
- Basically two methods:
  1. Transform problem into linear system of equations ($\approx$ linearization)
  2. Compute Gröbner bases with degree respecting ordering

But, what if it is infeasible to compute the sets S_Z and/or the annihilators?

Direct analysis

In some cases, direct analysis brings valid equations (but not necessarily with the lowest degree).

Examples:

- Toyocrypt
- E_0
- Sober

Gröbner bases - elimination theory

1. Use specifications to set up equations in K, z_t and memory states:

$$f(memory_t, K_t) = z_t, \quad \delta(memory_t, K_t) = memory_{t+1}.$$

2. Define ideal with these equations.
3. Use Gröbner bases and elimination theory to get rid off memory variables $memory_t$.

Task

Given:
System of equations:

$$\begin{aligned} 0 &= F(K_{t_1}, \ldots, K_{t_1+r-1}, z_{t_1}, \ldots, z_{t_1+r-1}) \\ 0 &= F(K_{t_2}, \ldots, K_{t_2+r-1}, z_{t_2}, \ldots, z_{t_2+r-1}) \\ 0 &= F(K_{t_3}, \ldots, K_{t_3+r-1}, z_{t_3}, \ldots, z_{t_3+r-1}) \\ &\vdots \end{aligned}$$

Task:
Compute the solution K

Linearization

Advantages:

- Effort polynomial in key size
- Easy to analyze

Disadvantages:

- Needs to store large (sparse) matrices.
- The knowledge of many keystream bits is necessary.
- Requires to have the maximum amount of linearly independent equations.

Example (Alg. attack on E_0)

- Key length: 128 bit
- Needs about 2^{23} linearly independent equations.

Extended linearization

Proposed solutions:

- XL (Shamir, Patarin, Courtois, Klimov; 2000)
  - Is inferior compared to F_5 (Ars, Faugére, Imai, Kawazoe, Sugita; 2004)
  - Running time is higher than predicted (Diem; 2004)
- XSL (Courtois, Pieprzyk; 2002)
  - Claimed effort is rather doubted (Cid, Leurent; 2005)

$\Rightarrow$ Linearization and its derivatives are not suited for practical attacks.

Solving overdefined system of equations with Gröbner bases

- F_5 is an adequate substitute for linearization (Ars, Faugére; 2003)
- Run time of F_5 is subexponential if #eqs $\approx n \cdot \log_2(n)$ (Bardet, Faugére, Salvy; 2003)

Outline

Summary

- Algebraic attacks are based on generating and solving system of equations.
- ”One equation is enough”
- All equations have a degree $\leq d \Rightarrow$ linearization.
- Special structure $\Rightarrow$ fast algebraic attacks

Open problems

- Keystream generators without linear feedback
- Efficient methods to find low-degree equations
- Criteria to exclude the existence of low-degree equations over many clocks
- Faster algorithms to solve system of equations