Stream Cipher Cryptanalysis - Algebraic Attacks

Frederik Armknecht
armknecht@th.informatik.uni-mannheim.de

University of Mannheim
(Germany)
Overview

1. Basics
2. Algebraic Attacks
3. Combiners without memory
4. Combiners with memory
5. Results
   (a) First Result
   (b) Second Result
6. Final remarks
1. Basics
Situation

Alice sends message $m = (m_1, m_2, \ldots)$ to Bob
Key Stream Generators

Key Stream Generator

\( K \)

Key Stream Bits

\( z_t \)

Alice

\( m_t \)

Bob

\( m_t \)

Eve

\( c_t \)

Message bits

Stream Cipher Cryptanalysis - Algebraic Attacks – p.5/65
Protocol

Alice:
- Encrypts message: \( c_t = m_t \oplus z_t \)
- Sends \( c_t \) to Bob

Bob:
- Receives \( c_t \) from Alice
- Decrypts message: \( c_t \oplus z_t = m_t \oplus z_t \oplus z_t = m_t \)

Eve:
- Eavesdrop only \( c_t \) → useless
Design Criteria

- Efficiency
- Security
Efficiency

- Fast = produce many key stream bits $z_t$ per second
- Use only little system resources
Hard to decrypt without knowing $\mathcal{K}$
- What does "hard" mean?

- Which information has Eve?
**Attack Scenario**

Eve *does know*:
- Key stream generator
- Some key stream bits $z_t$

Eve *doesn’t know*:
- Secret key $K$

Kerckhoffs’ principle:
Security depends only on secrecy of $K$
Eve: Recover secret key $\mathcal{K}$

- Fast correlation attacks (see talk of W. Meier)
- Backtracking attacks
- BDD-based attacks
- Algebraic attacks (see talk of W. Meier)
- ...
2. Algebraic Attacks
Overview

Algebraic attacks against

- **Stream ciphers**
  - Toyocrypt, LILI-128 (Courtois, Meier; 2003)
  - Bluetooth key stream generator (Armknecht; 2002)

- **Block ciphers**
  - AES, Serpent (Courtois, Pieprzyk; 2002)
1. Set up system of equations in key bits and output bits
2. Solve it
Linear feedback shift register

Initial value = secret key $K$

```
Linear feedback shift register

\[
\begin{align*}
\mathbf{z}(t) &= \mathbf{x}(1)_t \\
\mathbf{z}(t) &= \mathbf{a}(1)_t \\
\mathbf{z}(t) &= \mathbf{b}(1)_t \\
\mathbf{x}(1)_t &= \mathbf{c}(1)_t \\
\mathbf{x}(2)_t &= \mathbf{c}(2)_t \\
\mathbf{x}(n)_t &= \mathbf{c}(n)_t
\end{align*}
\]

Coefficients

Internal State

Output

Initial value = secret key $K$
```
LFSR - Example

Clock | $x_t^{(3)}$ | $x_t^{(2)}$ | $x_t^{(1)}$ | Output
--- | --- | --- | --- | ---
1 | $x^{(3)}$ | $x^{(2)}$ | $x^{(1)}$ | $z_1 = x^{(1)}$
2 | $x^{(1)} \oplus x^{(3)}$ | $x^{(3)}$ | $x^{(2)}$ | $z_2 = x^{(2)}$
3 | $x^{(1)} \oplus x^{(2)} \oplus x^{(3)}$ | $x^{(1)} \oplus x^{(3)}$ | $x^{(3)}$ | $z_3 = x^{(3)}$
4 | $x^{(1)} \oplus x^{(2)}$ | $x^{(1)} \oplus x^{(2)} \oplus x^{(3)}$ | $x^{(1)} \oplus x^{(3)}$ | $z_4 = x^{(1)} \oplus x^{(3)}$
... | ... | ... | ... | ...

Stream Cipher Cryptanalysis - Algebraic Attacks – p.17/65
Linear feedback shift register (LFSR) of length $n$:

$$\mathcal{K} = (x^{(1)}, \ldots, x^{(n)})$$

Advantages:
- Very fast and simple
- $z_t \approx$ random bits

Disadvantage:
- Linearity: $z_t = c_1^t x^{(1)} \oplus \ldots \oplus c_n^t x^{(n)} =: F_{z_t}(x^{(1)}, \ldots, x^{(n)})$
Algebraic attack on LFSRs:

1. System of linear equations

\[ z_1 = c_1^1 x^{(1)} \oplus \ldots \oplus c_1^n x^{(n)} = F_1(x^{(1)}, \ldots, x^{(n)}) \]

\[ z_2 = c_2^1 x^{(1)} \oplus \ldots \oplus c_2^n x^{(n)} = F_2(x^{(1)}, \ldots, x^{(n)}) \]

\[ \vdots \]

2. Solve it (e. g. Gauss)
Use LFSRs to build cryptographically stronger key stream generators?
3. Combiners without memory
Combiners without memory

$k$ LFSRs, 0 memory bits ($(k, 0)$-combiner):

\[
\begin{align*}
\mathcal{LFSR} 1 & \quad x^{(1)}_t \\
\vdots & \quad \vdots \\
\mathcal{LFSR} k & \quad x^{(k)}_t \\
\mathcal{K}_1 & \\
\vdots & \\
\mathcal{K}_k & \\
\mathcal{K} = (\mathcal{K}_1, \ldots, \mathcal{K}_k) & n \text{ bit secret key}
\end{align*}
\]

\[F \quad z_t\]

nonlinear
Eve doesn’t know any linear equations
⇒ System of linear equations?

Eve does know the following equation:

\[ z_t = F(x_t^{(1)}, \ldots, x_t^{(k)}) \]

⇒ System of nonlinear equations!
1. System of nonlinear equations

\[ z_1 = F(x_1^{(1)}, \ldots, x_1^{(k)}) = F_1(x_1, \ldots, x_n) \]
\[ z_2 = F(x_2^{(1)}, \ldots, x_2^{(k)}) = F_2(x_1, \ldots, x_n) \]
\[ \vdots \]

2. Solving (e.g. Gröbner Bases, Linearization, XL, …)
Linearization: Idea

1. System of nonlinear equations → system of linear equations
2. Solve system of linear equations (hope: only few solutions)
3. Use solution(s) to get solution of original system of nonlinear equations
System of **linear** equations
- equation: combination of variables (e. g. $x$, $y$)

System of **nonlinear** equations
- equation: combination of monomials (=products of variables, e. g. $x$, $y$, $xy$)

Idea: Treat monomials like (new) variables $\Rightarrow$ no products
System of **nonlinear** equations:

\[
\begin{align*}
x \oplus y \oplus xy &= 1 \\
y \oplus xy &= 1
\end{align*}
\]

**New Variables:** \( M_1 := x, \ M_2 := y \) and \( M_3 := xy \)

**New system of linear equations:**

\[
\begin{align*}
M_1 \oplus M_2 \oplus M_3 &= 1 \\
M_2 \oplus M_3 &= 1
\end{align*}
\]
Applying Gauss reveals:

\[ M_1 = 0 \]
\[ M_2 \oplus M_3 = 1 \]

⇒ Two solutions:

\[ M_1 = 0, \quad M_2 = 0, \quad M_3 = 1 \]
\[ M_1 = 0, \quad M_2 = 1, \quad M_3 = 0 \]
1. Solution

\[
0 = M_1 = x \\
0 = M_2 = y \\
1 = M_3 = xy
\]

Solution doesn’t make sense!
2. Solution

\[ 0 = M_1 = x \]
\[ 1 = M_2 = y \]
\[ 0 = M_3 = xy \]

Solution correct!
3 LFSRs, 0 memory bits \((3, 0)\)-combiner:

\[
\begin{align*}
\text{LFSR A} & \rightarrow a_t \\
\text{LFSR B} & \rightarrow b_t \\
\text{LFSR C} & \rightarrow c_t
\end{align*}
\]

\[
\begin{align*}
\text{if } c_t &= 0 \\
\text{then } z_t &= a_t \\
\text{else } z_t &= b_t
\end{align*}
\]

\[z_t\]

degree 2
Combining Function $F$

- LFSR A, LFSR B, LFSR C output $a_t$, $b_t$, $c_t$ at clock $t$
- Combiner function

$$z_t = F(a_t, b_t, c_t) = \begin{cases} a_t & , c_t = 0 \\ b_t & , c_t = 1 \end{cases}$$

$$= a_t \oplus a_t c_t \oplus b_t c_t$$

$\Rightarrow \deg(F) = 2$
System of equations

\[
\begin{align*}
  z_1 &= F(a_1, b_1, c_1) = a_1 \oplus a_1c_1 \oplus b_1c_1 \\
  z_2 &= F(a_2, b_2, c_2) = a_2 \oplus a_2c_2 \oplus b_2c_2 \\
  z_3 &= F(a_3, b_3, c_3) = a_3 \oplus a_3c_3 \oplus b_3c_3 \\
  z_4 &= F(a_4, b_4, c_4) = a_4 \oplus a_4c_4 \oplus b_4c_4 \\
  \vdots \\
  a_t, b_t, c_t &= \text{outputs of LFSRs} \\
  \Rightarrow &\text{only monomials of degree } \leq \deg(F) \text{ occur!}
\end{align*}
\]
### Simple example

<table>
<thead>
<tr>
<th>Clock</th>
<th>LFSR A</th>
<th>LFSR B</th>
<th>LFSR C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
</tr>
<tr>
<td>2</td>
<td>$a_2$</td>
<td>$b_1$</td>
<td>$c_2$</td>
</tr>
<tr>
<td>3</td>
<td>$a_1 \oplus a_2$</td>
<td>$b_1$</td>
<td>$c_3$</td>
</tr>
<tr>
<td>4</td>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1 \oplus c_3$</td>
</tr>
<tr>
<td>5</td>
<td>$a_2$</td>
<td>$b_1$</td>
<td>$c_1 \oplus c_2 \oplus c_3$</td>
</tr>
<tr>
<td>6</td>
<td>$a_1 \oplus a_2$</td>
<td>$b_1$</td>
<td>$c_1 \oplus c_2$</td>
</tr>
<tr>
<td>7</td>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_2 \oplus c_3$</td>
</tr>
<tr>
<td>8</td>
<td>$a_2$</td>
<td>$b_1$</td>
<td>$c_1$</td>
</tr>
<tr>
<td></td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>
**System of nonlinear equations**

\[ z_t = F(a_t, b_t, c_t) = a_t \oplus a_t c_t \oplus b_t c_t \]

<table>
<thead>
<tr>
<th>Clock</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( z_1 = a_1 \oplus a_1 c_1 \oplus b_1 c_1 )</td>
</tr>
<tr>
<td>2</td>
<td>( z_2 = a_2 \oplus a_2 c_2 \oplus b_1 c_2 )</td>
</tr>
</tbody>
</table>
| 3     | \( z_3 = (a_1 \oplus a_2) \oplus (a_1 \oplus a_2) \cdot c_3 \oplus b_1 c_3 \)  
       |        | \( = a_1 \oplus a_2 \oplus a_1 c_3 \oplus a_2 c_3 \oplus b_1 c_3 \) |
| 4     | \( z_4 = a_1 \oplus a_1 \cdot (c_1 \oplus c_3) \oplus b_1 \cdot (c_1 \oplus c_3) \)  
       |        | \( = a_1 \oplus a_1 c_1 \oplus a_1 c_3 \oplus b_1 c_1 \oplus b_1 c_3 \) |
|       | \( \vdots \) | \( \vdots \) |
Linearization

Occuring monomials:

\[ a_1, a_2, b_1, c_1, c_2, c_3, \]
\[ a_1 c_1, a_1 c_2, a_1 c_3, a_2 c_1, a_2 c_2, a_2 c_3, \]
\[ b_1 c_1, b_1 c_2, b_1 c_3 \]

New variables:

\[ M_1, M_2, M_3, M_4, M_5, M_6, \]
\[ M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, \]
\[ M_{13}, M_{14}, M_{15} \]
## System of linear equations

<table>
<thead>
<tr>
<th>Clock</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$z_1 = M_1 \oplus M_7 \oplus M_{13}$</td>
</tr>
<tr>
<td>2</td>
<td>$z_2 = M_2 \oplus M_{11} \oplus M_{14}$</td>
</tr>
<tr>
<td>3</td>
<td>$z_3 = M_1 \oplus M_2 \oplus M_9 \oplus M_{12} \oplus M_{15}$</td>
</tr>
<tr>
<td>4</td>
<td>$z_4 = M_1 \oplus M_7 \oplus M_9 \oplus M_{13} \oplus M_{15}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Complexity

\[ n = \text{number of bits of } K, \quad d = \text{degree of } F \]

- # monomials: \( \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{d} \approx n^d \) (polynomial in \( n \))
- Key stream bits needed: \( \approx n^d \)
- Operations: \( \approx n^{d\omega}, \ \omega \leq 3 \)

\Rightarrow \text{Algebraic attacks in polynomial time (if } d \text{ and } k \text{ const.)}
Improving security

Higher degree $d \Rightarrow$

+ Higher security against algebraic attacks
+ Lower security against correlation attacks
– Lower efficiency

Increasing $d$ is a bad strategy!

Another approach?
4. Combiners with memory
Combiners with memory

$k$ LFSRs, $l$ memory bits ($(k, l)$-combiner):

\[ z_t = x_t^{(1)} \oplus \cdots \oplus x_t^{(k)} \oplus F(c_t^{(1)}, \ldots, c_t^{(l)}) \oplus C(c_{t+1}^{(1)}, \ldots, c_{t+1}^{(l)}) \]
Bluetooth - standard for wireless communication connecting

- Mobile Phones
- Handhelds
- Personal Computers
- Automotives
Example: Bluetooth

3 LFSRs, 0 memory bits \((3,0)-\text{combiner}\):
### Best attacks known so far

<table>
<thead>
<tr>
<th>Attack</th>
<th>$z_t$ needed</th>
<th>operations</th>
<th>memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Divide and Conquer; SAC ’01 (Fluhrer, Lucks)</td>
<td>$2^{43}$</td>
<td>$2^{73}$</td>
<td>$10638$ small</td>
</tr>
<tr>
<td></td>
<td>132</td>
<td>$2^{84}$</td>
<td></td>
</tr>
<tr>
<td>BDD-based; Eurocrypt ’02 (Krause)</td>
<td>128</td>
<td>$2^{77}$</td>
<td>$2^{77}$</td>
</tr>
<tr>
<td>Algebraic Attack; Crypto ’03 (Krause, Armknecht) (improved by Courtois)</td>
<td>$2^{23}$ succ.</td>
<td>$2^{46}$</td>
<td>$2^{46}$</td>
</tr>
</tbody>
</table>
Algebraic attack?

Polynomial-size systems of low-degree equations for \((k, l)\)-combiners?
5. Results
(a) First result
1. Approach

Combining function:

\[ F(x_t^{(1)}, \ldots, x_t^{(k)}, c_t^{(1)}, \ldots, c_t^{(l)}) = z_t \]

<table>
<thead>
<tr>
<th>Clock</th>
<th>Memory Bits</th>
<th>Output LFSRs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>t</td>
<td>(c_t^{(1)}, \ldots, c_t^{(l)})</td>
<td>(x_t^{(1)}, \ldots, x_t^{(k)})</td>
<td>(z_t)</td>
</tr>
<tr>
<td>(t - 1)</td>
<td>(c_{t-1}^{(1)}, \ldots, c_{t-1}^{(l)})</td>
<td>(x_{t-1}^{(1)}, \ldots, x_{t-1}^{(k)})</td>
<td>(z_{t-1})</td>
</tr>
<tr>
<td>(t - 2)</td>
<td>(c_{t-2}^{(1)}, \ldots, c_{t-2}^{(l)})</td>
<td>(x_{t-2}^{(1)}, \ldots, x_{t-2}^{(k)})</td>
<td>(z_{t-2})</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>1</td>
<td>(c_1^{(1)}, \ldots, c_1^{(l)})</td>
<td>(x_1^{(1)}, \ldots, x_1^{(k)})</td>
<td>(z_1)</td>
</tr>
</tbody>
</table>
Problem

\[ z_t = F(\underbrace{x_t^{(1)}, \ldots, x_t^{(k)}}_{\text{secret key bits}}, \underbrace{c_t^{(1)}, \ldots, c_t^{(l)}}_{\text{new unknowns}}) \]

\((x_t^{(1)}, \ldots, x_t^{(k)})\) linear combinations of secret key bits

\((c_t^{(1)}, \ldots, c_t^{(l)})\) new unknowns !!

⇒ Number of unknowns after \(t\) clocks:

\[ n + t \cdot l > t \]

⇒ More unknowns than equations

⇒ System of equations unsolvable

Stream Cipher Cryptanalysis - Algebraic Attacks – p.48/65
Avoiding the new unknowns?

Output LFSRs

\( (x_1^{(1)}, \ldots, x_{k}^{(k)}) \) \( \quad \cdots \quad \) \( (x_{t-1}^{(1)}, \ldots, x_{t-1}^{(k)}) \)

\( (c_1^{(1)}, \ldots, c_{l}^{(l)}) \) \( \quad \rightarrow \quad C \quad \rightarrow \quad \cdots \quad \rightarrow \quad C \rightarrow (c_t^{(1)}, \ldots, c_t^{(l)}) \)

Initial value of memory bits
2. Approach

\[ F_t( x_t^{(1)}, \ldots, x_t^{(k)}; x_1^{(1)}, \ldots, x_{t-1}^{(k)}; c_1^{(1)}, \ldots, c_1^{(l)} ) = z_t \]

<table>
<thead>
<tr>
<th>Clock</th>
<th>Memory Bits</th>
<th>Output LFSRs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( t )</td>
<td>( c_t^{(1)}, \ldots, c_t^{(l)} )</td>
<td>( x_t^{(1)}, \ldots, x_t^{(k)} )</td>
<td>( z_t )</td>
</tr>
<tr>
<td>( t - 1 )</td>
<td>( c_{t-1}^{(1)}, \ldots, c_{t-1}^{(l)} )</td>
<td>( x_{t-1}^{(1)}, \ldots, x_{t-1}^{(k)} )</td>
<td>( z_{t-1} )</td>
</tr>
<tr>
<td>( t - 2 )</td>
<td>( c_{t-2}^{(1)}, \ldots, c_{t-2}^{(l)} )</td>
<td>( x_{t-2}^{(1)}, \ldots, x_{t-2}^{(k)} )</td>
<td>( z_{t-2} )</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( 1 )</td>
<td>( c_1^{(1)}, \ldots, c_1^{(l)} )</td>
<td>( x_1^{(1)}, \ldots, x_1^{(k)} )</td>
<td>( z_1 )</td>
</tr>
</tbody>
</table>
Polynomial time attack?

\[ F_t(x_t^{(1)}, \ldots, x_t^{(k)}; x_1^{(1)}, \ldots, x_{t-1}^{(k)}, c_1^{(1)}, \ldots, c_1^{(l)}) = z_t \]

**Advantage:**  
# unknowns = \( n + l \)

⇒ system of equations solvable (at least in principle)

**Disadvantage:** \( \deg(F_t) \) can be arbitrarily high

⇒ #monomials can go up to \( 2^{n+l} \) (exp. in \( n \))

⇒ Alg. att. using linearization is not in polynomial time!
3. Approach

"Ad-hoc" equation: \[ \tilde{F}(x_t^{(1)}, \ldots, x_t^{(k)}, z_1, \ldots, z_{t+r-1}) = 0 \]

<table>
<thead>
<tr>
<th>Clock</th>
<th>Memory Bits</th>
<th>Output LFSRs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>(t + r - 1)</td>
<td>(c_t^{(1)}, \ldots, c_t^{(l)})</td>
<td>(x_t^{(1)}, \ldots, x_t^{(k)})</td>
<td>(z_{t+r-1})</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>(t)</td>
<td>(c_t^{(1)}, \ldots, c_t^{(l)})</td>
<td>(x_t^{(1)}, \ldots, x_t^{(k)})</td>
<td>(z_t)</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>(1)</td>
<td>(c_1^{(1)}, \ldots, c_1^{(l)})</td>
<td>(x_1^{(1)}, \ldots, x_1^{(k)})</td>
<td>(z_1)</td>
</tr>
</tbody>
</table>
System of "Ad-hoc" equations

\[
0 = \tilde{F}(x_1^{(1)}, \ldots, x_r^{(k)}, z_1, \ldots, z_r)
\]

\[
0 = \tilde{F}(x_2^{(1)}, \ldots, x_{r+1}^{(k)}, z_2, \ldots, z_{r+1})
\]

\[\vdots\]

- Existence ?
- Degree ?
Theorem (Krause, Armknecht; 2003)

\[ \forall (k, l) \text{-combiner} \exists \text{ "Ad-hoc" equation } \tilde{F} \neq 0 \text{ of} \]

\[ \text{degree } \leq \left\lfloor \frac{k(l + 1)}{2} \right\rfloor \]

with

\[ 0 = \tilde{F}(x_t^{(1)}, \ldots, x_t^{(k)}, z_t, \ldots, z_{t+l}) \]
Polynomial time attack

- Degree of "Ad-hoc" equations bounded

- ⇒ Number of monomials is polynomial in $n$

- ⇒ Polynomial time attacks for $(k, l)$-combiners (poly. in $n$)
5. Results

(b) Second result
Finding

"Ad-hoc" equations $\tilde{F}$

of low degree?
Theorem (Krause, Armknecht; 2003)

\((k, l)\)-combiner, degree \(d \geq 1\), number of clocks \(r\).

\(\exists\) algorithm for finding all "Ad-hoc" equations \(\tilde{F}\) of degree \(\leq d\) with

\[
0 = \tilde{F}\left(x^{(1)}_t, \ldots, x^{(k)}_{t+r-1}, z_t, \ldots, z_{t+r-1}\right)
\]

\(r\) successive clocks
Bluetooth standard for wireless communication: 
(4, 4)-combiner

- Theorem: "Ad-hoc" equation of degree $\leq 10$
- Algorithm: "Ad-hoc" equation of degree $4$ !

$\Rightarrow$ Algebraic attack on Bluetooth:

- solve system of linear eq. with $\approx 2^{23}$ unknowns
- needs $\approx 2^{23}$ keystream bits $z_t$

Best attack on Bluetooth !
Ad-hoc relation for Bluetooth

\[ 0 = \tilde{F}(x_t^{(1)}, \ldots, x_t^{(k)}, z_t, \ldots, z_{t+3}) \]

- 4 successive clocks
- degree 4

More successive clocks \(\Rightarrow\) lower degree?
Better Ad-hoc relation for Bluetooth?

<table>
<thead>
<tr>
<th># succ. clocks</th>
<th>lowest degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/</td>
</tr>
<tr>
<td>2</td>
<td>/</td>
</tr>
<tr>
<td>3</td>
<td>/</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>( r )</td>
<td>( \leq 3 ) ?</td>
</tr>
</tbody>
</table>
Open questions

- Better ad-hoc relations for Bluetooth?
- Lower bound for degree of ad-hoc relations?
- Faster algorithm for solving system of equations
- ...

Stream Cipher Cryptanalysis - Algebraic Attacks – p.62/65
6. Final remarks
Summary

- Algebraic Attacks
- LFSRs, Combiners without memory
- Combiners with memory (e.g. Bluetooth)
- Result: Existence "Ad-hoc" equations $\tilde{F}$
- Result: Algorithm for finding them
- Result: Application to Bluetooth (best attack yet)
- Block ciphers
- Provable security
- Secret Key / Public Key Cryptography