Algebraic Cryptanalysis:

From Plug-and-Pray Experimental Approach to Constructive Optimization

Nicolas T. Courtois

University College London, UK

Topics

- AC, "Algebraization", I/O equations method.
  - challenges: AES[Courtois-Pieprzyk], ECDLP[Diem].
- Two Philosophies, 1.+2.

1. Classical approach: XL, XSL, predictions,
2. Algebraic coding and optimization. NEW!

Overdefined heuristics, phase transitions
=> how "Degree of regularity" can be reduced
with help of redundancy and oracles!

Key Question

- Can do better than just contemplate these transitions?
- Can we explicitly engineer phase transitions to happen?

Toy examples :

- ElimLin on Simon block cipher: the force of an asymptotic
- ECC Coding: how redundancy leads to explicit I/O equations => explicitly constructed degree falls.

Planet Earth A.D. 2016

Mafia Economy
Manufacture of Toxic Waste
Debt Slaves

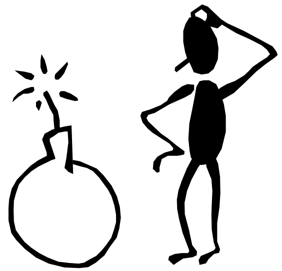
Solution

Travel to a Different Planet!

Which Planet?

1. A planet where a crypto currency is
at the centre of a more inclusive economy

2. A planet where quantum computers
break RSA, ECDLP etc…



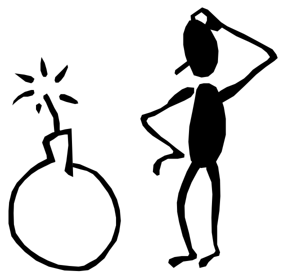
3. A planet where algebraic cryptanalysis
breaks AES, ECDLP etc…

الجبر

Which Planet?

1. A planet with a crypto currency

2. A planet full of quantum computers



3. A planet full of algebraic cryptanalysts

الجبر

=> all 3 planets have MORE jobs for crypto researchers…

!

Algebraic Cryptanalysis [Shannon]

Breaking a « good » cipher should require:

"as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type"

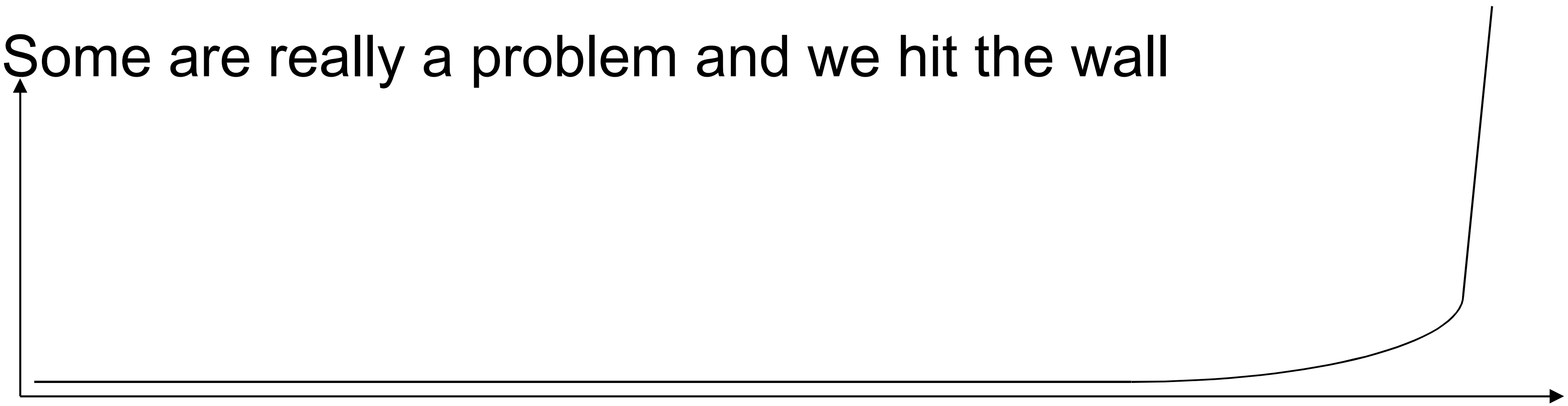
[Shannon, 1949]

This Talk

I will review some my research on Algebraic Cryptanalysis
in the last 15+ years and try to focus
on some "strategic" questions, key principles, big picture

False/Real difficulties.

- Some things are not necessarily a problem and CAN be solved/circumvented
- Some are really a problem and we hit the wall



False or Real Difficulty?

Not necessarily a problem:

1. Lack of algebraic structure, not clear how to even start doing any sort of “algebraic” attack.
2. Many rounds, large systems of equations with lots of variables.
3. NP hard problems – hard instances?

Real difficulties:

e.g. ECDLP=>
Semaev polys

1. Complexity grows exponentially – oops.
2. Bad equations topology / density / connectivity.
3. Mars vs. Venus problem – incompatible constraints.

Challenges (1)

AES, cf. Courtois-Pieprzyk attack

Challenges (2)

ECDLP, cf. PKC 2016 paper:

Algebraic approaches for the Elliptic Curve Discrete Logarithm Problem over prime fields

Christophe Petit[1], Michiel Kosters[2], and Ange Messeng[3]

Comment:

- Another ”plug-and-pray” attack which just does <u>not</u> work => bad “equations topology”

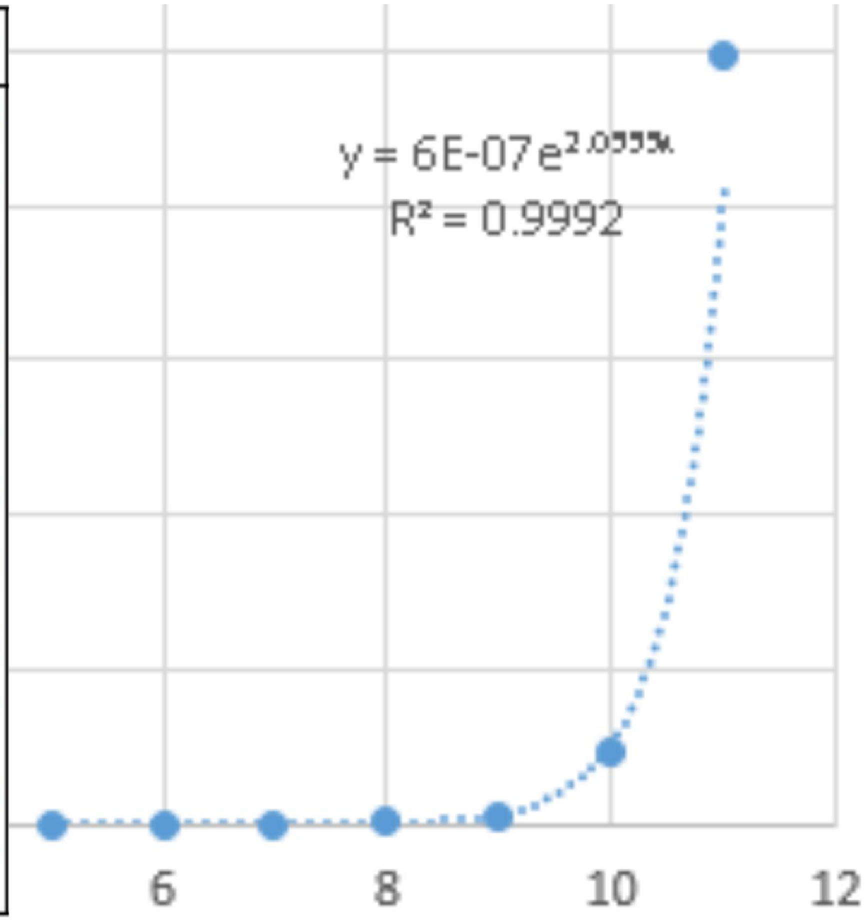
Challenges (2)

ECDLP, cf. PKC 2016 paper:

$O(e^{2.0*n})$

Table 4. Prime case, $p-1$ subgroups

n_1	p	D_{av}	D_{av}^{corr}	nbsols	Av. time (s)
1	5	4.00	4.00	0.59	0.00
2	17	4.00	4.00	0.79	0.00
3	73	4.00	4.00	0.84	0.00
4	257	4.01	4.00	1.14	0.00
5	1153	4.48	4.00	1.34	0.02
6	4289	5.00	5.00	1.08	0.13
7	17921	5.36	5.00	0.99	1.14
8	65537	5.36	5.00	0.96	9.09
9	262657	5.78	5.00	1.06	59.87
10	1051649	6.36	6.00	0.96	454.79
11	4206593	6.29	6.00	0.76	4975.07



Algebraization:

A Tool For Cryptanalysis

MQ Problem

Find a solution to a system of
m quadratic equations with
n variables
over a field/ring.

Cryptography and MQ

Claim: 50 % of all applied cryptography depends on the hardness of MQ.

For example:
RSA is based on MQ with $m=1$ and $n=1$:
factoring N ⇔ solving $x^2=C \bmod N$.

MQ Problem

Multivariate Version

[n variables]

Jean Dieudonné

[French Mathematician]

Book “Calcul infinitésimal”, Hermann, 1980

[..] Everybody in mathematics knows that going from one to several variables is an important jump that is accompanied by great difficulties and calls for completely new methods. […]

MQ Problem over GF(2)

Find a solution (at least one),

i.e. find $(x_0, \ldots, x_{n-1})$ such that:

$$\begin{cases} 1 = x_1 + x_0x_1 + x_0x_2 + \ldots \\ 0 = x_1x_2 + x_0x_3 + x_7 + \ldots \\ \vdots \end{cases}$$

Dense MQ

Dense MQ is VERY hard.   Best attacks ≈ $2^{0.8765n}$

Also a good candidate for PQ crypto.

=> Allows to build a provably secure stream cipher based on MQ directly!

C. Berbain, H. Gilbert, and J. Patarin:

QUAD: A Practical Stream Cipher with Provable Security, Eurocrypt 2005

- open problem: design a provably secure block cipher…

Schneier [Applied Cryptography book]

[…] Any algorithm that gets its security from the composition of polynomials over a finite field should be looked upon with scepticism, if not outright suspicion. […]

- Actually any cipher e.g. AES can be seen in this way… Including provably secure QUAD.
- ECDLP is also 'based' on hardness of solving polynomials over finite fields.
  - Igor Semaev: Summation polynomials and the discrete logarithm problem on elliptic curves, eprint 2004/031.

Algebraization:

Theorem:

Every function over finite fields is a polynomial function.

[can be proven as a corollary of Lagrange's interpolation formula] $P(X) = \sum_{i=1}^{t} Y_i \cdot \prod_{1 \le j \le t, j \ne i} \frac{X - X_j}{X_i - X_j}$

False over rings!

Better Method: I / O Degree:

Consider function $f : GF(2)^n \to GF(2)^m$,
$f(x) = y$, with $x = (x_0, \ldots, x_{n-1})$, $y = (y_0, \ldots, y_{m-1})$.

Definition [The I/O degree] The I/O degree of f is the smallest degree of the algebraic relation

$$g(x_0, \ldots, x_{n-1}; y_0, \ldots, y_{m-1}) = 0$$

that holds with certainty for every couple (x, y) such that $y = f(x)$.

These can be used directly for algebraic coding.

[sometimes more equations are needed see the notion of “describing degree”]

Two Philosophies

Two major ways to approach the general problem of solving large system of non-linear polynomial/algebraic equations.

1. Either we expand the number of monomials.
   – work in polynomial ideals, XL F5 etc…
2. Or we expand the number of variables
   – MC-efficient coding
   – algebraic coding

Two Philosophies

In both case we have two quantities:

R = number of equations

T = number of monomials

Main idea: R grows FASTER than T.

Two Philosophies

Two major ways to approach the general problem of solving large system of non-linear polynomial/algebraic equations.

1. Either we expand the number of monomials.
   – work in polynomial ideals, XL F5 etc…
2. Or we expand the number of variables
   – MC-efficient coding
   – algebraic coding

NEW!

???

previous research tends to show that this is a "bad idea".
E.g. XL is preferred to Linearization [Eurocrypt 2000].

In a Way we also have

Two major ways to approach the general problem of solving large system of non-linear polynomial/algebraic equations.

1. Plug-and-pray ☹
   – Build experiments, maybe it works?
2. More constructive!
   – More freedom for the attacker
   – Algebraic optimization problems

Find an “economical” way to expand the problem with redundancy so that the “degree of regularity” decreases the most

Philosophy 2 Is Not Stupid

Two major ways to approach the general problem of solving large system of non-linear polynomial/algebraic equations.

1. Either we expand the number of monomials.
   – work in polynomial ideals, XL F5 etc…
2. Or we expand the number of variables
   – MC-efficient coding
   – algebraic coding

Glossary

- MC = Multiplicative Complexity, informally counting the number of multiplications in algorithms
  - trying to do it with less

- REMARK:
  AES and Simon have <u>INCREDIBLY low MC</u>.

$x \to x^{-1}$ n=4 [Boyar and Peralta 2008-9]

eprint.iacr.org/2009/191/

5x

$t_1 = x_1 + x_2$	$t_2 = x_1 \times x_3$	$t_3 = x_4 + t_2$
$t_4 = t_1 \times t_3$	$y_4 = x_2 + t_4 \quad (*)$	$t_5 = x_3 + x_4$
$t_6 = x_2 + t_2$	$t_7 = t_6 \times t_5$	$y_2 = x_4 + t_7 \quad (*)$
$t_8 = x_3 + y_2$	$t_9 = t_3 + y_2$	$t_{10} = x_4 \times t_9$
$y_1 = t_{10} + t_8 \quad (*)$	$t_{11} = t_3 + t_{10}$	$t_{12} = y_4 \times t_{11}$
$y_3 = t_{12} + t_1 \quad (*)$		

Fig. 1. Inversion in $GF(2^4)$.

5 AND 11 XOR

Bit-Slice Gate Complexity

PRESENT S-box

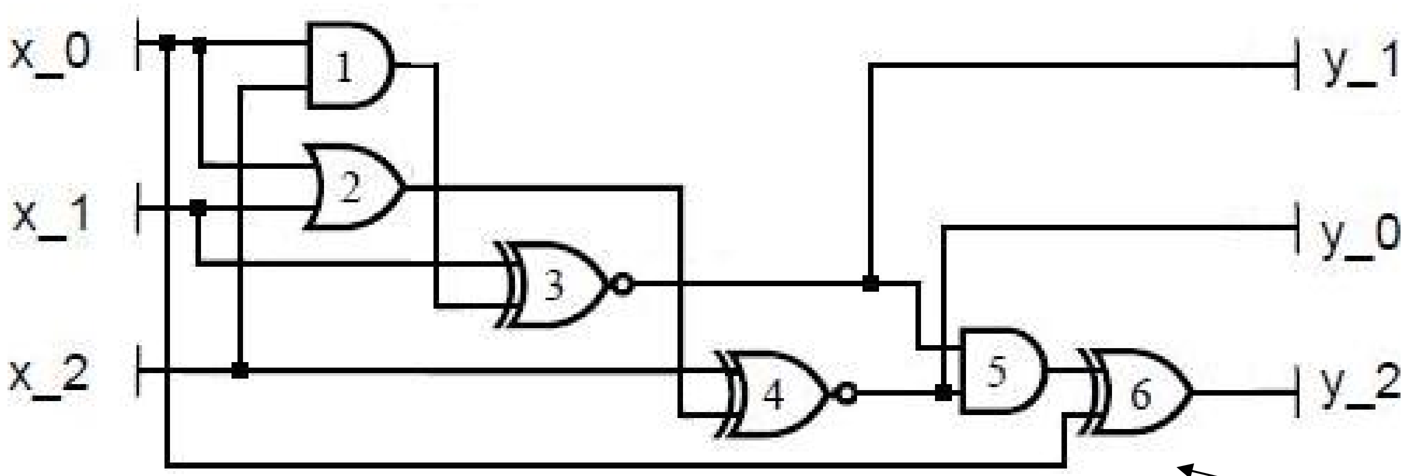
- Naïve implementation = 39 gates
- Logic Friday [Berkeley] = 25 gates
- Our result = 14 gates.


```
T1=X2^X1; T2=X1&T1; T3=X0^T2; Y3=X3^T3; T2=T1&T3; T1^=Y3; T2^=X1;
T4=X3|T2; Y2=T1^T4; T2^=~X3; Y0=Y2^T2; T2|=T1; Y1=T3^T2;
```

Fig. 1. Our implementation of the PRESENT S-box with only 14 gates

Another S-box – CTC2

Our new design:



NEW!

PROVEN OPTIMAL

Best Paper!

IARIA

International Academy, Research, and Industry Association

BEST PAPER AWARD

Multiplicative Complexity and Solving Generalized Brent Equations
With SAT Solvers

By

Nicolas Courtois, Daniel Hulme, Theodosis Mourouzis

Presented during COMPUTATION TOOLS 2012, The Third International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, held in Nice, France - July 22-27, 2012

IARIA Board

GOST Attacks with a SAT Solver

eprint.iacr.org/2011/626

Rounds	4	8		8		8		8	8
Key size	128	256							
Data	2 KP	2 KP		3 KP		4 KP		6 KP	$\approx$ 600 KP
See	Fact 3	[52]	Fact 5	[35]	Fact 6	[35]	Fact 7	Fact 127	Fact 10
cf.	page 13	Fact 15 page 40			page 26		page 26	page 207	page 35
cf. also	[97]				Fact 16		Fact 13		
Memory bytes	small	$\mathbf{2^{43}}$	2^{46}	2^{68}	small	2^{69}	small		small
Time	$\mathbf{2^{24}}$	2^{128}	$\mathbf{2^{127}}$	$\mathbf{2^{107}}$	$\mathbf{2^{110}}$	2^{94}	$\mathbf{2^{94}}$	$\mathbf{2^{83}}$	$\mathbf{2^{50}}$

Table 1. Principal attacks on 8 rounds of GOST with 2,3,4 and more KP

Two Philosophies

In both case we have two quantities:

R = number of equations

T = number of monomials

Main idea: R grows FASTER than T.

Which is simultaneously real and “impossible”.

More precisely let

- F = number linearly independent equations.
- F cannot grow faster than T, but R can.

The saturation when linear dependencies do appear because they have to is frequently what we look for.

The principle of XL:

Multiply the initial equations by low-degree monomials:

$$1 = x_5 + x_0 x_1 + x_0 x_2$$

becomes:

$$x_1 \cdot 1 = x_1 \cdot (x_5 + x_0 x_1 + x_0 x_2)$$

(degreee 3 now).

How XL works:

Initial system: m equations and $n^2/2$ terms.

Multiply each equation by
a product of any D-2 variables:

- Equations $R = m \cdot \binom{n}{D-2}$
- Terms $T = \binom{n}{D}$

Idea: One term can be obtained in many different ways,
T grows slower than R.

Necessary condition: R/T > 1
gives $m \cdot \binom{n}{D-2} / \binom{n}{D} > 1$ and thus $D \approx n/\sqrt{m}$

If sufficient, the complexity of XL would be about

$$T^\omega = \binom{n}{n/\sqrt{m}}^\omega$$

Sub-exponential !?!

XL works quite well

n	10	10	10	10	10	20	20	20	20	20	64	64
m	10	14	16	17	18	20	40	50	60	65	512	1024
D	3	3	3	3	3	3	3	3	3	3	3	3
R	110	154	176	187	198	420	840	1050	1260	1365	33280	66560
T	176	176	176	176	176	1351	1351	1351	1351	1351	43745	43745
Free	110	154	174	175	175	420	840	1050	1260	1350	33280	43744

Figure 1: XL simulations for $D = 3$.

n number of variables.

m number of equations.

D we generate equations of total degree $\leq D$ in the x_i.

R number of equations generated (independent or not). $R = m \cdot \binom{n}{D-2}$

T number of monomials of degree $\leq D$ $T = \binom{n}{D}$

Free number of linearly independent equations among the R equations.

◇ **XL will work when** $Free \geq T - D$.

The behaviour of XL

It is possible to predict the exact number of linearly independent equations in XL.

D	$Free$
3	$Min(T,R)$
4	$Min\left(T,R-\binom{m}{2}-m\right)$
5	$Min\left(T,R-(n+1)\binom{m}{2}-(n+1)m\right)$
6	$Min\left(T,R-\left[\binom{n}{2}+\binom{n}{1}+\binom{n}{0}\right]\cdot\left[\binom{m}{2}+\binom{m}{1}\right]+\binom{m}{3}+m^2\right)$

And “XSL”

“XSL is not an attack, it is a dream“

Vincent Rijmen, AES designer

The XL idea:

Multiplying the equations

by one or several variables.

The XSL variant:

Multiplying the equations

by one or several monomials (out of monomials present).

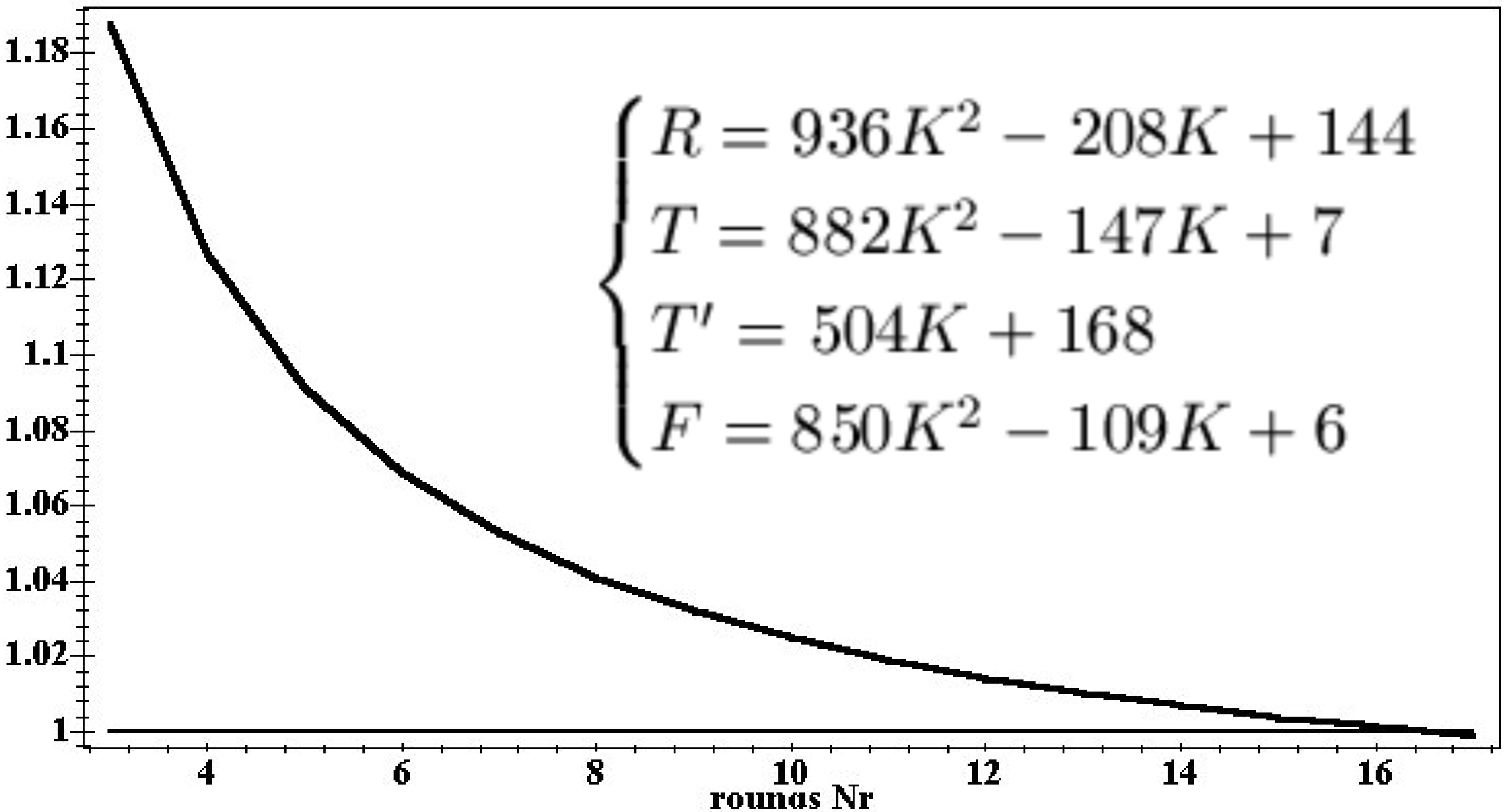
XL and XSL

Both work well, they operate a specific phase transition.

The curve reaches another curve and stays there.

n	24	24	24
m	16	27	32
D	5	5	5
R	37200	62775	74400
T	55455	55455	55455
F	33800	53325	55454

Simulations on a “Toy Cipher”



Free/(T-T’) - XSL works for up to 16 rounds. !

Three Stages

Algebraic attacks on block ciphers work in 3 stages:

1. Write good equations – overdefined, sparse or both.

2. Expand - to obtain a very overdefined system.

3. Solve at saturation / phase transition point.

Reinvented in 2006

Algebraic attacks on block ciphers today:

1. Write good equations – overdefined, sparse or both.

2. Expand - avoid / minimise impact of… 

3. Final "in place" deduction / inference / elimination method.
   - ElimLin alone and T' method. Very powerful.

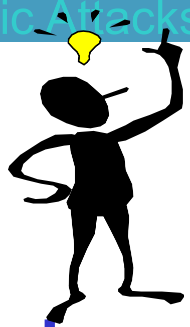
Reinvent it in 2016

Algebraic attacks on block ciphers today:

1. Write EVEN BETTER equations –
   => even more overdefined $R/T \approx 1$. NEW!
   => redundant "algebraic codes" NEW!
   => work on equations topology/density. NEW!

2. Expand - avoid / minimise impact of…

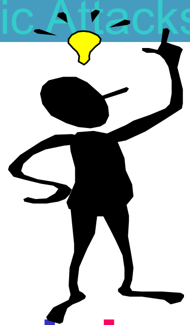
3. Final "in place" deduction / inference / elimination method.
   - ElimLin alone and T' method. Very powerful.



Part 1.

Find good equations: such that:

$$\frac{R}{T} = 1/4 \text{ or so..}$$



Can do Better?

Find better equations: such that:

$$\frac{R}{T} \approx 1 \text{ already}$$

+questions of equations density and topology

The Redundancy+Oracle idea:

We can decrease the “regularity degree” by adding variables AND new facts coming from an oracle.

The Redundancy+Oracle idea:

Example 1: ElimLin.

Oracle=encryption oracle.

Example 2: EC point splitting.

Oracle=block box EC point addition.

A Thought Experiment

EC point splitting.

$$P1 + P2 = Q$$

+ extra equations to code a “factor basis”.

More Overdefined

Same point splitting Pb.

$$\begin{cases} P1 + P2 = Q \\ P1 + D = P1' \\ P2 + E = P2' \\ P1' + P2' = (Q + D + E) \end{cases}$$

oracle

- added 2 constants D,E
- 2 new vars
  - linear ECC code expansion of vars
- 3 new eqs!

+ same x extra equations to code a “factor basis”.

- strict improvement: 2+x → 3+x

The Same Happens in ElimLin

By "magic" the regularity degree decreases with K

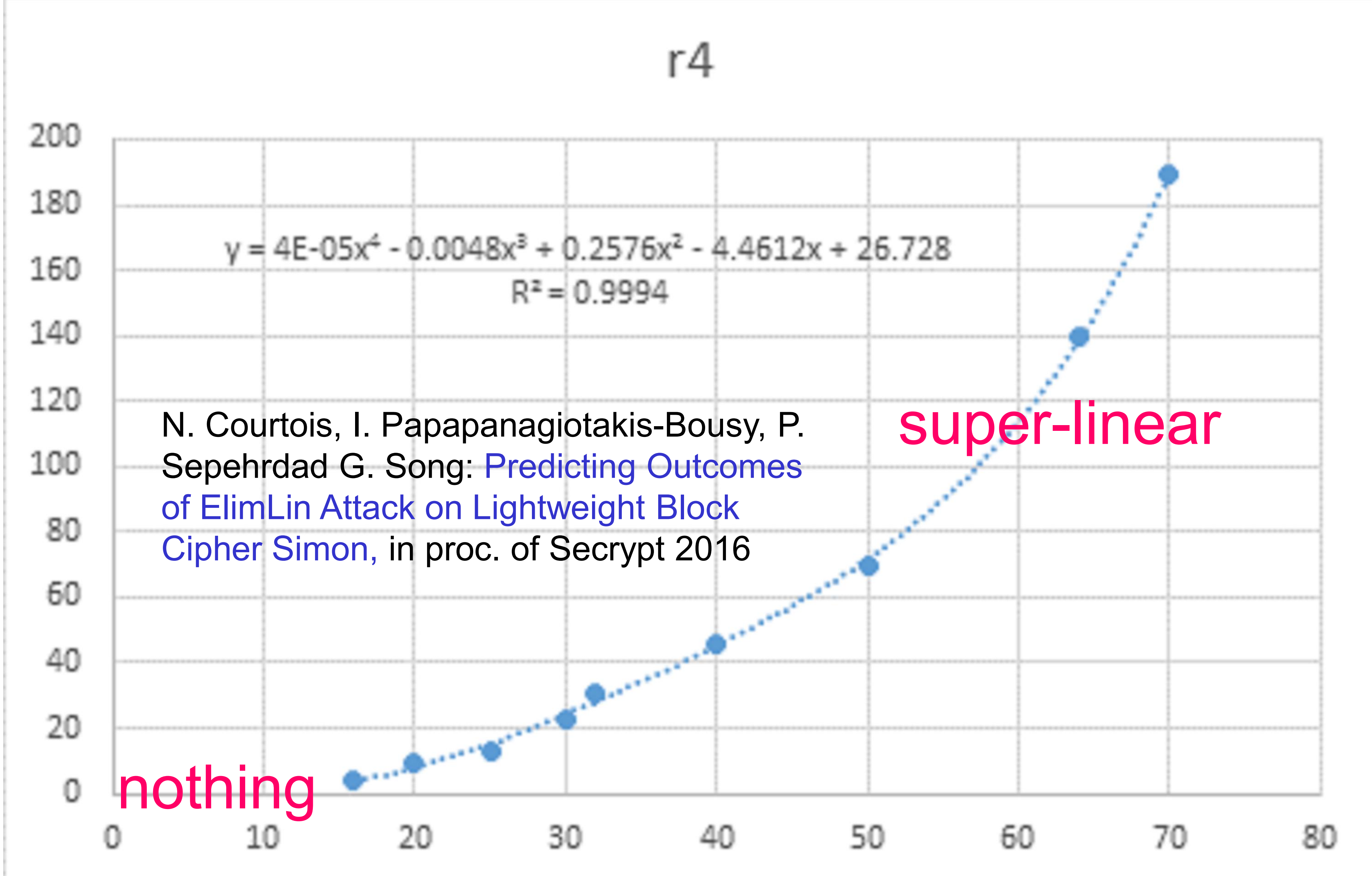
K= data complexity (K KP or K CP).

Asymptotic Aspects

Something VERY disturbing happens in ElimLin.

How quickly R or/and F grow when K increases?

8 Rounds of Simon 64/128



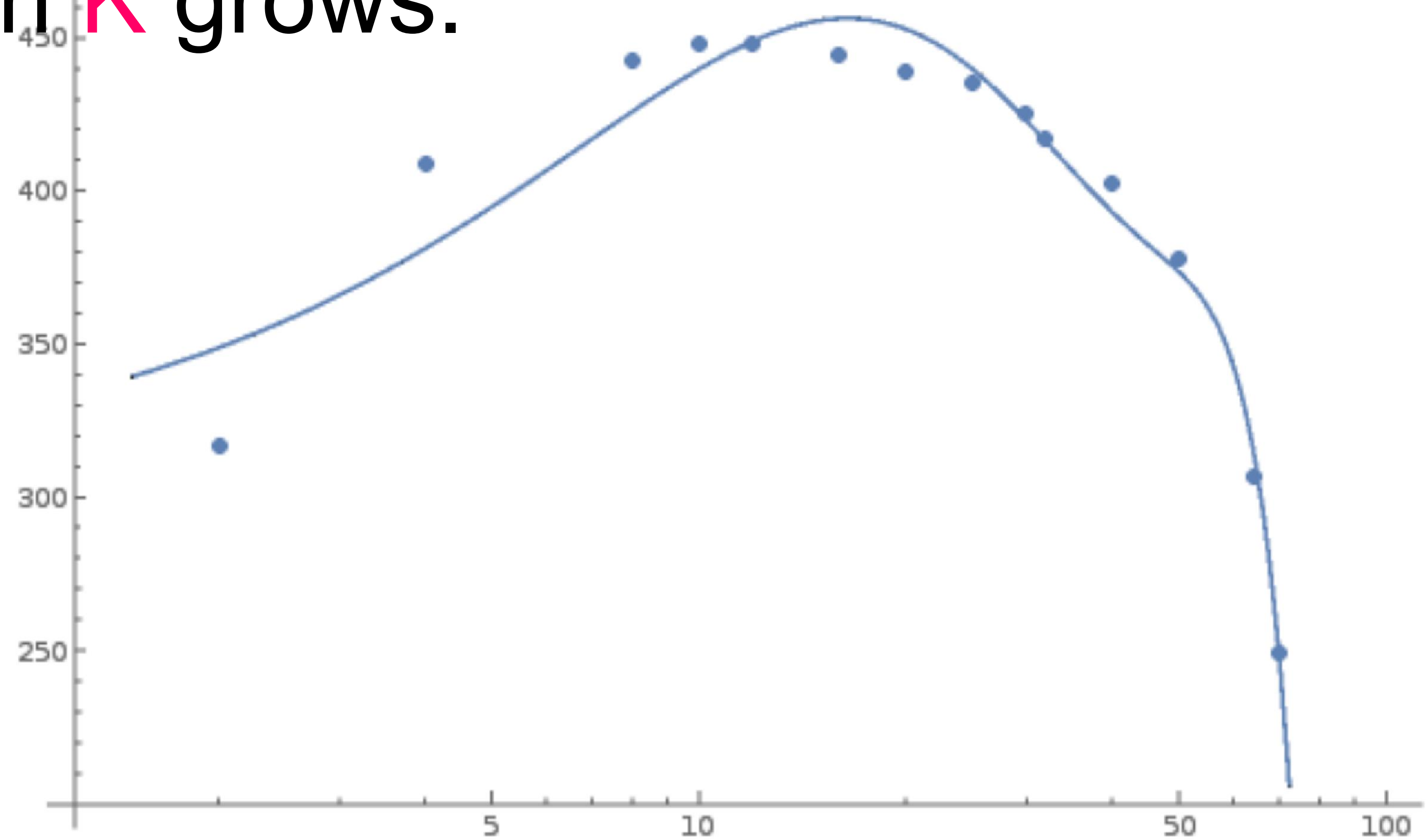
The Impossible Happens

Remark: In the long run it CANNOT be super-linear. $\rightarrow$linear when K$\rightarrow\infty$

However in the long run the cipher is broken for a fixed value of K.

Redundancy Up and Downs

#Variables at the end of ElimLin when K grows.



Back to EC Point Splitting Questions

Can we also produce a system of equations with fast growth due to redundant ECC coding?

$$P1 + P2 = Q$$

+ extra equations to code a “factor basis”.

Elaborate Prototype [eprint 2016/704]

$$P1 + P2 = Q$$

+ a very new unique method to code a "factor basis"

simulations with 9390489 incremental simulator $K=4$								
K value		4						
$K1$ value		3						
$K2$ value		3						
/maxorder value		6						
K' value	predictor	4	6	8	10	12	14	16
#vars	$2(1+K(K'-1))$	26	42	58	74	90	106	122
$F_{1+0}^{J=1}(K) =$	$2K(0.5K'^2 - 1.5K' + 1)$	24	80	168	288	440	624	840
order 1	$2K \cdot FQ$	24	80	168	288	440	624	840
$F_{2+0}^{J=2}(K) =$	$K(K-1)(K'^2-6K'+9)$	12	108	300	588	972	1452	2028
order $2+0$	$F_{2+0}^{J=2}(K)$	12	108	300	588	972	1452	2028
$F_{1+1}^{J=2}(K) =$	$K^2(K'^2 - 12K' + 22)$	–	–	–	32	352	800	1376
order $1+1$	$F_{1+1}^{J=2}(K)$	0	0	0	32	352	800	1376
order 2		12	108	300	606	1324	2252	3404
predicted $J \leq 2$	$\sum$	36	188	468	908	1764	2876	4244
actual ≤ 2		36	188	468	894	1764	2876	4244
$F_{2+1}^{J=3}(K) =$	$(2K^2/2-2K)(K'-2)$	–	–	–	192	240	288	336
order $2+1$		0	0	0	192	240	288	336
order $3+0$		0	0	0	0	0	0	0
order 3		0	0	0	192	240	288	336
order $4-6$		0	0	24	0	0	0	0
F total order 1-6		36	188	492	1100	2004	3164	4580
$T2$		352	921	1712	2776	4096	5672	7504
F/T2		0.10	0.20	0.29	0.40	0.49	0.55	0.61

Are We Making Any Progress?

Possibly this approach is stupid and NOT as good as traditional highly-optimized Gröbner basis approach.

=> Everybody uses Semaev polynomials + ”plug-and-pray” GB.

$$S_3(x_1,x_2,x_3) = (x_1-x_2)^2x_3^2-2\left[(x_1+x_2)(x_1x_2+A)+2B\right]x_3+(x_1x_2-A)^2-4B(x_1+x_2)$$

Are We Making Any Progress?

Possibly this approach is stupid and NOT as good as traditional highly-optimized Gröbner basis approach.

=> Even if so, I believe this approach is BETTER because we avoid ”plug-and-pray” and construct our degree falls and other equations more explicitly. More control/insights on what we do.

Merits of Redundancy

Linear ECC Code expansion

=>

NEW very regular families of I/O equations which we can construct explicitly

Example: New ECC I/O relations

D73 Theorem [Courtois 2016]

Theorem 4.2.1 (D73 Theorem). We consider the following set of variables on EC, a special form of ECC Code with 3 inputs and 7 outputs for any Weierstrass elliptic curve modulo a large P.

$$(P1, P2, P3) \mapsto \begin{array}{ccc} P1 & P2 & P1+P2 \\ P1+P3 & P2+P3 & P1+P2+P3 \\ P3 & & \end{array}$$

If all the 7 points are distinct from the ECC neutral element ∞ we have:

```
sx1*sx2*(sx23-sx13) +sx1*sx3*(sx12-sx23) +sx2*sx3*(sx13-sx12)
+sx123[sx1*(sx13-sx12)+sx2*(sx12-sx23)+sx3*(sx23-sx13)] = 0
```