Multiplicative Complexity

Nicolas T. Courtois

University College London, UK

Definition [informal]

- Every function can be represented as a number of multiplications + linear functions over a finite field/ring.
- We call MC (Multiplicative Complexity) the minimum number of multiplications needed.

MC is one of the most important PRACTICAL and theoretical Problems in Computer Science. Why? Answer in these slides.

Roadmap

- bi-linear and tri-linear problems such as complex / matrix multiplication
- general case
  - arbitrary vectorial Boolean functions
    - in cryptography called S-boxes
- some prominent cipher systems
    - and their algebraic vulnerabilities

Glossary

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

1805

Gauss in 1805

multiplying two complex numbers:

- naïve method

4x

$$(a + bi) \cdot (c + di) = (ac-bd) + (bc+da)i$$

- Gauss method:

$$P1 = c(a+b)$$
$$P2 = a(d-c)$$
$$P3 = b(c+d)$$

3x

$$(a + bi) \cdot (c + di) = (P1-P3) + (P1+P2)i$$

MM = Matrix Multiplication

- entry size = n^2
- naïve algorithm = n^3
- amazingly enough, many computer scientists believe it could be nearly quadratic…
  - like = n^2 (log n)^{sth}
  - which in fact would be linear!
    - this is in the input size = n^2
- there is a proven lower bound of n^2*log n [Raz 2002]

MM = “Meta-Algorithm”?

Representation Theory:

any finite group will be seen as matrices of certain particular form, matrix multiplication will be used to compute in the group.

Sort of magical trick to “compute” things unrelated to matrices.

Equivalence of MM and Other Problems

A speed up in MM will automatically result in a speed improvement of many other algorithms:

- Gauss: solving linear equations
- solving of non-linear polynomial equations…
- transitive closure of a graph or a relation on a finite set
- recognising if a word of length n belongs to a context-free language
- many many other…

$$$ Importance of MM

- <u>At least</u>
  Hundreds of Megawatts * Years
  are spent in linear algebra operations
  - Code breaking by intelligence agencies
  - Google page ranking
  - Computer graphics x millions of GPU chips
  - Scientific computations
  - Etc.

Best Known Exponents

- $O(n^{2.3755})$ obtained in 1987 by Coppersmith-Winograd, best known until now!
- June 2010:
  Andrew Stothers obtained $n^{2.3737}$
- 2011: beaten by Virginia Vassilevska Williams [Berkeley] who obtained $n^{2.3727}$

could we join the race???

James Eve [Newcastle Uni. UK, 2008]

"I am very confident that I have found the right approach and that what I have done has cracked or is very close to cracking the problem of efficient algorithms for multiplying and inverting matrices"

Which would be n^2 (log n) ?? or similar.

Donald Knuth has been reviewing his paper in 2008 and asked questions. James Eve died in 2008 before he could answer these questions…

Improving MM

Naïve = n^3

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

Bi-Linear Non-Commutative Algorithm

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

8x

Strassen [1969]

$$\begin{aligned}
\mathbf{M}_1 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2}) \\
\mathbf{M}_2 &:= (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1} \\
\mathbf{M}_3 &:= \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2}) \\
\mathbf{M}_4 &:= \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1}) \\
\mathbf{M}_5 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2} \\
\mathbf{M}_6 &:= (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2}) \\
\mathbf{M}_7 &:= (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2})
\end{aligned}$$

another bi-linear method

7x

$$\begin{aligned}
\mathbf{C}_{1,1} &= \mathbf{M}_1 + \mathbf{M}_4 - \mathbf{M}_5 + \mathbf{M}_7 \\
\mathbf{C}_{1,2} &= \mathbf{M}_3 + \mathbf{M}_5 \\
\mathbf{C}_{2,1} &= \mathbf{M}_2 + \mathbf{M}_4 \\
\mathbf{C}_{2,2} &= \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_6
\end{aligned}$$

Lower Complexity

- Trading multiplications (expensive) for additions (much cheaper)

- The algorithm CAN be applied recursively.

- Result = $n^{2.807}$

Remark

- the algorithm is bi-linear
- but the problem is somewhat tri-linear:
- 2 inputs + 1 output,
- linear(A_ij) x linear(B_kl) are combined linearly again!

And in fact it HAS a tri-linear formal algebraic representation:

Formal Tri-Linear Representation

$$(x_{11}y_{11}+x_{12}y_{21})z_{11}+(x_{11}y_{12}+x_{12}y_{22})z_{12}+(x_{21}y_{11}+x_{22}y_{21})z_{21}+(x_{21}y_{12}+x_{22}y_{22})z_{22}=$$

$$(x_{11}+x_{22})(y_{11}+y_{22})(z_{11}+z_{22})+(x_{21}+x_{22})y_{11}(z_{21}-z_{22})+x_{11}(y_{12}-y_{22})(z_{12}+z_{22})+$$

$$x_{22}(y_{21}-y_{11})(z_{11}+z_{21})+(x_{11}+x_{12})y_{22}(-z_{11}+z_{12})+(x_{21}-x_{11})(y_{11}+y_{12})z_{22}+$$

a trick to write many equations as one single equation (!)

provides better understanding…

minimum number of x = rank of this tri-linear form (a.k.a. Tensor Rank)

= its Multiplicative Complexity (MC)

5 Symmetries

Tri-linear view unlocks a hidden world of symmetries of the problem

1. One can permute the r indexes i.
2. One can cyclically shift the three sets of matrices, $A^{(i)}, B^{(i)}$ and $C^{(i)}$ for $1 \leq i \leq r$ becomes $B^{(i)}, C^{(i)}$ and $A^{(i)}$ for $1 \leq i \leq r$.
3. One reverse the order and transpose: $A^{(i)}, B^{(i)}$ and $C^{(i)}$ for $1 \leq i \leq r$ becomes $(C^{(i)})^T, (B^{(i)})^T$ and $(A^{(i)})^T$ for $1 \leq i \leq r$.
4. One can rescale as follows: $a_i A^{(i)}, b_i B^{(i)}$ and $c_i C^{(i)}$ for $1 \leq i \leq r$ where a_i, b_i, c_i are rational coefficients with $a_i b_i c_i = 1$ for each $1 \leq i \leq r$.
5. This method is called "sandwiching". We replace $A^{(i)}, B^{(i)}$ and $C^{(i)}$ for $1 \leq i \leq r$ by $UA^{(i)}V^{-1}$, $VB^{(i)}W^{-1}$ and $WC^{(i)}U^{-1}$, where U, V, W are three arbitrary invertible matrices.

Invariants

All the known symmetries leave invariant determinants???
a set of 3 x r matrices nxn.

This can be used to prove that two solutions are NOT equivalent.

It is known that ALL solutions to Strassen's 2x2 problem are the same (isomorphic wrt to these symmetries).

Brent Equations [1970]

Obtained directly from the tri-linear form.

$$\forall i \forall j \forall k \forall l \forall m \forall n$$

$$\sum_{i=1}^{r} A_{ij}^{(i)} B_{kl}^{(i)} C_{mn}^{(i)} = \delta_{ni} \delta_{jk} \delta_{lm}$$

3x3 Matrices

- Laderman [1976]; 23 multiplications.
- Doing 22 (or showing it cannot be done) is one of the most famous problems in computer science, 35 years, in every book about algorithms and data structures…
- In 1986 Johnson and McLoughlin found some new solutions (for 23)

3x3 Matrices

In 2011 we solved the Brent equations with a SAT solver

We also prove that it is a NEW solution NOT isomorphic to Laderman and neither to Johnson-McLoughlin.

Courtois Bard and Hulme:
"A New General-Purpose Method to Multiply 3x3 Matrices Using Only 23 Multiplications",
http://arxiv.org/abs/1108.2830

3x3 Matrices

We have FULLY automated the problem:

- Write Brent equations
- Consider only solutions in 0,1 = integers modulo 2.
- Convert to SAT with Courtois-Bard-Jefferson method
- Lift the solution from GF(2) to the general bigger fields by another constraint satisfaction algorithm (easy in practice).

As it is a fully automated process of discovery, we are very close to doing 22, just need more CPUs…

```
P01 := (a_2_3) *  (-b_1_2+b_1_3-b_3_2+b_3_3);
P02 := (-a_1_1+a_1_3+a_3_1+a_3_2) *  (b_2_1+b_2_2);
P03 := (a_1_3+a_2_3-a_3_3) *  (b_3_1+b_3_2-b_3_3);
P04 := (-a_1_1+a_1_3) *  (-b_2_1-b_2_2+b_3_1);
P05 := (a_1_1-a_1_3+a_3_3) *  (b_3_1);
P06 := (-a_2_1+a_2_3+a_3_1) *  (b_1_2-b_1_3);
P07 := (-a_3_1-a_3_2) *  (b_2_2);
P08 := (a_3_1) *  (b_1_1-b_2_1);
P09 := (-a_2_1-a_2_2+a_2_3) *  (b_3_3);
P10 := (a_1_1+a_2_1-a_3_1) *  (b_1_1+b_1_2+b_3_3);
P11 := (-a_1_2-a_2_2+a_3_2) *  (-b_2_2+b_2_3);
P12 := (a_3_3) *  (b_3_2);
P13 := (a_2_2) *  (b_1_3-b_2_3);
P14 := (a_2_1+a_2_2) *  (b_1_3+b_3_3);
P15 := (a_1_1) *  (-b_1_1+b_2_1-b_3_1);
P16 := (a_3_1) *  (b_1_2-b_2_2);
P17 := (a_1_2) *  (-b_2_2+b_2_3-b_3_3);
P18 := (-a_1_1+a_1_2+a_1_3+a_2_2+a_3_1) *  (b_2_1+b_2_2+b_3_3);
P19 := (-a_1_1+a_2_2+a_3_1) *  (b_1_3+b_2_1+b_3_3);
P20 := (-a_1_2+a_2_1+a_2_2-a_2_3-a_3_3) *  (-b_3_3);
P21 := (-a_2_2-a_3_1) *  (b_1_3-b_2_2);
P22 := (-a_1_1-a_1_2+a_3_1+a_3_2) *  (b_2_1);
P23 := (a_1_1+a_2_3) *  (b_1_2-b_1_3-b_3_1);
expand(P02+P04+P07-P15-P22-a_1_1*b_1_1-a_1_2*b_2_1-a_1_3*b_3_1);
expand(P01-P02+P03+P05-P07+P09+P12+P18-P19-P20-P21+P22+P23-
a_1_1*b_1_2-a_1_2*b_2_2-a_1_3*b_3_2);
expand(-P02-P07+P17+P18-P19-P21+P22-a_1_1*b_1_3-a_1_2*b_2_3-a_1_3*b_3_3);
expand(P06+P08+P10-P14+P15+P19-P23-a_2_1*b_1_1-a_2_2*b_2_1-a_2_3*b_3_1);
expand(-P01-P06+P09+P14+P16+P21-a_2_1*b_1_2-a_2_2*b_2_2-a_2_3*b_3_2);
expand(P09-P13+P14-a_2_1*b_1_3-a_2_2*b_2_3-a_2_3*b_3_3);
expand(P02+P04+P05+P07+P08-a_3_1*b_1_1-a_3_2*b_2_1-a_3_3*b_3_1);
expand(-P07+P12+P16-a_3_1*b_1_2-a_3_2*b_2_2-a_3_3*b_3_2);
expand(-P07-P09+P11-P13+P17+P20-P21-a_3_1*b_1_3-a_3_2*b_2_3-a_3_3*b_3_3);
```

23x

Our Solution

arxiv.org/abs/1108.2830

MC of Tri-Linear Functions

Remember Gauss in 1805?

multiplying two complex numbers:

- naïve method

4x

$(a + bi) \cdot (c + di) = (ac-bd) + (bc+da)i$

- Gauss method:

$P1 = c(a+b)$

$P2 = a(d-c)$

3x

$P3 = b(c+d)$

$(a + bi) \cdot (c + di) = (P1-P3) + (P1+P2)i$

What About 3 Complex Numbers?

- naïve method

$(a + bi) * (c + di) * (e + fi) = (a c e - a d f - b c f - b d e)$
$+ i(a c f + a d e + b c e - b d f)$

16x

In GF(2) we can do 5 multiplications total!

P1:=(a+b+e+f)*(c+d+e+f);

P2:=(a+e)*(d+e);

P3:=(c+f)*(b+f);

5x

Im := P4:= (P1+P2+P3+a+d+e)*(P1+e+f);

Re := P5:= (P1+e+f)*(P1+P4+a+b+c+d+1);

Our Paper

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

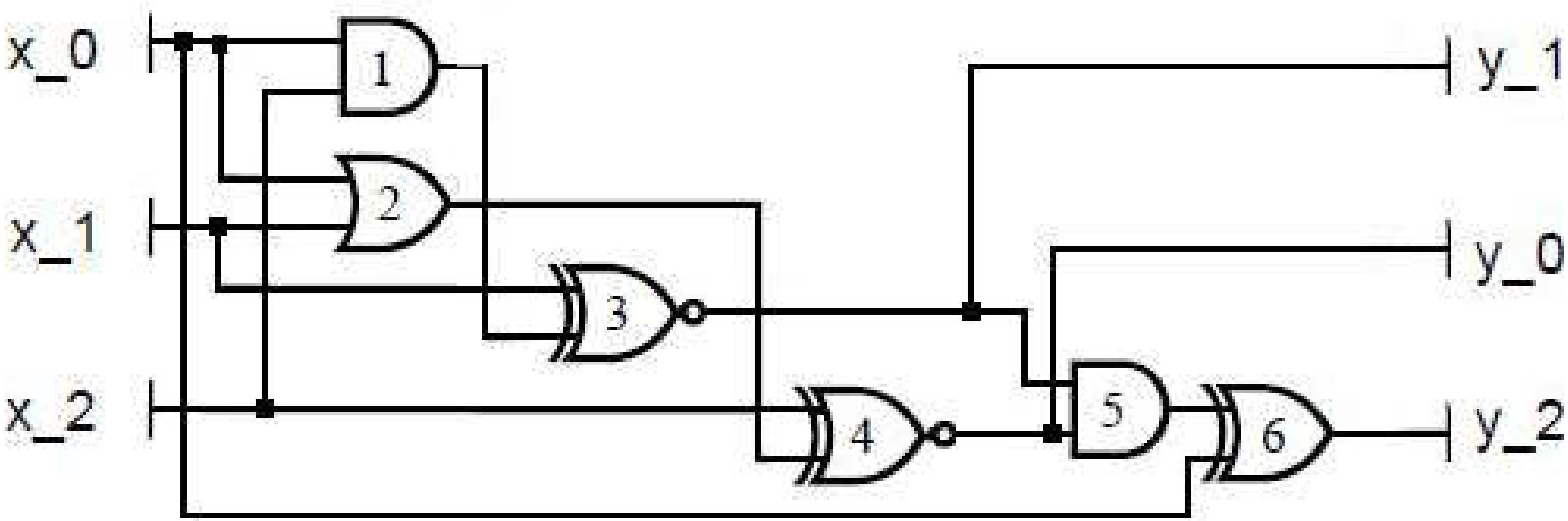
Presented during COMPUTATION TOOLS 2012, The Third International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, held in Nice, France - July 22-27, 2012

IARIA Board

MC of Arbitrary Functions

Logic Synthesis

a mundane problem of practical electronics solved by engineers…



Complexity

= the lofty name given by scientists
to the same problems…

Complexity Theory:

most of it is about "what we don't know":

Complexity Theory – Positive Aspect

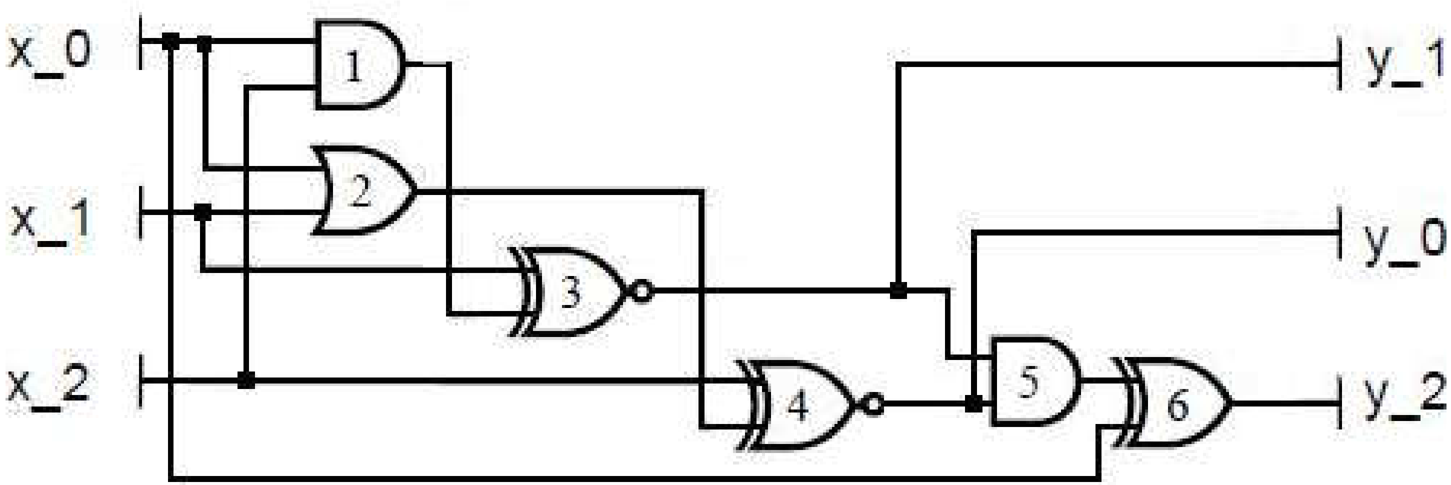
Also defines NP-hard problems.

They are sort of “universal” problems.

If an algorithm solves 3-SAT in PTIME than we can also solve Travelling Salesman in PTIME and all the other famous problems

Circuit Complexity

- Multiplicative Complexity (MC) = minimum number of 2-input AND gates, NOT and XOR gates go for free.
- Bitslice Gate Complexity (BGC) is the minimum number of 2-input gates of types XOR,OR,AND needed.
- Gate Complexity (GC) is the minimum number of 2-input gates of types XOR,OR,AND,NAND,NOR,NXOR.
- NAND Complexity (NC) = 2-input NAND gates only



Motivation

Motivation

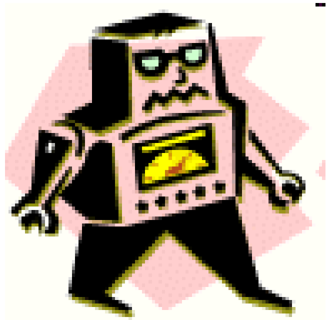
- silicon = $$$
- software encryption = $$$
- secure implementation in smart cards = $$$

- cryptanalysis

Crypto and MC

Cryptography and MC

- Most of energy and silicon in smart cards and SSL web servers is spent on cryptography which could be improved with “lower MC”
  - (for all sorts of algorithms, RSA, ECC also symmetric ciphers use multiplications or AND gates etc.).



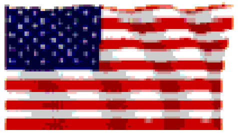
AES and MC

AES

Advanced Encryption Standard:
US government standard and a (de facto) world standard for commercial applications.

Key sizes 128, 192 and 256 bits.

- In 2000 NIST selected Rijndael as the AES.
- Serpent was second in the number of votes.



11 years later:

In 2011, the year in which AES is becoming standard in every new Intel CPU… (i5 and above)

AES was broken (but really only in theory).

Today's most competitive ciphers are precisely PRESENT Serpent and GOST…

- Unhappily GOST was also broken in 2011.
- Serpent not very popular still.
- PRESENT is popular within research community but not widely used..

=> MC is at the heart of optimisation of ALL these ciphers.

AES S-box

$$x \to x^{-1}$$

in GF(256)

AES S-box

$$x \rightarrow x^{-1}$$

in GF(256)

BTW. Its “Implicit” Multiplicative Complexity = 1

$xy=1$

$x \to x^{-1}$ 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

$x \rightarrow x^{-1}$ n=8 or Full-Size AES S-box

$t_2 = y_{12} \times y_{15}$	$t_3 = y_3 \times y_6$	$t_4 = t_3 + t_2$
$t_5 = y_4 \times x_7$	$t_6 = t_5 + t_2$	$t_7 = y_{13} \times y_{16}$
$t_8 = y_5 \times y_1$	$t_9 = t_8 + t_7$	$t_{10} = y_2 \times y_7$
$t_{11} = t_{10} + t_7$	$t_{12} = y_9 \times y_{11}$	$t_{13} = y_{14} \times y_{17}$
$t_{14} = t_{13} + t_{12}$	$t_{15} = y_8 \times y_{10}$	$t_{16} = t_{15} + t_{12}$
$t_{17} = t_4 + t_{14}$	$t_{18} = t_6 + t_{16}$	$t_{19} = t_9 + t_{14}$
$t_{20} = t_{11} + t_{16}$	$t_{21} = t_{17} + y_{20}$	$t_{22} = t_{18} + y_{19}$
$t_{23} = t_{19} + y_{21}$	$t_{24} = t_{20} + y_{18}$	
$t_{25} = t_{21} + t_{22}$	$t_{26} = t_{21} \times t_{23}$	$t_{27} = t_{24} + t_{26}$
$t_{28} = t_{25} \times t_{27}$	$t_{29} = t_{28} + t_{22}$	$t_{30} = t_{23} + t_{24}$
$t_{31} = t_{22} + t_{26}$	$t_{32} = t_{31} \times t_{30}$	$t_{33} = t_{32} + t_{24}$
$t_{34} = t_{23} + t_{33}$	$t_{35} = t_{27} + t_{33}$	$t_{36} = t_{24} \times t_{35}$
$t_{37} = t_{36} + t_{34}$	$t_{38} = t_{27} + t_{36}$	$t_{39} = t_{29} \times t_{38}$
$t_{40} = t_{25} + t_{39}$		
$t_{41} = t_{40} + t_{37}$	$t_{42} = t_{29} + t_{33}$	$t_{43} = t_{29} + t_{40}$
$t_{44} = t_{33} + t_{37}$	$t_{45} = t_{42} + t_{41}$	$z_0 = t_{44} \times y_{15}$
$z_1 = t_{37} \times y_6$	$z_2 = t_{33} \times x_7$	$z_3 = t_{43} \times y_{16}$
$z_4 = t_{40} \times y_1$	$z_5 = t_{29} \times y_7$	$z_6 = t_{42} \times y_{11}$
$z_7 = t_{45} \times y_{17}$	$z_8 = t_{41} \times y_{10}$	$z_9 = t_{44} \times y_{12}$
$z_{10} = t_{37} \times y_3$	$z_{11} = t_{33} \times y_4$	$z_{12} = t_{43} \times y_{13}$
$z_{13} = t_{40} \times y_5$	$z_{14} = t_{29} \times y_2$	$z_{15} = t_{42} \times y_9$
$z_{16} = t_{45} \times y_{14}$	$z_{17} = t_{41} \times y_8$	

Fig. 3. The middle non-linear section

32x

eprint.iacr.org/2009/191/

151 gates,
cheapest known

Can we do 4?

Boyar and Peralta has proven that 4 is impossible. Manual proof.

We can do this routinely in an automated way.

Two sorts of SAT solvers:

- stochastic
- complete

some of these output a file which is a formal proof of UNSAT.

SAT Solvers in the Cloud

UCL spin-off company

solving SAT problems on demand…

commercial but also for free…

PRESENT and MC

Theorem [Courtois et al. 2010]

The Multiplicative Complexity of the PRESENT S-box is exactly 4.

(cheaper than AES at the same size which has 5)

Our Method

Quantified SAT Problem:

$$\forall i \forall j \forall k \forall l \forall m \forall n$$

Equations…

Convert to SAT and say that holds for sufficiently many small weight cases…

Generic very powerful method. We also use it for many other things…

But not so good for MM 23 result, Brent Equations are another sort of more “formal algebraic” method and can be seen as the same with a suitable choice of basis…

Bit-Slice 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

PRESENT Software

We have co-authored an open-source implementation of PRESENT, the best currently known.

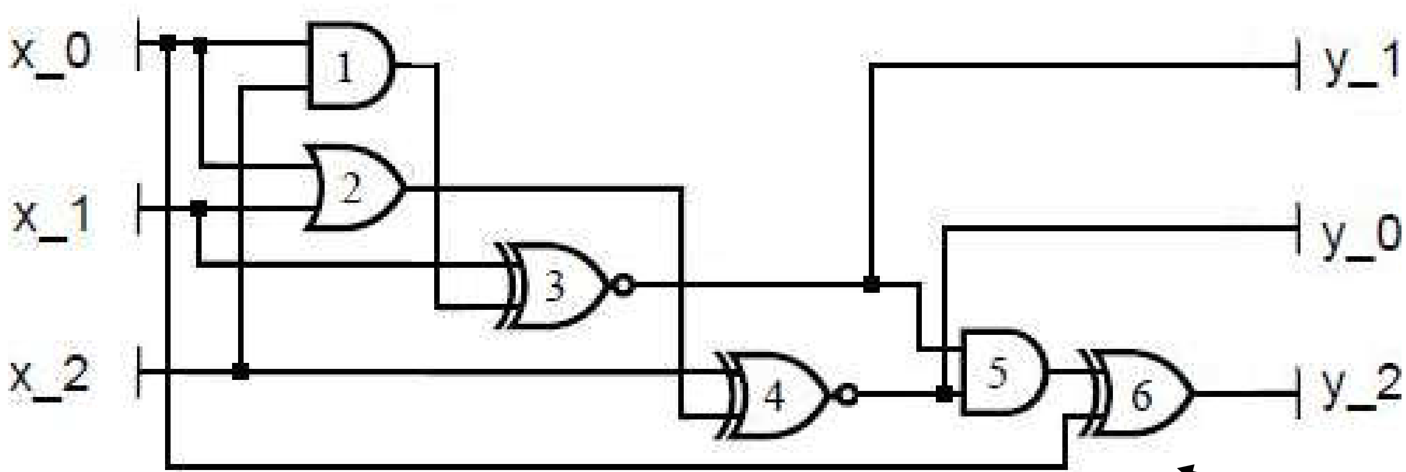
algebraic_attacks / present_bitslice.c

dd3845601204 266 loc 8.7 KB

```
/**
 * Bit-Slice Implementation of PRESENT in pure standard C.
 * v1.5 26/08/2011
 *
 * The authors are
 *  Martin Albrecht <martinralbrecht@googlemail.com>
 *  Nicolas T. Courtois <firstinitial.family_name@cs.ucl.ac.uk>
 *  Daniel Hulme <firstname@satalia.com>
 *  Guangyan Song <firstname.lastname@gmail.com>
 * This work was partly funded by the Technology Strategy Board
 * in the United Kingdom under Project No 9626-58525.
 *
 * NEW FEATURES in this version:
 * - it contains an optimized sbox() using 15 only gates, instead of 39
 *   previously
 * - it now supports both 80-bit and 128-bit PRESENT
 * - it contains test vectors for both versions
 *
 * This is a simple and straightforward implementation
 * it encrypts at the speed of
 *   59 cycles per byte on Intel Xeon 5130 1.66 GHz
 * this can be compared to for example
 *   147 cycles per byte for optimized triple DES on the same CPU
```

Another S-box – CTC2

Our new design:



PROVEN OPTIMAL

More About CTC2 S-box.

Theorem 3.1.

- The Multiplicative Complexity (MC) is exactly 3
  - 3 AND + any number of XOR gates.
- The Bitslice Gate Complexity (BGC) is exactly 8
  - (allowed are XOR,OR,AND,OR).
- The Gate Complexity (GC) is exactly 6
  - in addition allowing NAND,NOR,NXOR.
- The NAND Complexity (NC) is exactly 12
  - only NAND gates and constants.

ALL PROVEN OPTIMAL

Optimal S-boxes

Theory of Optimal S-boxes

There is a theory of “optimal S-boxes” which are the best possible w.r.t. linear and differential criteria to build ciphers…

On the Classification of 4 Bit S-Boxes

G. Leander[1,*] and A. Poschmann[2]

[1] GRIM, University Toulon, France
Gregor.Leander@rub.de
[2] Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum, Germany
poschmann@crypto.rub.de

Affine Equivalence

We call two S-boxes S_1, S_2 equivalent if there exist bijective linear mappings A, B and constants $a, b \in \mathbb{F}_2^4$ such that

$$S'(x) = B(S(A(x) + a)) + b.$$

If two S-boxes S_1 and S_2 are equivalent in the above sense we denote this by $S_1 \sim S_2$.

Abstract. In this paper we classify all optimal 4 bit S-boxes. Remarkably, up to affine equivalence, there are only 16 different optimal S-boxes.

Affine Equivalence

Only 16 S-boxes are "good".

On the Classification of 4 Bit S-Boxes

G. Leander[1,*] and A. Poschmann[2]

[1] GRIM, University Toulon, France
Gregor.Leander@rub.de
[2] Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum, Germany
poschmann@crypto.rub.de

4x4 occur in Serpent, PRESENT, GOST, [AES…]

not surprising that some of the S-boxes of the Serpent cipher are linear equivalent. Another advantage of our characterization is that it eases the highly non-trivial task of choosing good S-boxes for hardware dedicated ciphers a lot.

Affine Equivalence => MC?!

Yes!

1. Determine another S-box for which our S-box is an affine equivalent of another S-box, for which the MC was already computed.
2. The affine equivalence can be determined by methods of [2] which are actually essentially the same methods which have been proposed at the same conference 10 years earlier [9] in a slightly different context.

Original algorithm: see

- Courtois Goubin Patarin, Eurocrypt 1998

Adaptation:

- Biryukov et al, Eurocrypt 2008

Affine Equivalence in GOST

Or do Russian code makers read French-German papers about crypto S-boxes…

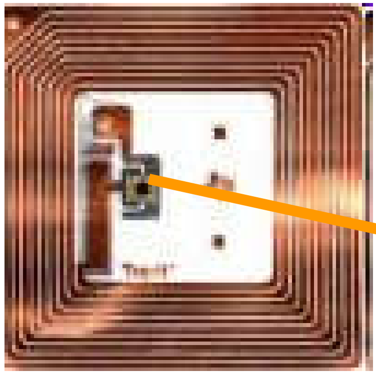
S-box Set Name	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$
GostR3411_94_TestParamSet	36	02	03	04		06	35	08
- their inverses		02	03	04		06		08
GostR3411_94_CryptoProParamSet			$Lu1$	14	G_{10}		G_8	
- their inverses			$Lu1$	14	G_{10}		G_8	
Gost28147_TestParamSet	21	21			25			28
- their inverses	21	21			25			28
Gost28147_CryptoProParamSetA	31	32	33	G_8	35	36	37	38
- their inverses	31	32	33	G_8			37	38
Gost28147_CryptoProParamSetB	G_{13}	G_{13}	G_{13}	G_{11}	G_7	G_7	G_{11}	G_6
- their inverses	G_{13}	G_{13}	G_{13}	G_{11}	G_7	G_7	G_{11}	G_6
Gost28147_CryptoProParamSetC	G_7	G_4	G_6	G_{13}	G_{13}	G_6	G_{11}	G_{13}
- their inverses	G_7	G_4	G_6	G_{13}	G_{13}	G_6	G_{11}	G_{13}
Gost28147_CryptoProParamSetD	G_{13}	G_{13}	G_{13}	G_4	G_{12}	G_4	G_{13}	G_7
- their inverses	G_{13}	G_{13}	G_{13}	G_4	G_{12}	G_4	G_{13}	G_7
GostR3411_94_SberbankHashParamset			74	75	76		78	
- their inverses			74	75	78		76	
GOST ISO 18033-3 proposal	G_9	G_9	G_9	G_9	G_9	G_9	G_9	G_9
- their inverses	G_9	G_9	G_9	G_9	G_9	G_9	G_9	G_9

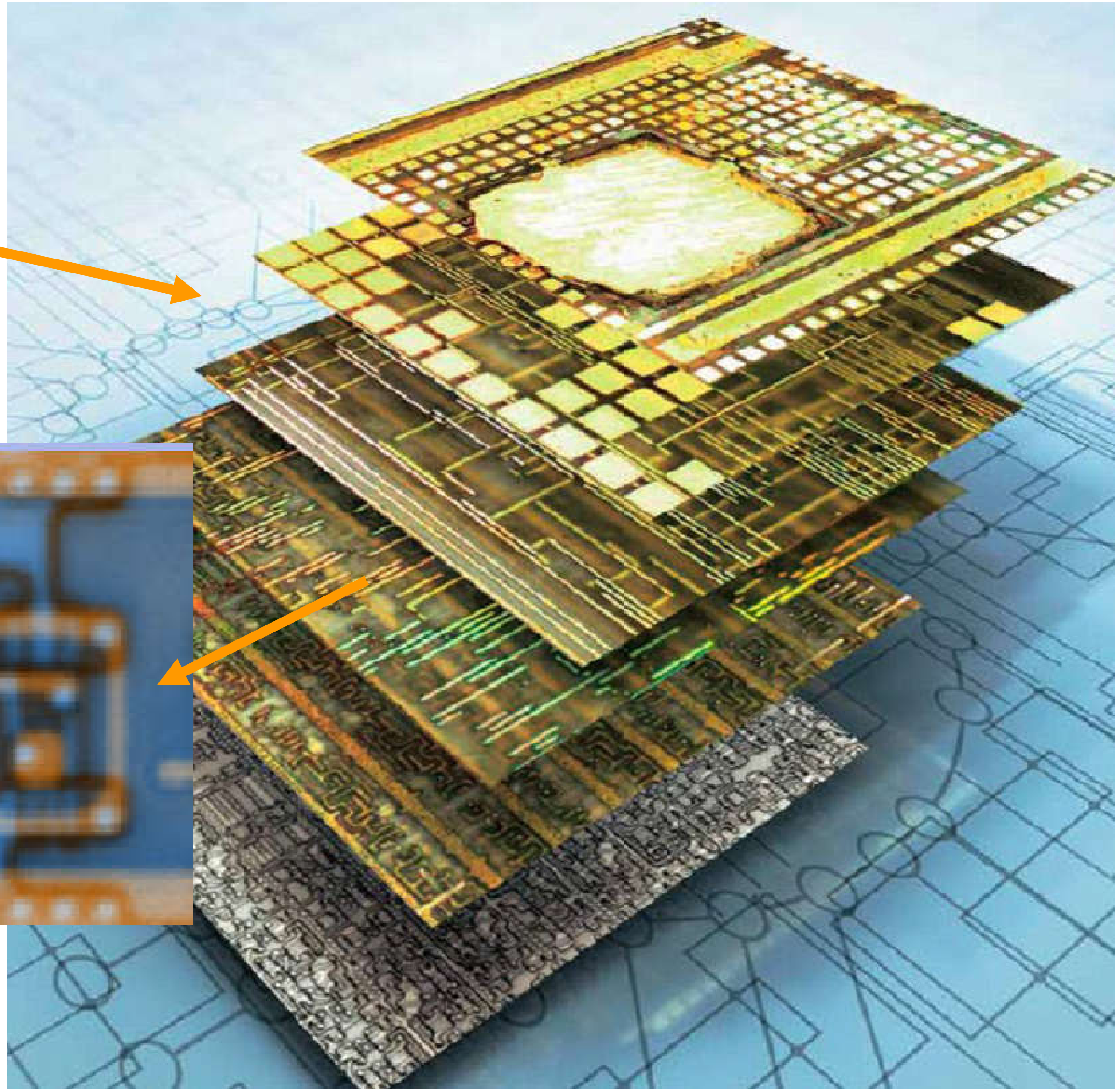
Affine Equivalence in GOST - Observations

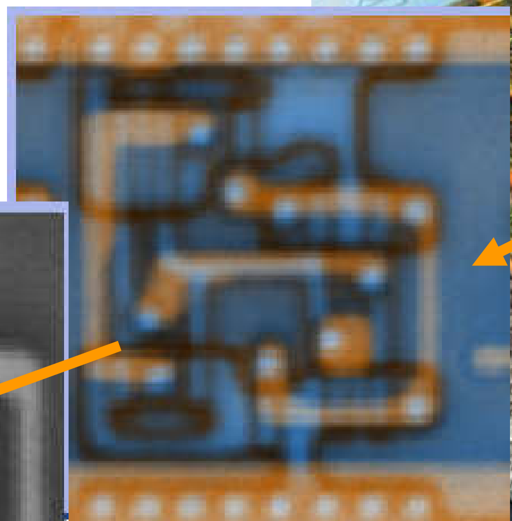
- There was a historical evolution of GOST S-boxes towards boxes of type G_i which are optimal against LC/DC
- most of more recent S-boxes which appear in OpenSSL are one of the G_i
- BTW. 12 out of these 'optimal' S-boxes are affine equivalent to their own inverse.
- Interestingly, only 9 of these 12 which are namely G_{4},G_{6},G_{7}, G_{8}, G_{9}, G_{10},G_{11},G_{12},G_{13} occur in our table for GOST, and only those which are equivalent to their inverse occur in this table.

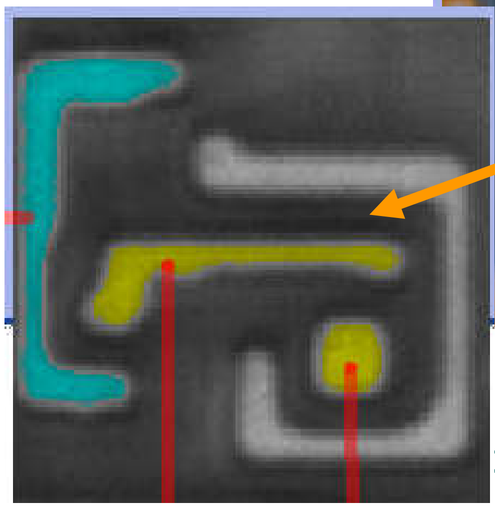
Reverse Engineering

Hardware Reverse-Engineering





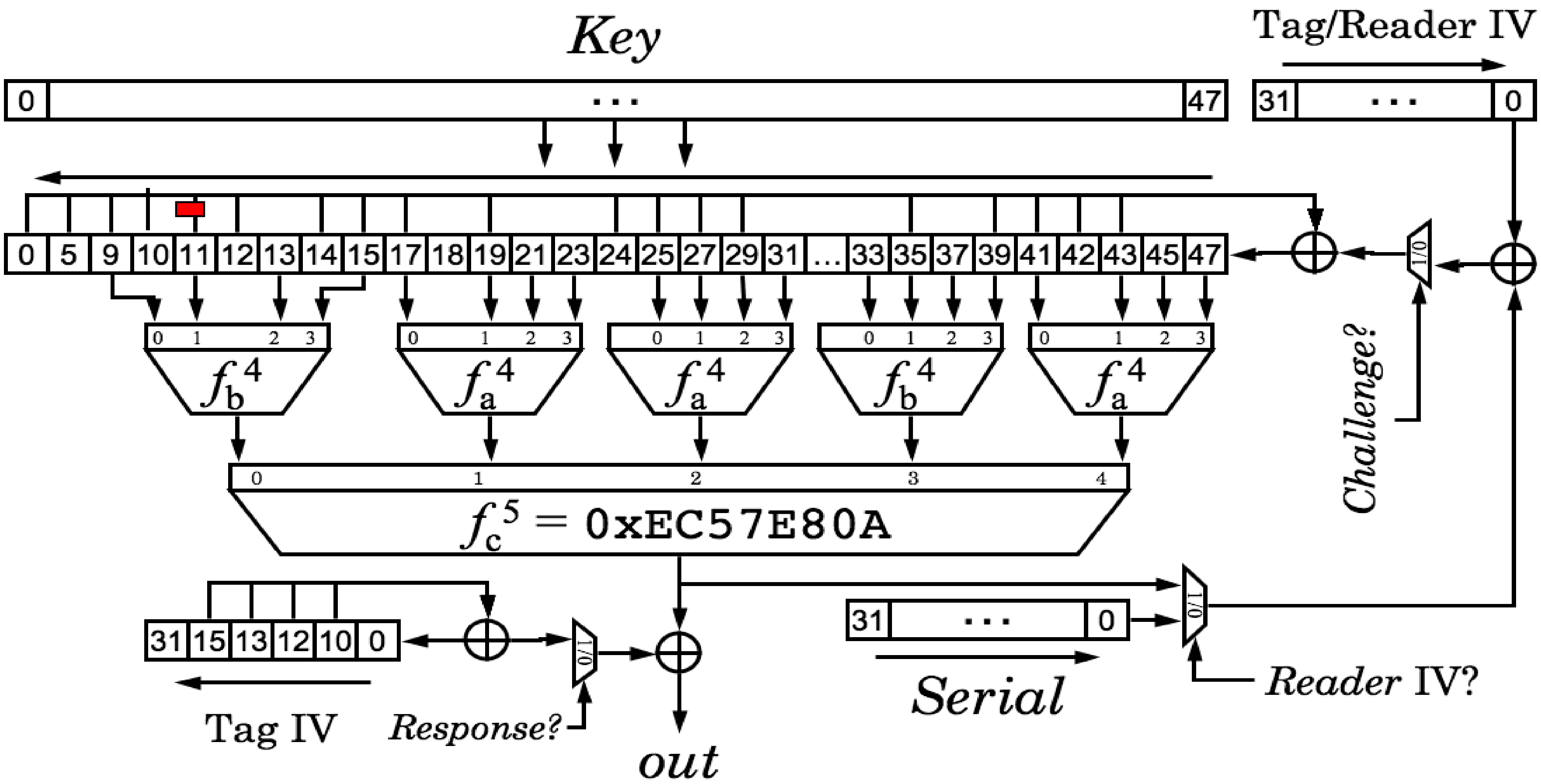




Software Reverse Engineering

- Reproduce the cipher by queries to it.
- Holy grail for serious hackers and cryptanalysts, even before we try to break a cipher system, we need to know the spec.
- Possible IF we can compute MC for circuits.
  - for small circuits WE CAN do it with SAT solvers.
  - In 2008/2009 Dutch researchers have published a “software reverse engineering method” for MiFare Classic Crypto-1 cipher.

Crypto1 Cipher

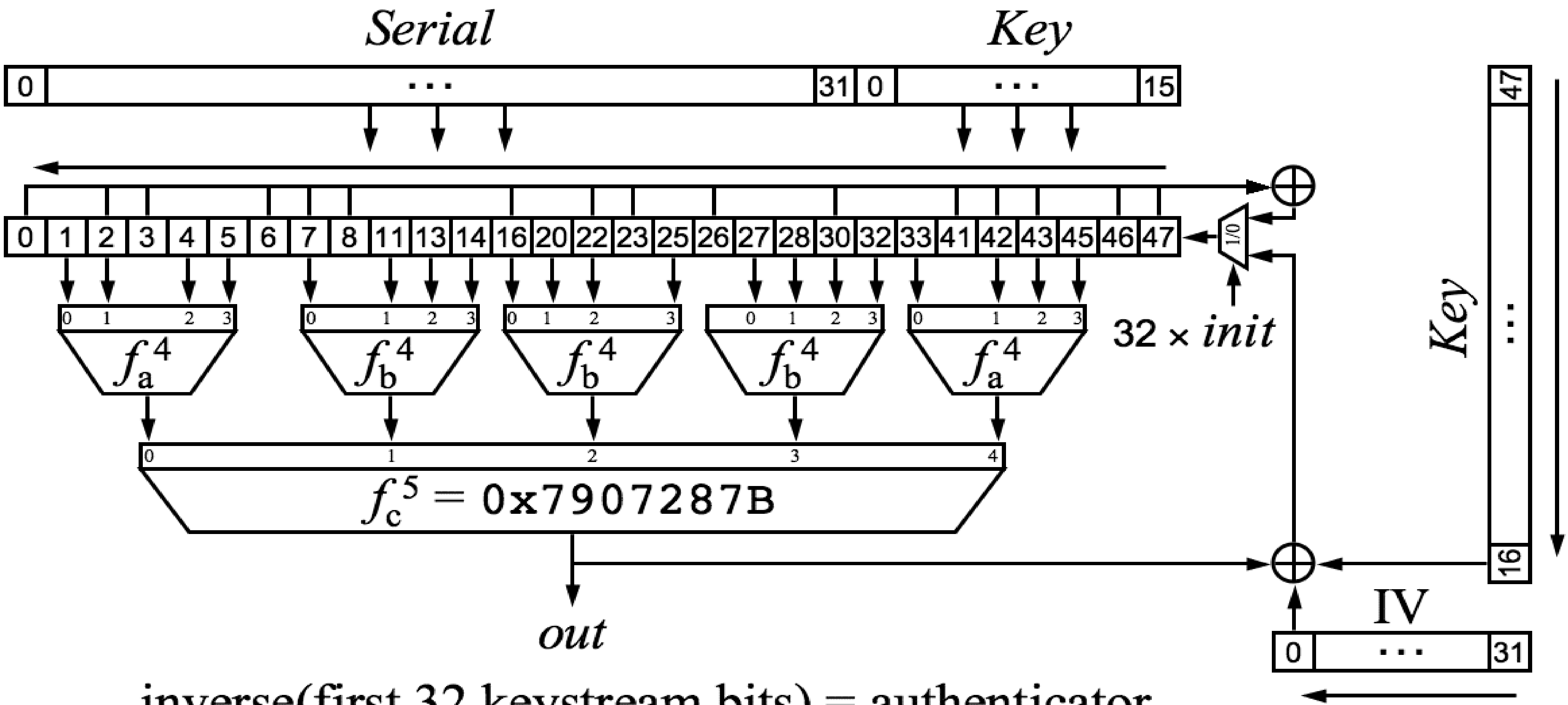


$$f_a^4 = \texttt{0x9E98} = (a+b)(c+1)(a+d)+(b+1)c+a$$

$$f_b^4 = \texttt{0xB48E} = (a+c)(a+b+d)+(a+b)cd+b$$

Tag IV ⊕ Serial is loaded first, then Reader IV ⊕ NFSR

Hitag2 Cipher



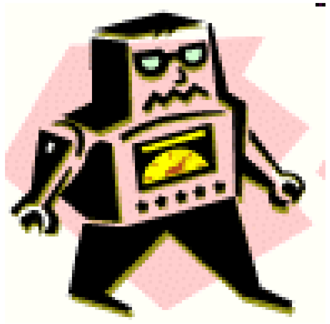
inverse(first 32 keystream bits) = authenticator

f_a^4 = 0x2C79 = abc+ac+ad+bc+a+b+d+1

f_b^4 = 0x6671 = abd+acd+bcd+ab+ac+bc+a+b+d+1

In the world of Serious Cryptanalysis

Beyond Crypto-1



…AC can break “any cipher”, if not too complex…

- We can break Hitag2 in 1 day
  - with a SAT solver.

Cf. Nicolas T. Courtois, Sean O'Neil and Jean-Jacques Quisquater: “Practical Algebraic Attacks on the Hitag2 Stream Cipher”,

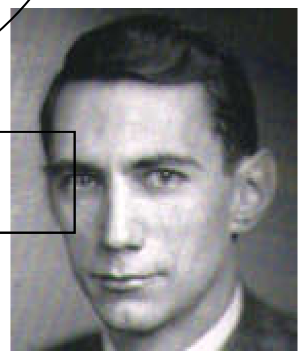
In ISC 2009, Springer.

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]



Motivation

Usual linear and differential cryptanalysis do require huge quantities of known/chosen plaintexts.

Q: What kind of cryptanalysis is possible when the attacker has

only one known plaintext (or very few) ?

Claim: This question did not receive sufficient attention.

Two Worlds:

- **The “approximation” cryptanalysis:**
  - Linear, differential, high-order differential, impossible differential, Jakobsen-Knudsen approximation, etc..
  - All are based on probabilistic characteristics true with some probability.
  - Consequently, <u>the security will grow exponentially</u> with the number of rounds, and <u>so does the number of required plaintexts</u> in the attacks (main limitation in practice).
- **The “exact algebraic” approach:**
  - Write equations to solve, true with probability 1.
  - Very small number of known plaintexts required.

Why Cryptographers Get It Wrong…

By assuming that 2^{43} time 2^{43} KP is feasible (it isn't) block ciphers have too many rounds.

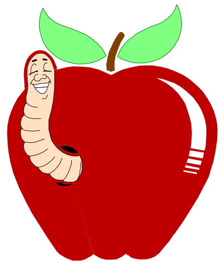
Some attacks which are really feasible, e.g. 2^{70} and 4 KP are never studied

⇒because somebody will say that they less practical than other already known attacks…

⇒in fact they are the only attacks feasible.

- ⇒iIn real-life applications the key will be changed and 2^{43} KP never happens while 2^{70} and 4 KP is costly but realistic.

What Makes Ciphers Vulnerable



Design of Symmetric Ciphers

A mix of sufficiently many
highly non-linear functions….

Def: "I / O Degree" = "Graph AI"

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 $y = f(x)$.

A "good" cipher should use at least some components with high I/O degree.

AES S-box

$$x \rightarrow x^{-1}$$

in GF(256)

BTW. Its “Implicit” Multiplicative Complexity = 1
I/O degree = 2
$xy=1$

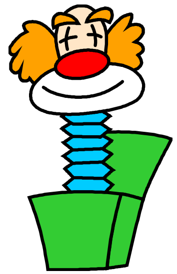
AES S-boxes

$$(y_1, \ldots, y_8) = S\,(x_1, \ldots, x_8)\ .$$

Theorem [Courtois-Pieprzyk]: For each S-box there are $r=39$ quadratic equations

with 16 variables x_i and y_i,

that are true with probability 1.



!

Optimal S-boxes ?

[Anne Canteaut, Marion Videau, Eurocrypt 2002]:

Optimal for linear, differential and high-order differential attacks.

We do not know any **worse** S-box in terms of r.

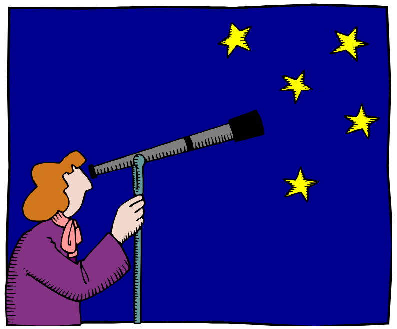
Power	-1	3	5	7
Equations / S-box r=	39	39	34	24

Break AES with Quadratic Equations?

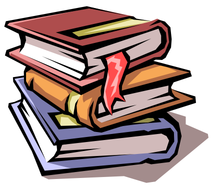
Rijndael 128 bit: to recover the secret key can be rewritten as MQ:

8000 quadratic equations
1600 variables in GF(2).

But how to solve it ?

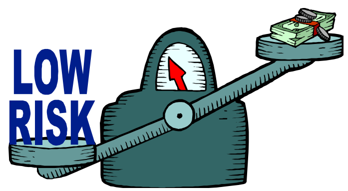


XL Algorithm, Gröbner Bases



- [Shamir, Patarin, Courtois, Klimov, Eurocrypt’2000]
- [Courtois, ICISC’02], [Courtois, Patarin, CT-RSA’03]
- Gröbner bases, Buchberger algorithm, F4, F5, F5/2 by Jean-Charles Faugère… …
- Recent many paper: Claus Diem, Gwenole Ars, Magali Bardet, Jean-Charles Faugère, Bruno Salvy, Makoto Sugita, Mitsuru Kawazoe, Hideki Imai, Jiun-Ming Chen, Nicolas Courtois, Bo-Yin Yang and others.

XL is too general. Deals with dense systems of equations. Our are sparse (easier).



The principle of XL:

Multiply the initial equations by low-degree monomials:

$$1 = x_5 + x_0x_1 + x_0x_2$$

becomes:

$$x_1 \cdot 1 = x_1 \cdot (x_5 + x_0x_1 + x_0x_2)$$

(degreee 3 now).

The idea of XL:

Multiply equations by low-degree monomials.

- Count new equations: R
- Count new monomials present: T

One term can be obtained in many different ways,

$\Rightarrow$ T grows slower than R.

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).

Block Ciphers

Nicolas Courtois, Joseph Pieprzyk:
Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, in Asiacrypt 2002.

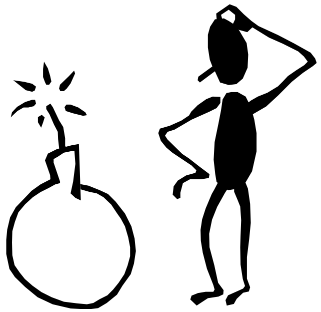
> 500 citations

LOTS of press speculation abut real and imaginary consequences of this…

Vincent Rijmen have said:

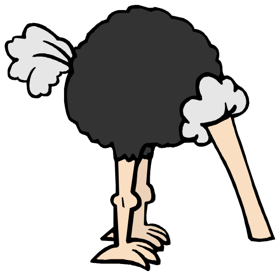
"XSL is not an attack, it is a dream"

Is AES Broken ?



It is widely believed that XSL does not work..

In fact there is no proof…



Stream Ciphers

UCL

Stream Ciphers

Nicolas Courtois, Willi Meier: Algebraic Attacks on Stream Ciphers with Linear Feedback, in EuroCrypt 2003.

> 500 citations

"Fast Moving Front" in computer science

(top 1% result in whole of CS)

THOMSON

Essential Science Indicators special TOPICS

Citing URL: http://www.esi-topics.com/fmf/2005/july05-NicolasCourtois

From •>>July 2005

Nicolas T. Courtois answers a few questions about this month's fast moving front in the field of Computer Science.

Field: Computer Science

Article: Algebraic attacks on stream ciphers with linear feedback

Authors: Courtois, NT; Meier, W
Journal: LECT NOTE COMPUT SCI, 2656: 345-359, 2003
Addresses:
Schlumberger Smart Cards, Cryptog Res, 36-38 Rue Princesse, BP 45, F-78430 Louveciennes, France.
Schlumberger Smart Cards, Cryptog Res, F-78430 Louveciennes, France.
FH Aargau, CH-5210 Windisch, Switzerland.

Why do you think your paper is highly cited?

This paper proposes a new, surprisingly powerful method for attacking stream ciphers. It allows us to break not only a few ciphers that were up until now believed quite secure, but also holds even deeper consequences. For about a decade, designers of ciphers have

DES Cipher

DES

At a first glance,
DES seems to be a very poor target:

there is (apparently)
no strong algebraic structure
of any kind in DES

What's Left ?

Idea 1: (IO)

Algebraic I/O relations.

Theorem [Courtois-Pieprzyk]:

Every S-box has a low I/O degree.

=>3 for DES.

Idea 2: (Very Sparse)

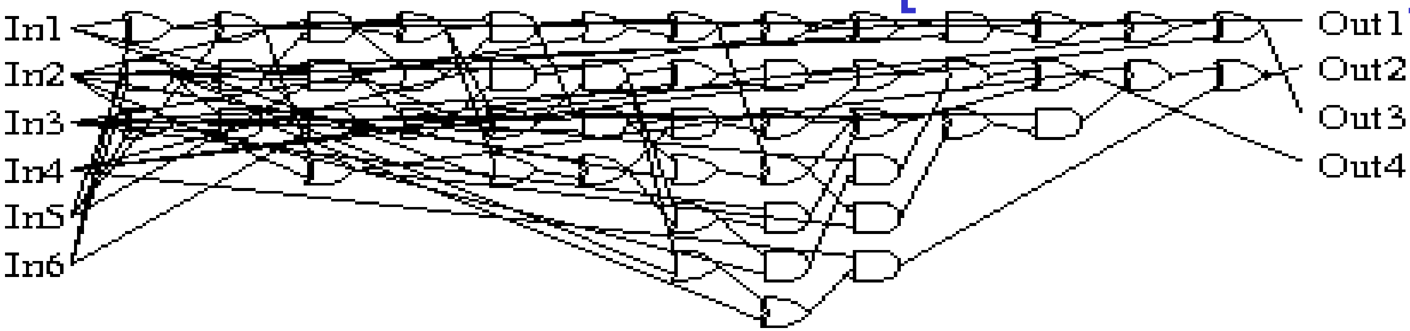
DES has been designed to be implemented in hardware.

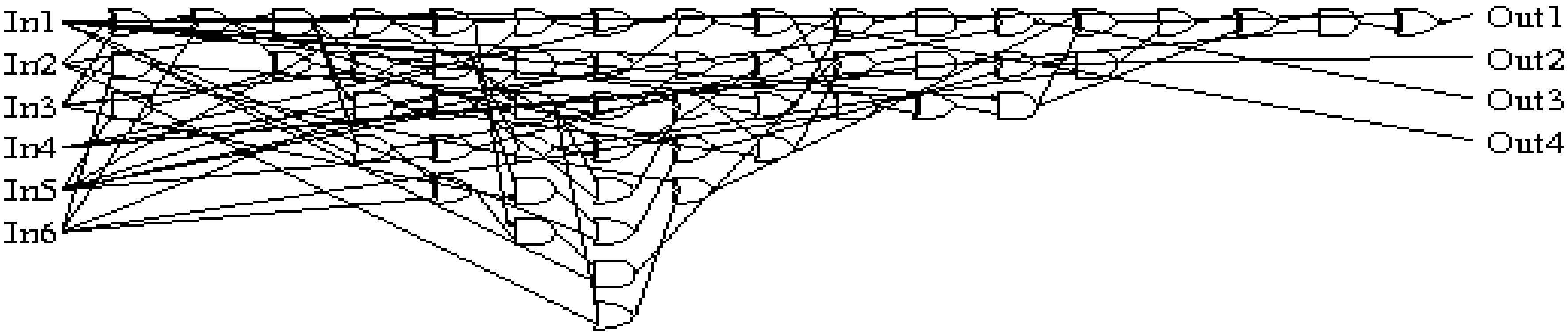
=> Very-sparse quadratic equations at the price of adding some 40 new variables per S-box.

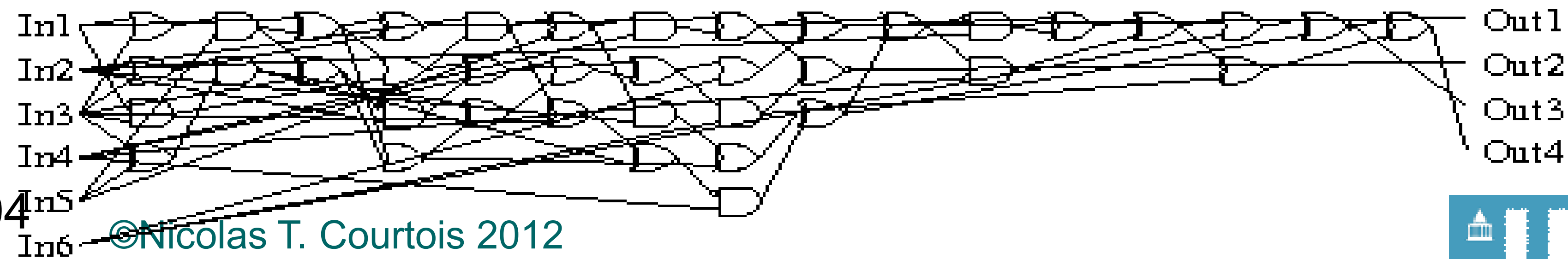
Results ?

Both Idea 1 (IO) and Idea 2 (VS) can be exploited in working key recovery attacks.

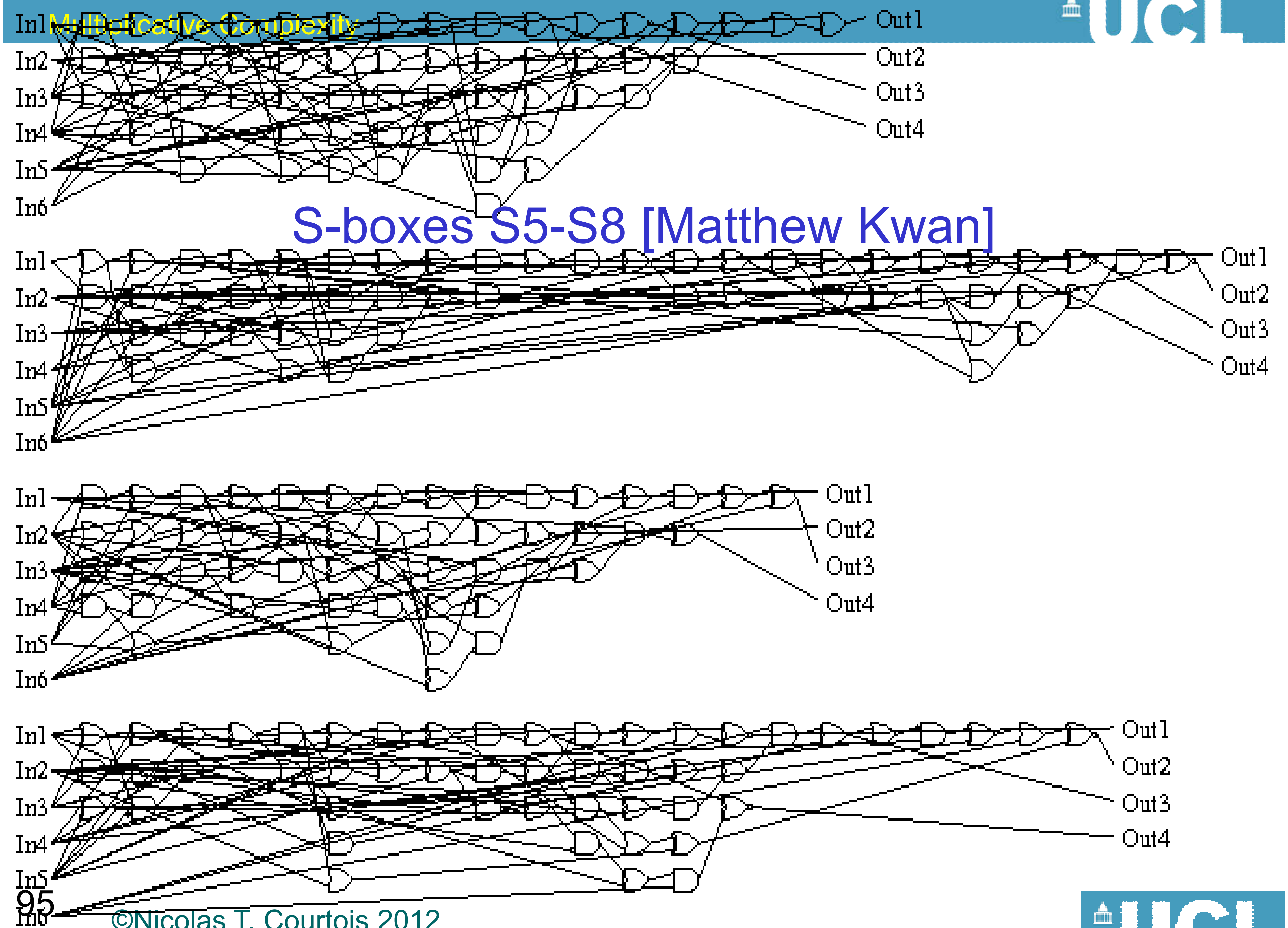
S-boxes S1-S4 [Matthew Kwan]







S-boxes S5-S8 [Matthew Kwan]



***DES Implementation [2013]

- 17% less gates still, by Roman Rusakov
- Bitslice
  - average of 44.125 gates per S-box
    (NB. they found several solutions with the same gate count)
  - vs. 53.375 for Kwan (his XNOR=>2gates).
  - cf. www.openwall.com/lists/john-users/2011/06/22/1
  - or the source code of John the Ripper

Results on DES

Nicolas T. Courtois and Gregory V. Bard:

Algebraic Cryptanalysis of the D.E.S.

In IMA conference 2007, pp. 152-169, LNCS 4887, Springer.

See also:

eprint.iacr.org/2006/402/

Two Attacks on Reduced-Round DES

Cubic IO + Equations ElimLin algorithm:

We recover the key of 5-round DES with 3 KP faster than brute force.

Circuit representation+ ANF-to-CNF + MiniSat 2.0.:

Key recovery for 6-round DES. Only 1 KP (!).

Some Pointers

Ready Software for Windows

Equations generators for some ciphers:

www.cryptosystem.net/aes/toyciphers.html

Some ready programs for algebraic cryptanalysis:

www.cryptosystem.net/aes/tools.html

Ready Encodings:

Some S-box representations:

www.nicolascourtois.com/equations/block/sboxes/misc_sboxes.ZIP

More ready S-box representations:

www.nicolascourtois.com/equations/block/gost/gost_boxes.ZIP

Monomial Encodings

Also bi-monomial: cf. Section 6.2-6.4

https://eprint.iacr.org/2003/184.pdf

Encodings Over GF(8)

Has been produced for CTC2

Related Works

http://eprint.iacr.org/2016/198.pdf - FSE 2016

https://eprint.iacr.org/2017/007.pdf

http://discovery.ucl.ac.uk/1462141/2/PhD_Thesis_Theodosis_Mourouzis.pdf

GOST Cipher

GOST 28148-89

- The Official Encryption Standard of Russian Federation.
- Declassified in 1994.
- Best single-key attack:
  - Shamir et al. 2^{192}
    - FSE 2012, Washington DC, March 2012

  - NEW attack by Courtois: 2^{179}
    - advanced differential attack, March 2012
  - MULTIPLE KEY attack by Courtois: 2^{101}
    - NEW: December 2012

GOST 28148-89

- Very high level of security (256 bits)
  - In theory secure for 200 years…
- Widely used, Crypto ++, Open SSL
- Central Bank of Russia and other Russian banks…
  - not a commercial algorithm for short-term security such as DES…
- Very competitive, less gates that simplified DES, much less than AES
  - [cf CHES 2010]
  - 800 G.E. while AES-128 needs >3100
- In 2010 GOST was also submitted to ISO to become an international standard.

GOST 28148-89

Table 1. Multiplicative Complexity for all known GOST S-Boxes

S-box Set Name	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$
GostR3411_94_TestParamSet	4	5	5	5	5	5	4	5
GostR3411_94_CryptoProParamSet	4	5	5	4	5	5	4	5
Gost28147_TestParamSet	4	4	4	4	4	5	5	5
Gost28147_CryptoProParamSetA	5	4	5	4	4	4	5	5
Gost28147_CryptoProParamSetB	5	5	5	5	5	5	5	5
Gost28147_CryptoProParamSetC	5	5	5	5	5	5	5	5
Gost28147_CryptoProParamSetD	5	5	5	5	5	5	5	5
GostR3411_94_SberbankHashParamset	4	4	4	5	5	4	4	4
GOST ISO 18033-3 proposal	5	5	5	5	5	5	5	5

GOST-P

A version of GOST with 8x PRESENT S-box

- Only 650 G.E.

MC = 4 each exactly (as we already proved).

The authors have obtained in 2011 for their work precisely on PRESENT cipher and 4-bit S-boxes, an "IT Security Price" of 100 000 € which is the highest scientific price in Germany awarded by a private foundation.

Modular Addition

+ modulo 2^{32}

in several ciphers: GOST, SNOW 2.0.

$$(x, y) \mapsto z = x \boxplus y \mod 2^n$$

Theorem 6.1.1. The Multiplicative Complexity (MC) of the addition modulo 2^n is exactly $n-1$.

Modular Addition

$$(x, y) \mapsto z = x \boxplus y \mod 2^n$$

Theorem 6.1.1. The Multiplicative Complexity (MC) of the addition modulo 2^n is exactly $n-1$.

$$(*) \begin{cases} z_0 = x_0 + y_0 \\ z_1 = x_1 + y_1 + c_1 \\ z_2 = x_2 + y_2 + c_2 \\ \vdots \\ z_i = x_i + y_i + c_i \\ \vdots \\ z_{n-1} = x_{n-1} + y_{n-1} + c_{n-1}, \end{cases}$$

$$(*') \begin{cases} c_1 = x_0 y_0 \\ c_2 = x_1 y_1 + (x_1 + y_1) c_1 \\ \vdots \\ c_i = x_{i-1} y_{i-1} + (x_{i-1} + y_{i-1}) c_{i-1} \\ \vdots \\ c_{n-1} = x_{n-2} y_{n-2} + (x_{n-2} + y_{n-2}) c_{n-2} \end{cases}$$

MC (+ Mod 2^n) = n-1 ???

Theorem 6.1.1. The Multiplicative Complexity (MC) of the addition modulo 2^n is exactly $n-1$.

Proof:

we have:

$$xy + (x + y)c = (x + c)(y + c) - c^2$$

1x each

$$x_0 y_0$$

$$x_1 y_1 + (x_1 + y_1) c_1$$

$$x_{i-1} y_{i-1} + (x_{i-1} + y_{i-1}) c_{i-1}$$

$$= x_{n-2} y_{n-2} + (x_{n-2} + y_{n-2}) c_{n-2}$$