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

Does Open Source Matter for Math Research?

"You can read Sylow's Theorem and its proof in Huppert's book in the library [...] then you can use Sylow's Theorem for the rest of your life free of charge, but for many computer algebra systems license fees have to be paid regularly [...]. You press buttons and you get answers in the same way as you get the bright pictures from your television set but you cannot control how they were made in either case.

With this situation two of the **most basic rules of conduct in mathematics** are violated: In mathematics **information is passed on free of charge** and **everything is laid open for checking**. Not applying these rules to computer algebra systems that are made for mathematical research [...] means moving in a most undesirable direction. Most important: Can we expect somebody to believe a result of a program that he is not allowed to see? Moreover: Do we really want to charge colleagues in Moldava several years of their salary for a computer algebra system?" – J. Neubüser (1993) (he started GAP in 1986).

What is SAGE?

A Distribution of free open source math software. 75MB source tarball that builds self-contained.

New Readable Code that fill in gaps in functionality; implement new algorithms.

A Unified Mainstream Interface to math software: to Magma, Macaulay2, Singular, Maple, MATLAB, Mathematica, Axiom, etc.

SAGE was started in 2005 and is led by William Stein (UW, Seattle). Version 2.0 is due to next week focusing on speed improvements. Version 3.0 will include parallell computing support.

SAGE Components

Arithmetic	GMP, MPFR, Givaro
Commutative algebra	Singular (libcf, libfactory)
Cryptography	OpenSSL, PyOpenSSL, PyCrypto
Group theory and combinatorics	GAP
Graph theory	NetworkX
Number Theory	PARI, NTL
Numerical computation	GSL, Numpy
Calculus, Symbolic comp	Maxima
Specialized math	many C/C++ programs ...
Command Line	IPython
Graphical Interface	Notebook, jsmath, Moin wiki
Plotting	Matplotlib, Tachyon, libgd
Networking	Twisted
Database	ZODB, Python Pickles
Programming language	Python, SageX/Pyrex (compiled python)

All core components are free and open source (mostly GPL'd).

The SAGE Website

Website	`http://modular.math.washington.edu/sage`
Online SAGE notebook	`http://sage.math.washington.edu:8100`
Documentation	Tutorial, Install Guide, Programming Guide, Reference Manual, Constructions.
Platforms	OS X, Linux, and Windows (Cygwin).
Mailing Lists	sage-devel (hundreds of messages/month), sage-announce, sage-forum, sage-support.
Wiki	`http://sage.math.washington.edu:9001/`
Bug Tracker	`http://sage.math.washington.edu:9002/sage_trac`
IRC Chatroom	`#sage-dev` on irc.freenode.net

Why Should You Care About Yet Another CAS?

- It is strictly open-source (GPL).
- My thesis is implemented mainly in SAGE (and Singular), including F_4 over $\mathbb{F}_2$
- It is also a meta Computer Algebra System: keep using your "old" code.
- It is very powerful already.
  - it can do whatever its components can do
  - can use huge amount of available Pyhthon libraries (e.g. use PyGame to visualize your algorithm)
- Easily extendable (Python, C, whatever CAS you like)
- But: Basic low-level (e.g. addition) commutative algebra arithmetic very slow right now. Higher levels (e.g. Gröbner basis calculation) are reasonable: Singular.
  - basic multivariate polynomial arithmetic will improve over Summer, some of my thesis results will eventually become part of SAGE

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.

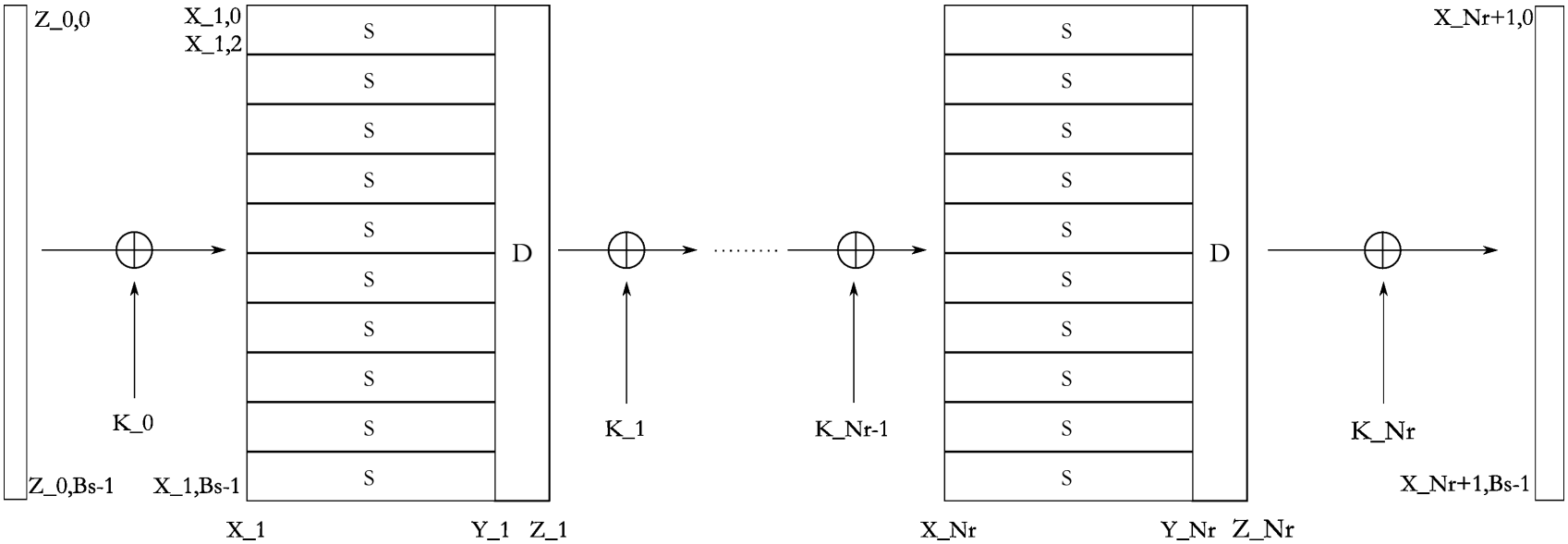
Why CTC?

- 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.
- This gives rise to 14 lineary independent quadratic equations in the input and output variables
- The diffusion layer is very thin:

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

- The key schedule is a simple permutation of wires.

$$P_0 + K_{0,0} + X_{1,0},$$
$$P_1 + K_{0,1} + X_{1,1},$$
$$P_2 + K_{0,2} + X_{1,2},$$

$$1 + X_{1,0} + X_{1,1} + X_{1,1}X_{1,0} + X_{1,2} + Y_{1,0},$$
$$1 + X_{1,1} + X_{1,2}X_{1,0} + Y_{1,1},$$
$$1 + X_{1,1} + Y_{1,0}X_{1,0} + Y_{1,1},$$
$$X_{1,2} + Y_{1,0} + Y_{1,1} + Y_{1,1}X_{1,0},$$
$$1 + X_{1,0} + X_{1,1} + X_{1,2}X_{1,1} + Y_{1,0} + Y_{1,1} + Y_{1,2},$$
$$1 + X_{1,0} + X_{1,1} + Y_{1,0} + Y_{1,0}X_{1,1} + Y_{1,1} + Y_{1,2},$$
$$X_{1,0} + Y_{1,1}X_{1,1} + Y_{1,2}X_{1,0},$$
$$1 + X_{1,1} + X_{1,2} + Y_{1,0} + Y_{1,2}X_{1,0} + Y_{1,2}X_{1,1},$$
$$Y_{1,0} + Y_{1,0}X_{1,2} + Y_{1,2} + Y_{1,2}X_{1,0},$$
$$X_{1,0} + X_{1,2} + Y_{1,0} + Y_{1,1}X_{1,2} + Y_{1,2},$$
$$1 + X_{1,0} + X_{1,1} + Y_{1,1} + Y_{1,2}X_{1,0} + Y_{1,2}X_{1,2},$$
$$X_{1,0} + Y_{1,1}Y_{1,0} + Y_{1,2},$$
$$1 + X_{1,0} + X_{1,1} + Y_{1,1} + Y_{1,2} + Y_{1,2}Y_{1,0},$$
$$X_{1,0} + X_{1,2} + Y_{1,0} + Y_{1,1} + Y_{1,2} + Y_{1,2}Y_{1,1},$$

$$Y_{1,0} + Y_{1,1} + Z_{1,0},$$
$$Y_{1,1} + Y_{1,2} + Z_{1,1},$$
$$Y_{1,0} + Z_{1,2},$$

$$K_{0,1} + K_{1,0},$$
$$K_{0,2} + K_{1,1},$$
$$K_{0,0} + K_{1,2},$$

$$Z_{1,0} + K_{1,0} + C_0,$$
$$Z_{1,1} + K_{1,1} + C_1,$$
$$Z_{1,2} + K_{1,2} + C_2,$$

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 ([Bri05])
XL Family(*)	multiply and reduce by linear algebra ([CKPS00], [CP02], [YCC04])
DR(*)	using Dixon Resultants ([TF05])
Zhuang-Zi	reduce to univariate case in extension field ([DGS06])

F_4 over $\mathbb{F}_2$

- Implementation is straight-forward as described in [Fau99]
- slightly optimized for $\mathbb{F}_2[x_1, \ldots, x_n]/ < x_1^2 + x_1, \ldots, x_n^2 + x_n >$
  - Monomials represented as bitsrings as in [Shi]
- slow in general, but for e.g. $CTC_{3,3,3}$:
  - Magma V2.11-2 GroebnerBasis() 18.42 s
  - F_4 26.14 s
  - Singular 3-0-2 std() 54.52 s
- Much room for improvements, some things are really dumb right now

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.
- Variable order: $K_{i,j} < X_{i,j} < Y_{i,j} < Z_{i,j}$ for $0 \leq i \leq Nr$ and $0 \leq j < Bs$
- Need to raise the degree of $Y_{i,j}$ and use only 3 S-box equations.
- 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.

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

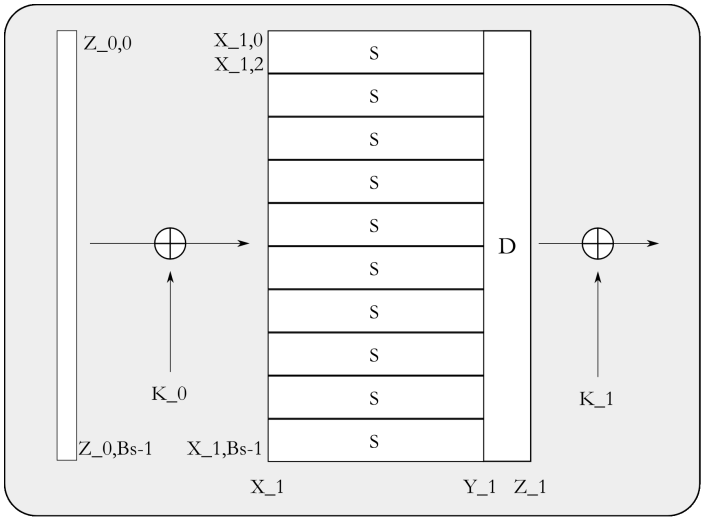
$$
\begin{aligned}
X_{1,2} + Y_{1,0}^2 + Y_{1,1} + Y_{1,1}X_{1,0}, \\
1 + X_{1,1} + X_{1,2}X_{1,0} + Y_{1,1}^2, \\
X_{1,0} + Y_{1,1}Y_{1,0} + Y_{1,2}^2
\end{aligned}
$$

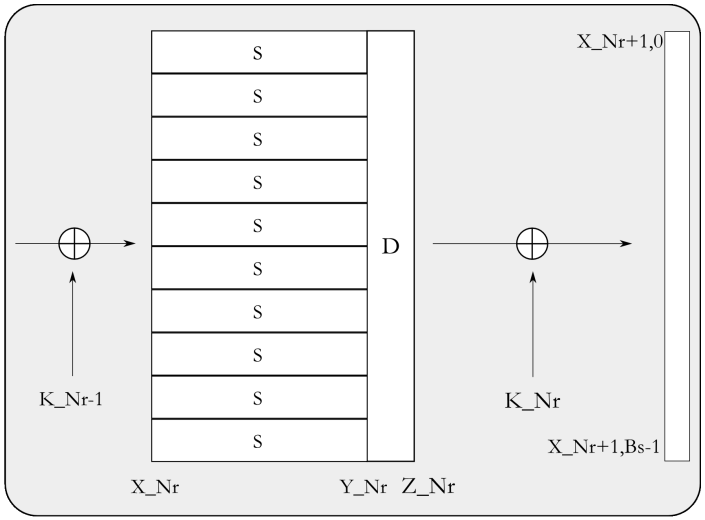
$$
\begin{aligned}
Y_{1,0} + Y_{1,1} + Z_{1,0}, \\
Y_{1,1} + Y_{1,2} + Z_{1,1}, \\
Y_{1,0} + Z_{1,2},
\end{aligned}
$$

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

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

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 $B = 2$.

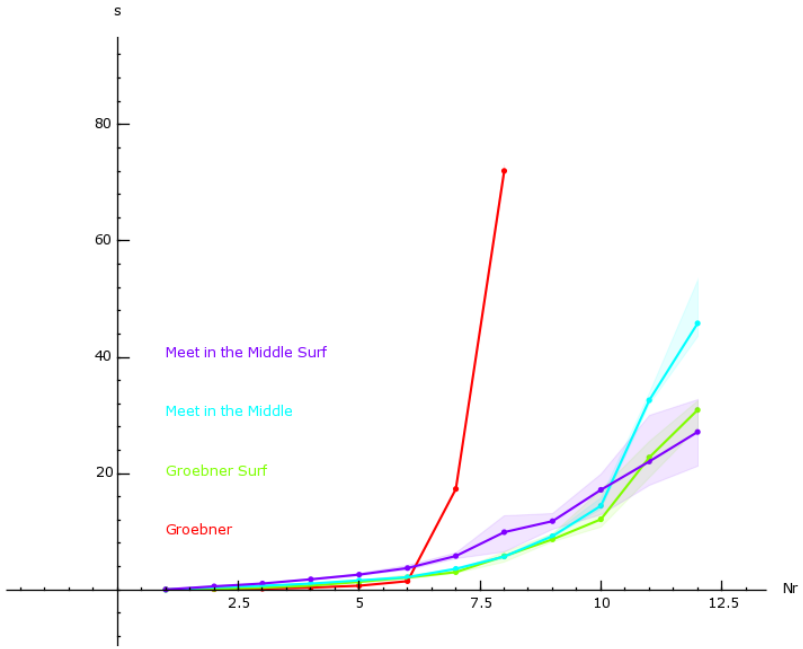
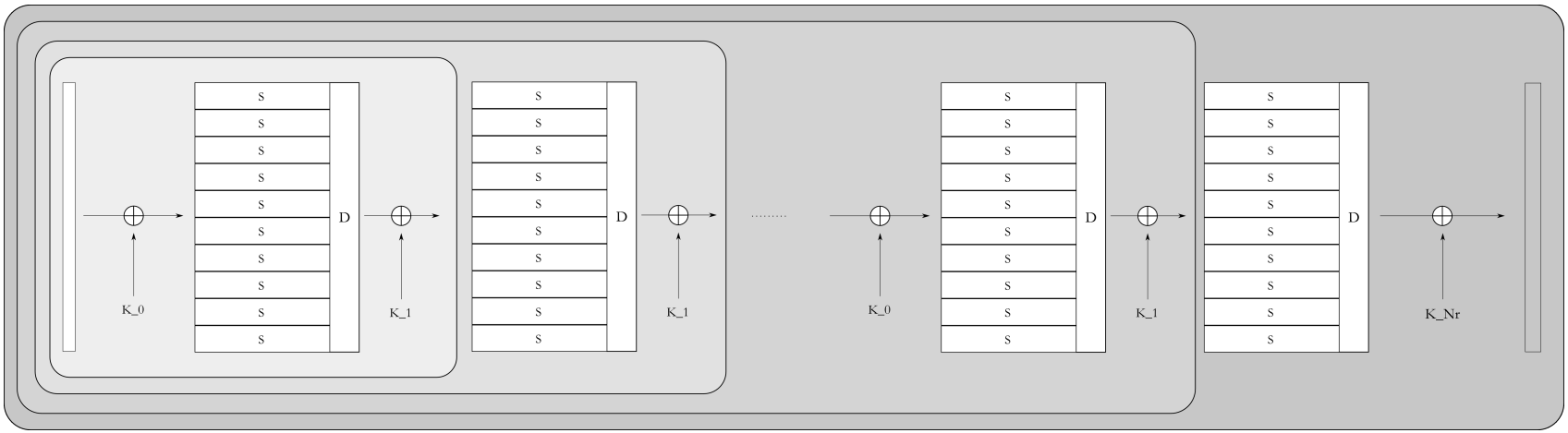


Figure: Runtimes for B=1 and term ordering *lex*

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 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.
- Why is this faster?
  - Exploits structure: We know the dependencies
  - Reduces intermediate basis grow: reduction after every round
  - Thus it is not faster for *degrevlex* in general

Gröbner Surfing Results II

A good monomial order:

```
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 III

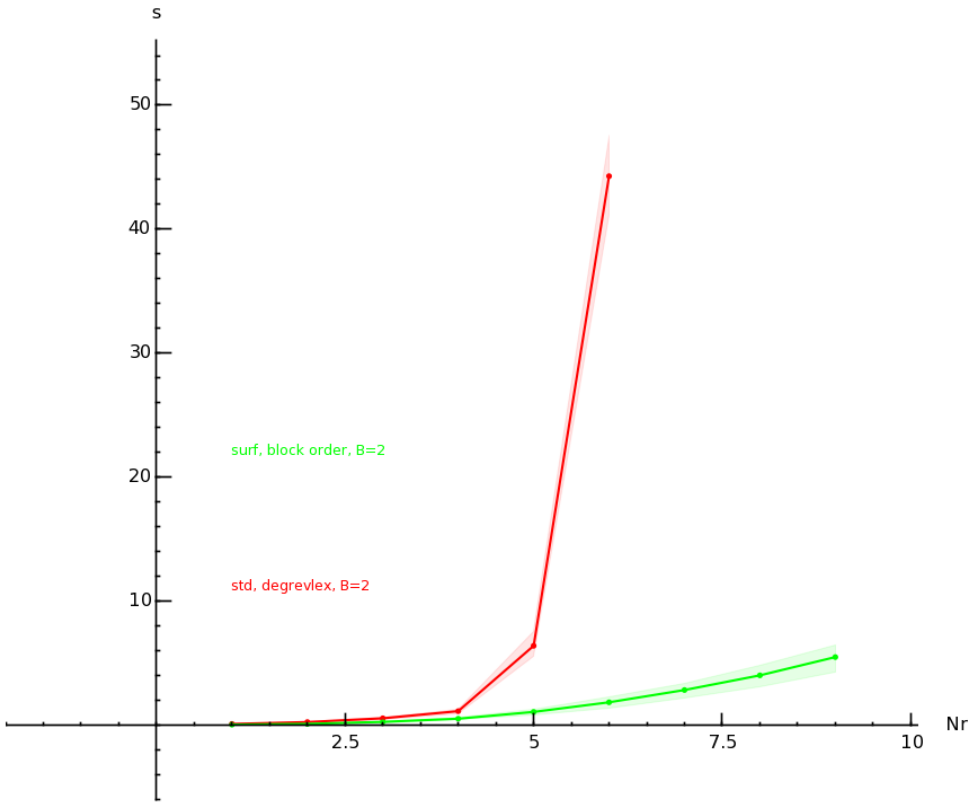


Figure: Runtimes for $B = 2$

Gröbner Surfing Results IV

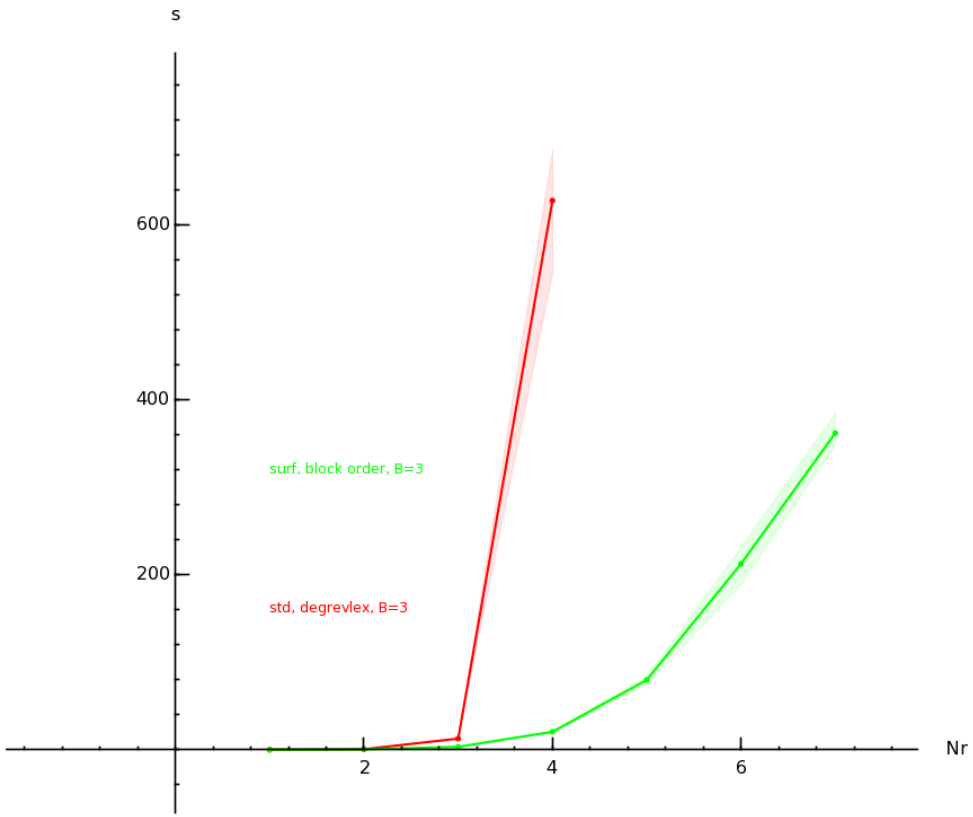


Figure: Runtimes for $B = 3$

Things to do

- Can an approach like "Gröbner Surfing" be applied to AES?
- Does it scale up?
- Tweak it: block borders, Gröbner basis algorithms, variable orders
- How to exploit the quasi parallelism of the S-boxes: Compute Gröbner basis for one round and $B = 85$
- Refine F_4 implementation

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.

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.

References II

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.

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.

References III

Mitsunari Shigeo.

Horatu / IPA-SMW.
available at: http://www.math.kobe-u.ac.jp/HOME/kimura/Makoto_Sugita.txt.

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.