Algebraic Attacks on the Courtois Toy Cipher

"However one should understand that the attack is quite simple and fatally will be re-discovered (and published)." [Cou06]

Martin Albrecht (malb@informatik.uni-bremen.de)

Outline

Outline

Algebraic Attacks: The Idea I

- Given a block cipher with a linear layer, key schedule, key addition, and S-boxes (XSL-cipher).
- Express this cipher as a multivariate (quadratic) equation system; usually over $\mathbb{F}_{2^n}$ with $n \geq 1$.
  - Easy for linear layer and key addition: linear equations.
  - Find as many linear-independent equations for S-boxes as possible either by brute force or use Gröbner bases.
  - If key schedule features S-boxes do the same, else use linear equations.
- Solve this equation system, the solution to the key variables is the desired key.

Algebraic Attacks: The Idea II

- Hope/Fear that these equation systems may be solved faster than the general (NP-hard) case.
  - Very few solutions (often only one).
  - Equations over (very small) finite fields.
  - May exclude any solutions from the algebraic closure.
  - Equation systems often **overdefined** yet **sparse**.
  - Equation systems highly **structured**.
- In a nutshell: Find the variety of an ideal spanned by a sparse, overdefined, highly structured given basis.

Algebraic Attacks: Equation Systems

AES (128-bit, 10 rounds) 8000 equations in 1600 variables over $\mathbb{F}_2$ or 5248 equations in 3968 variables over $\mathbb{F}_{2^8}$.

CTC (255-bit, 6 rounds) 11985 equations in 6375 variables over $\mathbb{F}_2$.

XSL $Bs \cdot Nr$ linear equations for the linear layer,
$Bs \cdot (Nr + 1)$ equations for the key addition,
$u \cdot Bs \cdot Nr$ equations for the S-boxes,
$v \cdot Nr$ equations for the key schedule. (u, v vary depending on cipher)

Outline

Background

On May 13th, 2006 Nicolas Courtois published a paper [Cou06] describing yet another toy cipher to attack algebraically. He dupes this cipher Courtois Toy Cipher (CTC).
In this paper he also provides steps to produce an overdefined system of multivariate polynomial equations that – if solved – provides the key used in the encryption process. Solving this equation system to break a cipher is called algebraic attack on block ciphers in literature.
He also claims to have developed a "fast algebraic attack on block ciphers" which solves this equation system for a quite large configuration. However, this attack is unpublished.

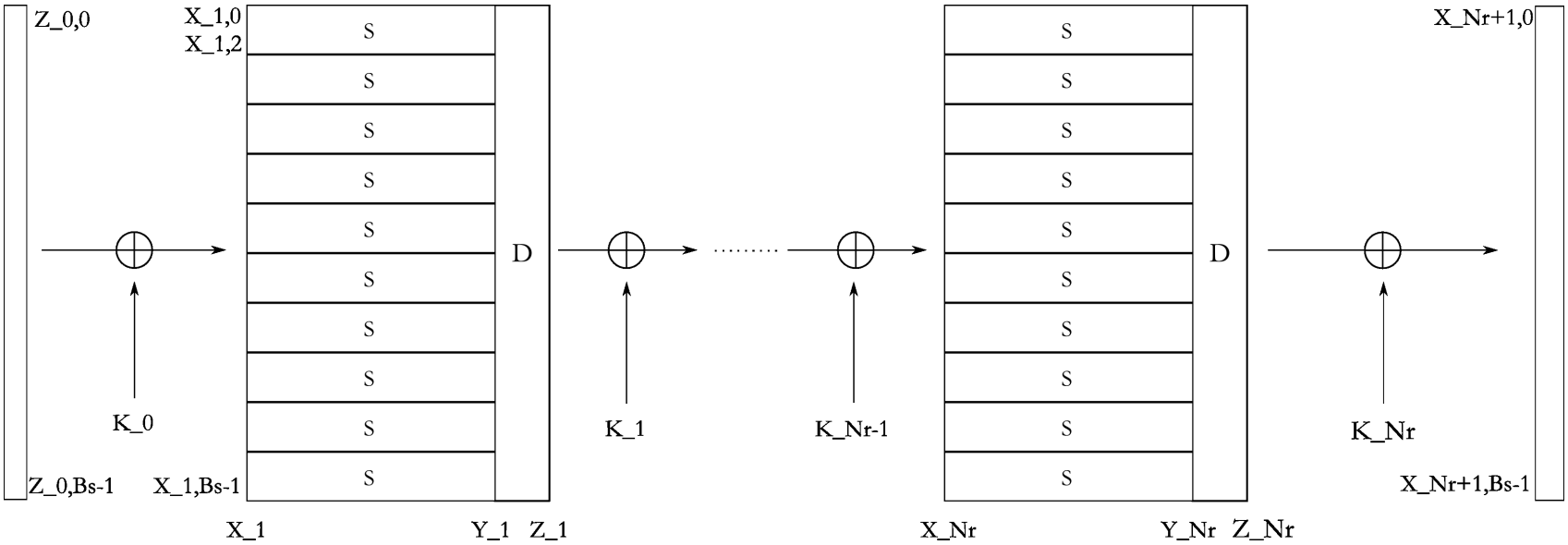
Why CTC?

- Resist the temptation to present yet another toy cipher, use toy cipher by "big names": CTC [Cou06] or SR [CMR05].
- CTC scales up (255-bit, six rounds) and down (3-bit, one round) well
- nice properties: overdefined S-box, thin diffusion layer
- "known" to be broken using an algebraic attack
- was expected to be resistant against well-known attacks (not true anymore, see [DK06])

CTC Overview: Bird's Eye View I

The cipher operates on block sizes which are multiples of 3. So the block size is $B \cdot s$ where $s = 3$ and B may be chosen. The cipher is defined in rounds where each round performs the same operation on the input data except that a different round key is added each time. The output of round $i-1$ is the input of round i. Each round consists of a parallell application of B S-boxes, the application of the linear diffusion layer, and a final key addition of the round key.

CTC Overview: Bird's Eye View II



CTC Overview: Some Details

- The S-Box is defined over $GF(2^3)$ as the non-linear random permutation $[7, 6, 0, 4, 2, 5, 1, 3]$. The transformation from $GF(2)^3$ to $GF(2^3)$ is the "natural"-mapping.
- The diffusion layer is very thin:

$$Z_{i,(257\%Bs)} = Y_{i,0} \text{ for all } i = 1 \ldots N_r,$$
$$Z_{i,(j\cdot 1987+257\%Bs)} = Y_{i,j} + Y_{i,(j+137\%Bs)} \text{ for } j \neq 0 \text{ and all } i.$$

- The key schedule is a simple permutation of wires.

$$
\begin{aligned}
P_0 + K_{0,0} + X_{1,0},\\
P_1 + K_{0,1} + X_{1,1},\\
P_2 + K_{0,2} + X_{1,2},
\end{aligned}
$$

$$
\begin{aligned}
1 + X_{1,0} + X_{1,1} + X_{1,1}X_{1,0} + X_{1,2} + Y_{1,0},\\
1 + X_{1,1} + X_{1,2}X_{1,0} + Y_{1,1},\\
1 + X_{1,1} + Y_{1,0}X_{1,0} + Y_{1,1},\\
X_{1,2} + Y_{1,0} + Y_{1,1} + Y_{1,1}X_{1,0},\\
1 + X_{1,0} + X_{1,1} + X_{1,2}X_{1,1} + Y_{1,0} + Y_{1,1} + Y_{1,2},\\
1 + X_{1,0} + X_{1,1} + Y_{1,0} + Y_{1,0}X_{1,1} + Y_{1,1} + Y_{1,2},\\
X_{1,0} + Y_{1,1}X_{1,1} + Y_{1,2}X_{1,0},\\
1 + X_{1,1} + X_{1,2} + Y_{1,0} + Y_{1,2}X_{1,0} + Y_{1,2}X_{1,1},\\
Y_{1,0} + Y_{1,0}X_{1,2} + Y_{1,2} + Y_{1,2}X_{1,0},\\
X_{1,0} + X_{1,2} + Y_{1,0} + Y_{1,1}X_{1,2} + Y_{1,2},\\
1 + X_{1,0} + X_{1,1} + Y_{1,1} + Y_{1,2}X_{1,0} + Y_{1,2}X_{1,2},\\
X_{1,0} + Y_{1,1}Y_{1,0} + Y_{1,2},\\
1 + X_{1,0} + X_{1,1} + Y_{1,1} + Y_{1,2} + Y_{1,2}Y_{1,0},\\
X_{1,0} + X_{1,2} + Y_{1,0} + Y_{1,1} + Y_{1,2} + Y_{1,2}Y_{1,1},
\end{aligned}
$$

$$
\begin{aligned}
Y_{1,0} + Y_{1,1} + Z_{1,0},\\
Y_{1,1} + Y_{1,2} + Z_{1,1},\\
Y_{1,0} + Z_{1,2},
\end{aligned}
$$

$$
\begin{aligned}
K_{0,1} + K_{1,0},\\
K_{0,2} + K_{1,1},\\
K_{0,0} + K_{1,2},
\end{aligned}
$$

$$
\begin{aligned}
Z_{1,0} + K_{1,0} + C_0,\\
Z_{1,1} + K_{1,1} + C_1,\\
Z_{1,2} + K_{1,2} + C_2,
\end{aligned}
$$

Outline

What is SAGE?

- Computer algebra system and open-source math software distribution started in 2005 and led by William Stein (UW, Seattle)
- Interfaces and ships with many computer algebra systems including *Singular*
- Makes it very easy to use e.g. Singular, PARI, Python code, and own optimized C code
- Gröbner Basis calculations are performed using Singular (with proper credits)
- Low-level multivariate polynomial arithmetic is slow and incomplete, my plan to change this
- Used to implement my thesis

Outline

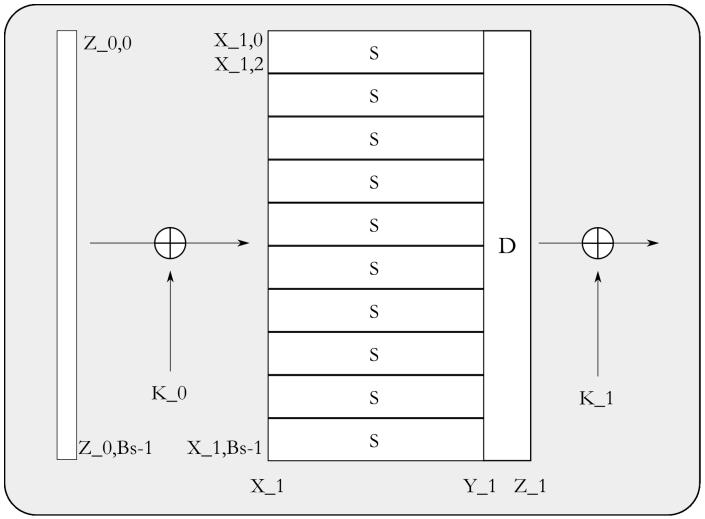
"Standard" Algorithms for Algebraic Attacks

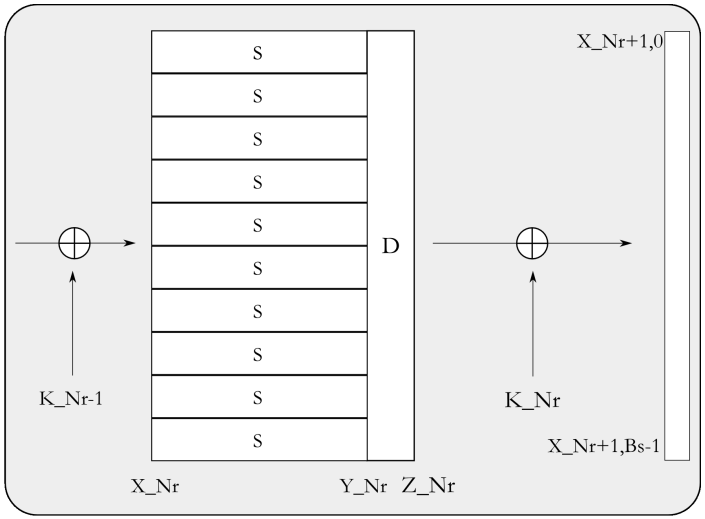
F_4(*)	Gröbner basis algorithm exploiting linear algebra ([Fau99])
F_5	Gröbner basis algorithm without reduction to zero ([Fau02])
SlimGB	Gröbner basis algorithm that keeps polynomials slim, not so interesting over finite fields ([Bri05])
XL Family(*)	multiply and reduce by linear algebra ([CKPS00], [CP02], [YCC04])
DR(*)	using Dixon Resultants ([TF05])
Zhuang-Zi	Creater bigger univariate polynomial and factor it ([DGS06])

Gröbner Basis without Reduction [BPW05]

- First Buchberger Criterion: *Suppose that we have $f, g \in G$, such that the leading monomials of f and g are pairwise prime. Then the S-polynomial of f and g reduces to zero.*
- We have n equations in n variables, so make sure each leading monomial is univariate and distinct from each other.
- We get a zero-dimensional Gröbner basis *CTCgb* for CTC ideals.
- Basis is still quadratic, but unclear how to exploit the fact that it is Gröbner basis; FGLM [FGLM93] and Gröbner Walk [CKM97] are too slow.

Meet-in-the-Middle [CMR05] Idea





Left ⟶ GB ⟵ Right

Meet-in-the-Middle Results

- Implemented on top of Singular's Gröbner basis engine
- Faster for *lex* monomial ordering than naïve approach
- Faster for *degrevlex* monomial ordering up to $B = 2$.

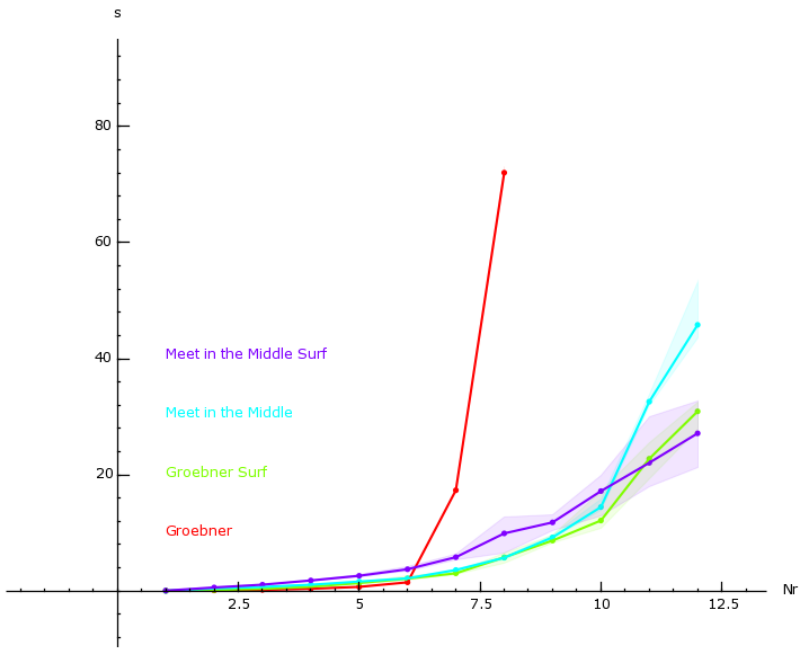
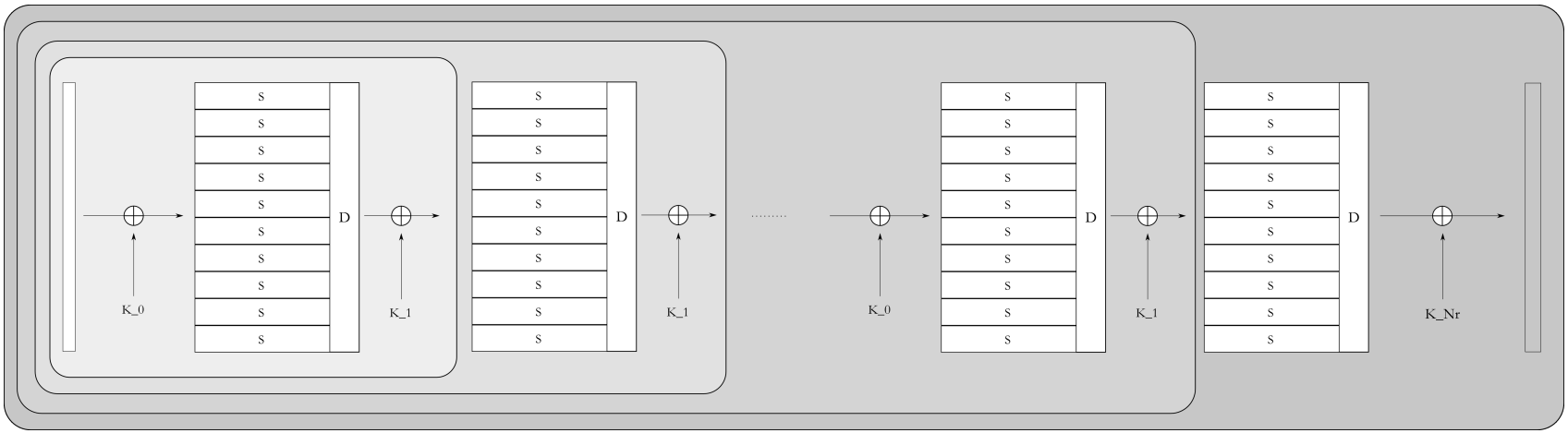


Figure: Runtimes for B=1 and term ordering *lex*

Gröbner Surfing: Idea



Gröbner Surfing: Algorithm

```
def groebner_surf(F):
    """
    Returns a Groebner basis for a given MQ problem F.

    INPUT:
        F -- MQ problem, separable in rounds

    OUTPUT:
        a Groebner basis for F with respect to F.ring().term_order()
    """
    singular.option("redSB")
    gb = singular(0,"ideal")
    R = F.ring()

    for i in range(len(F.round)):
        gb = (gb + singular(list(F.round[i]),"ideal")).std()

    return [R(e) for e in gb]
```

Gröbner Surfing: Correctness

Correctness Algorithm is in fact selection strategy. It doesn't affect correctness.

Termination Buchberger's Algorithm terminates, thus Nr times Buchberger's Algorithm terminate as well.

Gröbner Surfing: Good Term Orders

```
sage: F,s = ctc_MQ(Nr=3,variable_order=1,term_order=block_order(Nr=3))
sage: F.ring()._singular_()
//    characteristic : 2
//    number of vars : 39
//         block   1 : ordering dp
//                   : names    K_3,2 K_3,1 K_3,0 Z_3,2 Z_3,1 Z_3,0
//                              Y_3,2 Y_3,1 Y_3,0 X_3,2 X_3,1 X_3,0
//                              K_2,2 K_2,1 K_2,0
//         block   2 : ordering dp
//                   : names    Z_2,2 Z_2,1 Z_2,0 Y_2,2 Y_2,1 Y_2,0
//                              X_2,2 X_2,1 X_2,0 K_1,2 Kv1,1 K_1,0
//         block   3 : ordering dp
//                   : names    Z_1,2 Z_1,1 Z_1,0 Y_1,2 Y_1,1 Y_1,0
//                              X_1,2 X_1,1 X_1,0 K_0,2 K_0,1 K_0,0
//         block   4 : ordering C
```

Gröbner Surfing Results I

- Lexicographical Term Ordering
  - Faster than naïve approach and Meet-in-the-Middle for $B = 1$
- Graded Reverse Lexicographical & Block Term Ordering
  - Split equation systems in blocks by rounds, use *degrevlex* in blocks [Wei06]
  - Faster than naïve Buchberger for *degrevlex*.
  - Also: Gröbner basis looks better, as blocks eliminate.

Gröbner Surfing Results II

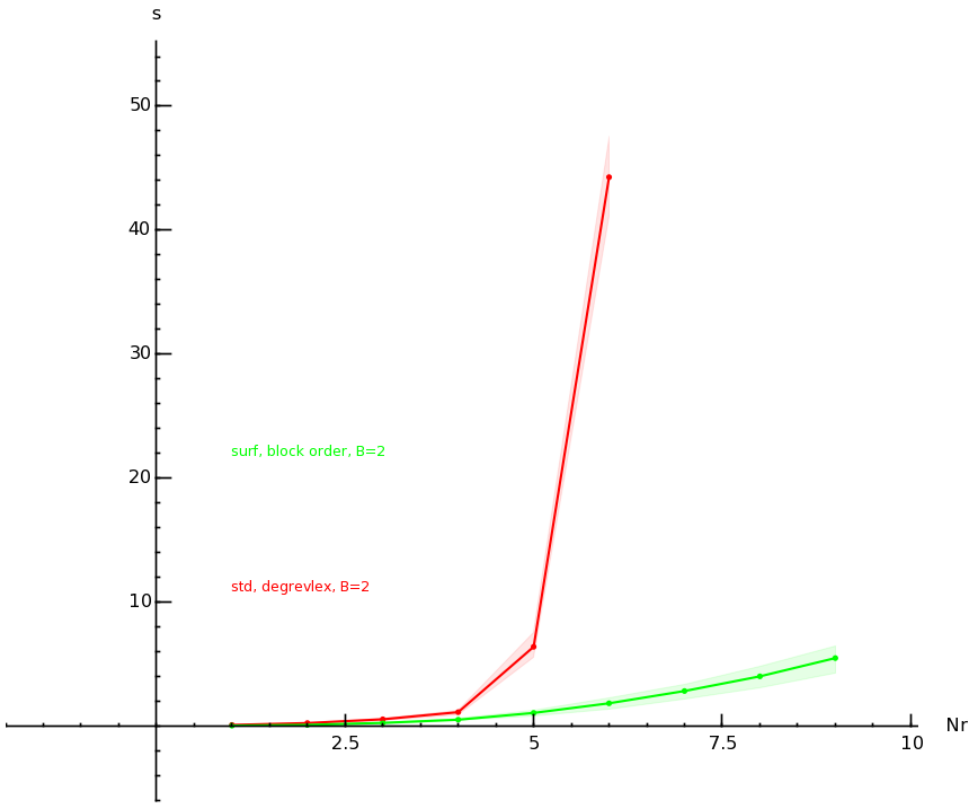


Figure: Runtimes for $B = 2$

Gröbner Surfing Results III

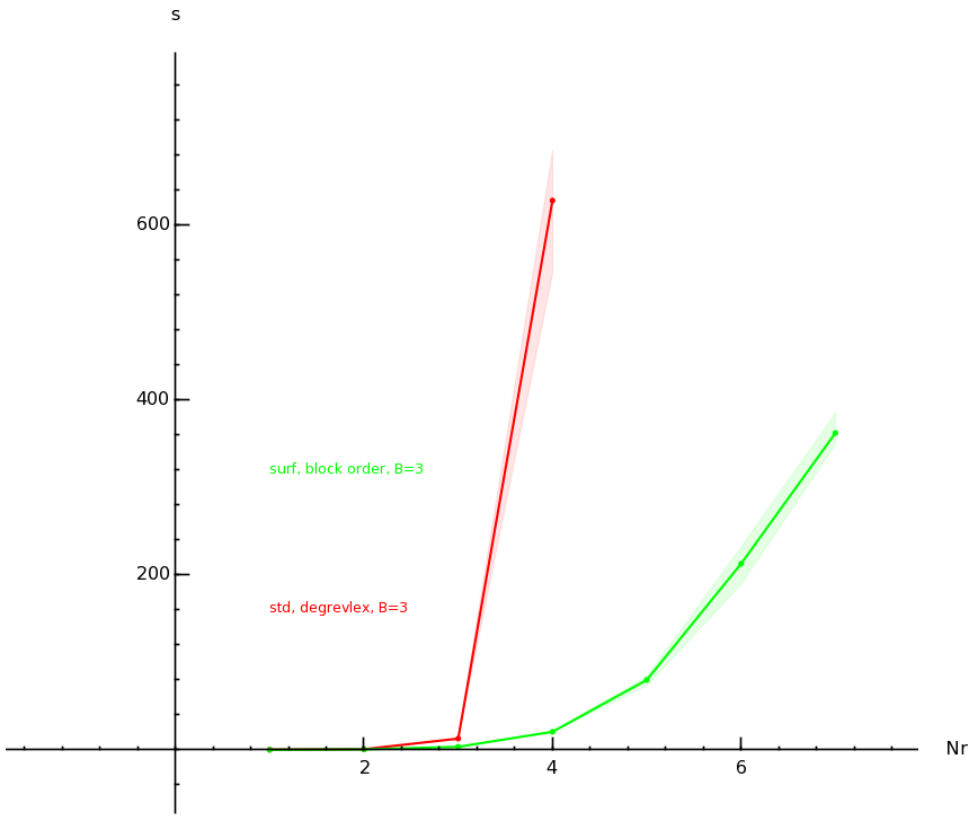


Figure: Runtimes for $B = 3$

Questions?

Thank you!

References I

Johannes Buchmann, Andrei Pychkine, and Ralf-Philipp Weinmann.

Block Ciphers Sensitive to Gröbner Basis Attacks.
Cryptology ePrint Archive, Report 2005/200, 2005.
available at: http://eprint.iacr.org/2005/200.

Michael Brickenstein.

Slimgb: Gröbner Bases with Slim Polynomials.
In *Reports On Computer Algebra 35*. Centre for Computer Algebra, University of Kaiserslautern, 2005.
available at: http://www.mathematik.uni-kl.de/~zca/Reports_on_ca/35/paper_35_full.ps.gz.

S. Collart, M. Kalkbrener, and D. Mall.

Converting Bases with the Gröbner Walk.
In *Journal of Symbolic Computation 24*, pages 465–469. Academic Press, 1997.

Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir.

Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations.
In *Proceedings of Eurocrypt 2000, LNCS 1807*, pages 392–407. Springer, 2000.

Carlos Cid, S. Murphy, and M. Robshaw.

Small Scale Variants of the AES.
In *Proceedings of Fast Software Encryption 2005, LNCS 3557*, pages 145–162. Springer, 2005.
available at http://www.isg.rhul.ac.uk/~sean/smallAES-fse05.pdf.

References II

Nicolas Courtois.

How Fast can be Algebraic Attacks on Block Ciphers?
Cryptology ePrint Archive, Report 2006/168, 2006.
available at: http://eprint.iacr.org/2006/168.pdf.

Nicolas Courtois and Josef Pieprzyk.

Cryptanalysis of Block Ciphers with Overdefined Systems of Equations.
Cryptology ePrint Archive, Report 2002/044, 2002.
available at http://eprint.iacr.org/2002/044.

Jintai Ding, Jason E. Gowe, and Dieter S. Schmidt.

Zhuang-Zi: A New Algorithm for Solving Multivariate Polynomial Equations over a Finite Field.
Cryptology ePrint Archive, Report 2006/38, 2006.
available at: http://eprint.iacr.org/2006/038.pdf.

Orr Dunkelman and Nathan Keller.

Linear Cryptanalysis of CTC.
Cryptology ePrint Archive, Report 2006/250, 2006.
available at: http://eprint.iacr.org/2006/250.pdf.

Jean-Charles Faugère.

A New Efficient algorithm for Computing Gröbner Basis (F4), 1999.
available at http://modular.ucsd.edu/129-05/refs/faugere_f4.pdf.

References III

Jean-Charles Faugère.

A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5).
In *Proceedings of ISSAC*, pages 75–83. ACM Press, 2002.

Jean-Charles Faugère, P. Gianno, P. Lazard, and T. Mora.

Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering.
In *Journal of Symbolic Computation 16*, pages 329–344. Academic Press, 1993.

Xijin Tang and Yong Feng.

A New Efficient Algorithm for Solving Systems of Multivariate Polynomial Equations.
Cryptology ePrint Archive, Report 2005/312, 2005.
available at `http://eprint.iacr.org/2005/312`.

Ralf-Philipp Weinmann.

Private communication, 12 2006.

Bo-Yin Yang, Jiun-Ming Chen, and Nicolas T. Courtois.

On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis.
In *Proceedings of Information and Communications Security; 6th International Conference 2004, LNCS 3269*, pages 401–413. Springer, 2004.