Cryptanalysis of GOST

Nicolas T. Courtois

University College London, UK

Outline

1. Cold War cryptography

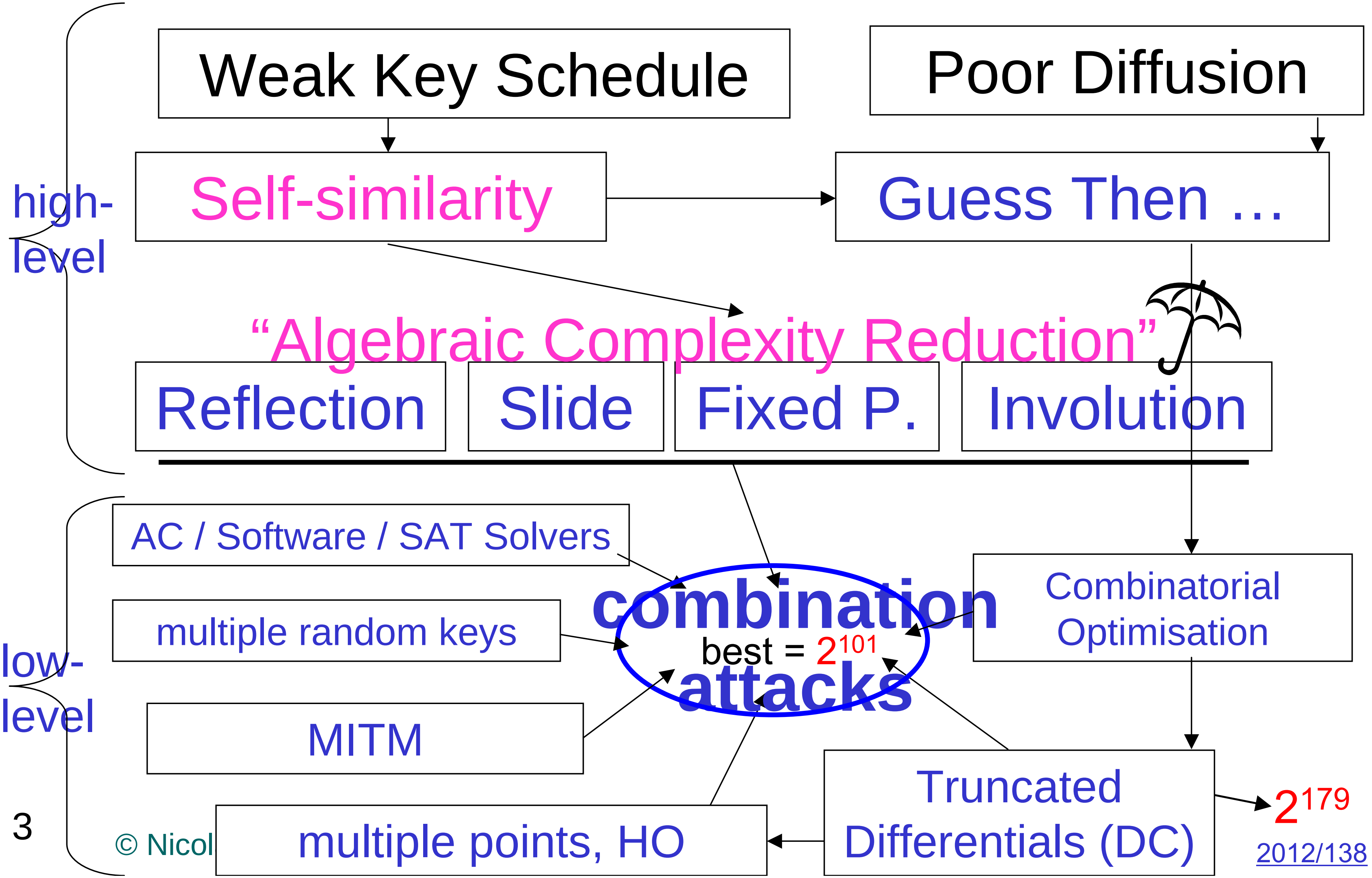
2. GOST: Russian encryption standard

3. GOST submission to ISO in 2010

4. How GOST can eventually be broken…

What's Wrong? >50 distinct attacks… Best = 2^{101}

cf. 2011/626



high-level

Weak Key Schedule

Poor Diffusion

Self-similarity

Guess Then …

"Algebraic Complexity Reduction"

Reflection

Slide

Fixed P.

Involution

low-level

AC / Software / SAT Solvers

multiple random keys

MITM

multiple points, HO

combination attacks

best = 2^{101}

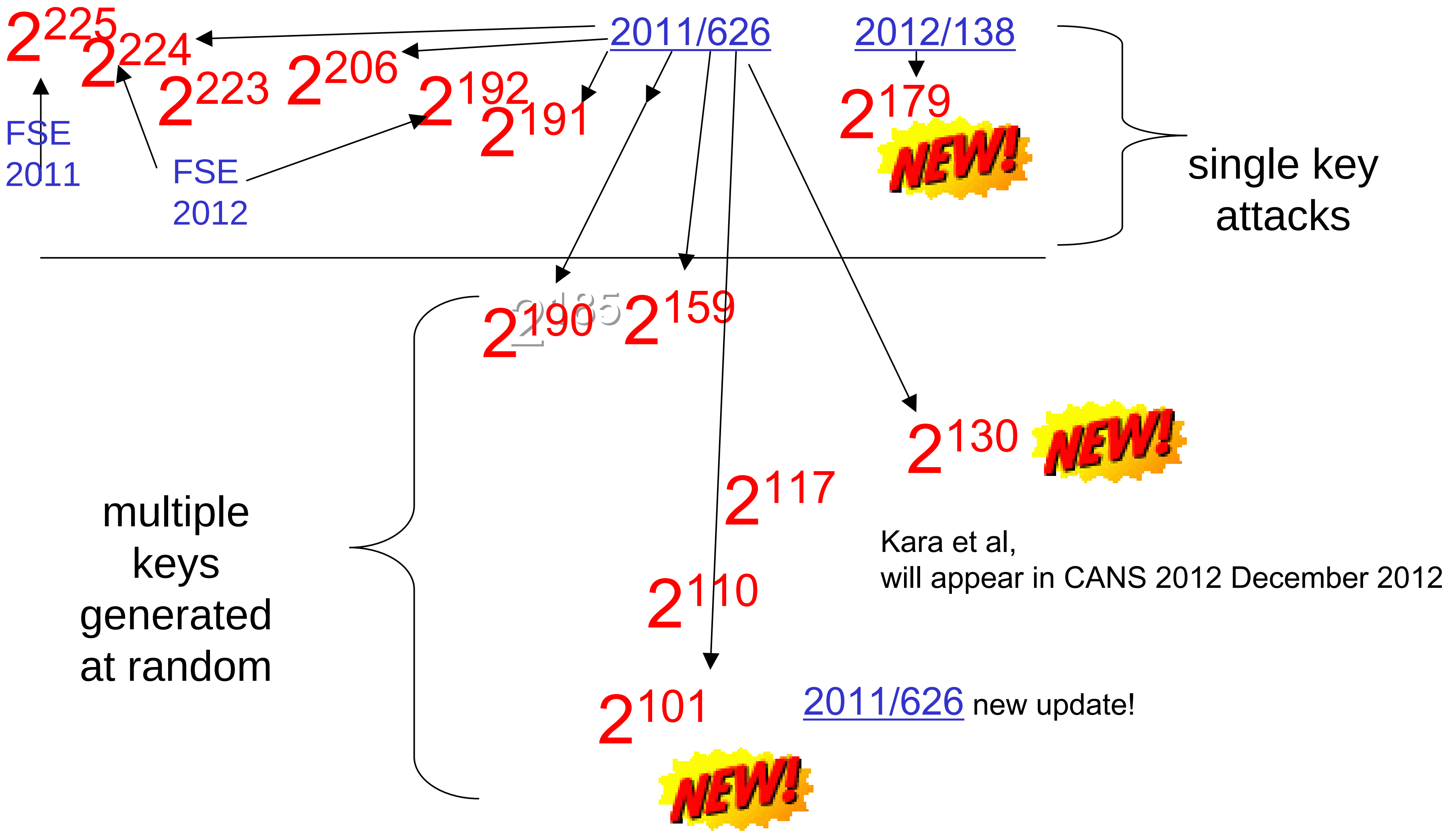
Combinatorial Optimisation

Truncated Differentials (DC)

2^{179}

2012/138

Development History



2^{225} 2^{224} 2^{223} 2^{206} 2^{192} 2^{191}

FSE 2011

FSE 2012

2011/626

2012/138

2^{179}

single key attacks

2^{190} 2^{185} 2^{159}

2^{130}

2^{117}

2^{110}

2^{101}

multiple keys generated at random

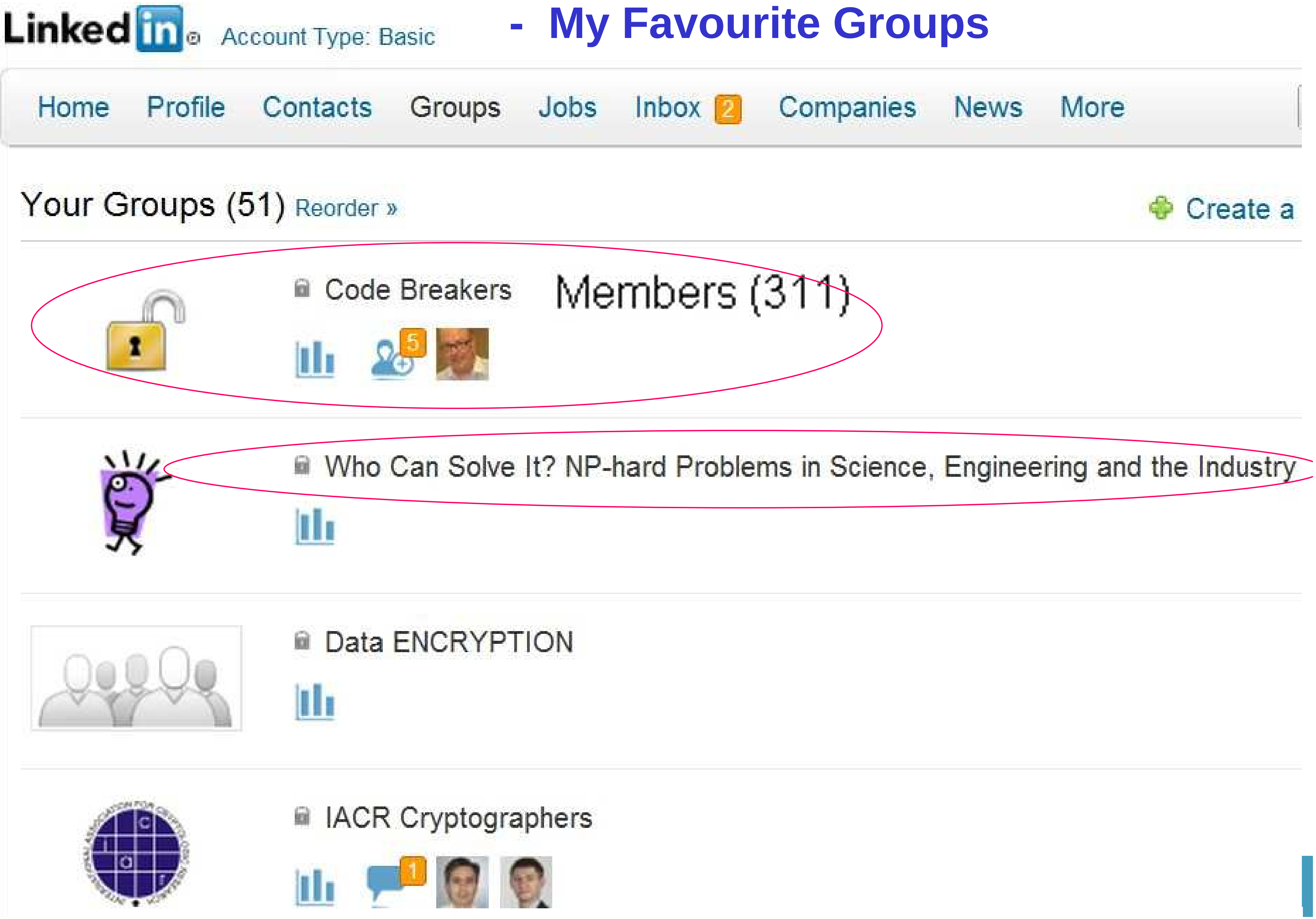
Kara et al,
will appear in CANS 2012 December 2012

2011/626 new update!

History: Cold War
Russia vs. USA

Как американцы контролируют территорию России
Космическая разведка США
Космическая разведка США
Космическая разведка США
Зоны радиолокационных полей
Зоны радиолокационных полей
Россия
Москва
США
Ракетоопасные направления
Позиции противоракет США
Радиолокационные станции дальнего обнаружения (США)
Стратегические бомбардировщики РФ
Станции предупреждения о ракетном нападении (Россия)
Стационарные (шахтные) межконтинентальные баллистические ракеты (МБР) России
Мобильные МБР России
Атомные подлодки ВМФ России с МБР

- My Favourite Groups



Account Type: Basic

Your Groups (51)

Code Breakers Members (311)

Who Can Solve It? NP-hard Problems in Science, Engineering and the Industry

Data ENCRYPTION

IACR Cryptographers

Russian Subtitles On:

code breakers ==

взломщики кодов

Cryptanalysis

from Greek

- kryptós, "hidden“
- analýein, "to untie“

Term coined in 1920

by William F. Friedman.

- Born in Moldavia
- Chief cryptologist at NSA,1950s.

History: 1918

- Tzarist secret services

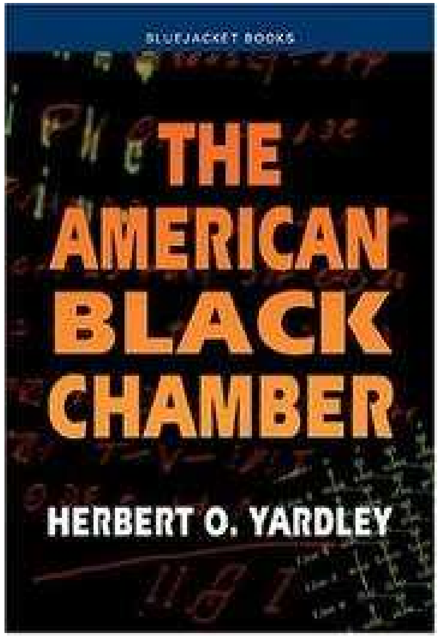
  => continued their work with the armies of white generals.

- In 1918 - 1920 almost all encrypted correspondence of the Soviet Army and Government was easily broken by
  - the white (counterrevolutionary) armed forces
  - the British
  - the Swedish
  - the Polish: broke key messages and won the War against Russia in 1920-1921

1930

1930: Russian code breaker Bokiy broke a U.S. code.

- US ciphers were really not good at that time…
  - In 1929 US government disbanded its Federal crypto services because… “Gentlemen don’t read each other’s mail”…



Fialka = Фиалка = Violet = M-125

Around 1965.

MUCH stronger than Enigma…

Used until 1987 in East Germany…

Fialka Versions

- Each country of the Warsaw pact had their own version
- Different keyboard, different fonts…
- Different SECRET set of 10 wheels.

Cold War Soviet Cryptanalysis

- Soviet Union was breaking codes and employed at least 100 cryptologists…

  [Source: Cryptologia, interviews by David Kahn with gen. Andreev=first head of FAPSI=Russian NSA]

Example: In 1967 GRU (Soviet Intelligence) was intercepting cryptograms from 115 countries, using 152 cryptosystems, and among these they broke 11 codes and “obtained” 7 other codes.

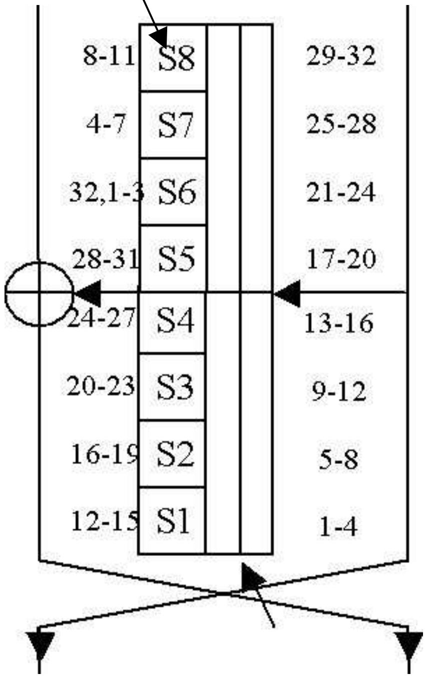
Was Fialka Broken?

- Israel have captured Fialka machines during the 6-day war in 1967 and … nothing more was disclosed.
- Austria would intercept and decrypt a fair proportion of Fialka traffic during the Cold War…
- In the 1970s the NSA would build a supercomputer to decrypt Fialka routinely

Secret Specs: ROTORS vs. S-boxes

FIALKA

GOST



Compare: Rotors of Enigma [1930s]

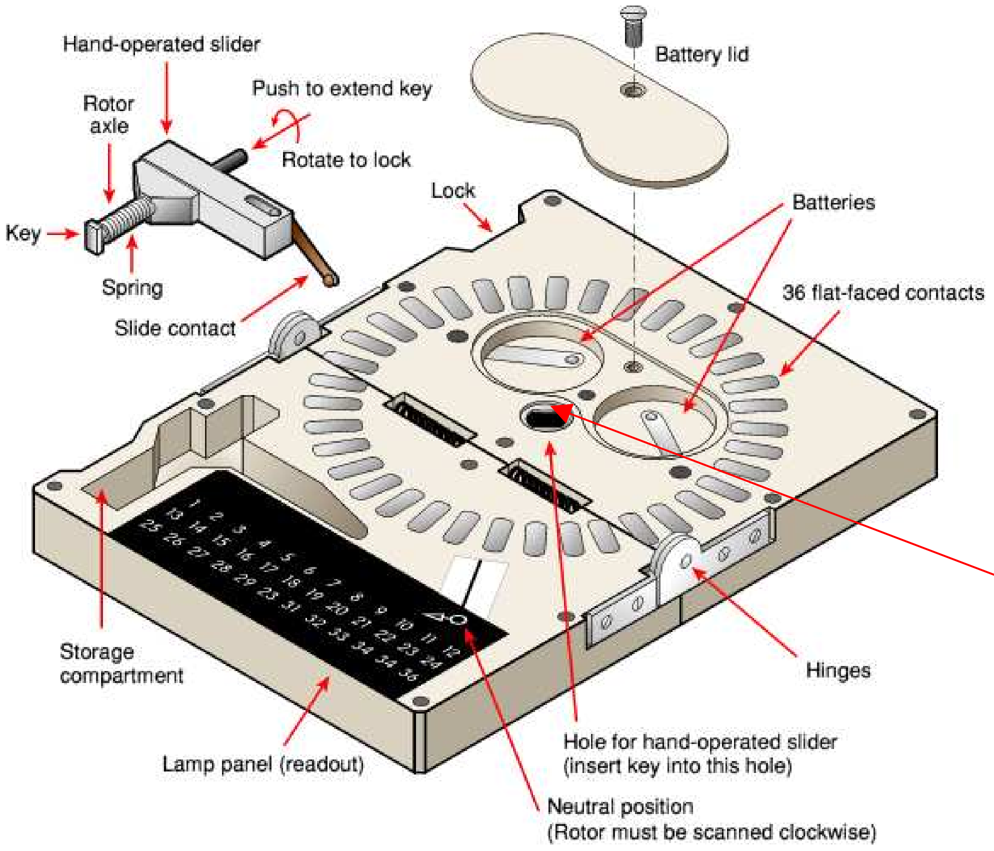
- The specs of Enigma were reverse-engineered by the Polish in early 1930s in tight collaboration with French intelligence… [and the British].
- Finding the rotors by Marian Rejewski was much harder than daily code breaking at Bletchley Park…

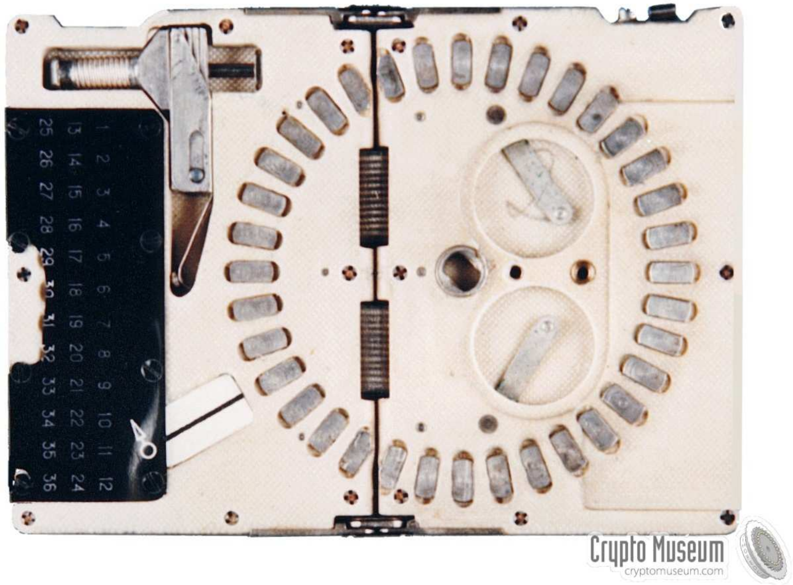
US Ciphers

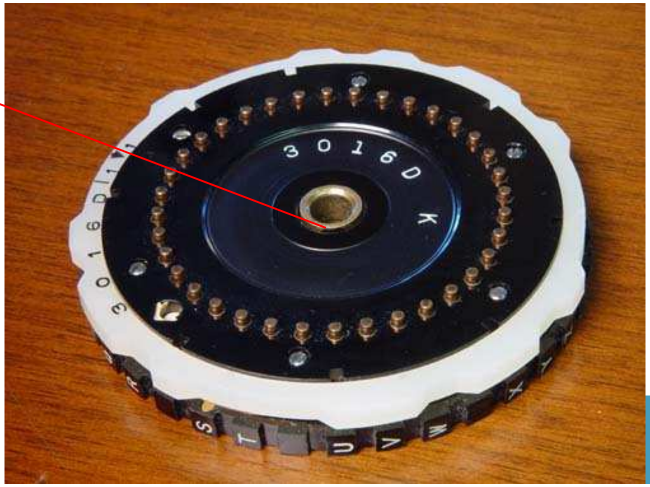
- US/NATO:
  Russia broke the NATO KW-7 cipher machine
  - the NSA did not see it was weak…
  - The spec became known because of a spy ring
    - by John A .Walker Jr + family.
    - was paid more than 1M USD (source: NSA)
    - to this day the spec has NOT been made public
  - greatest exploit in KGB history,
  - allowed the Soviet Union to “read millions” of American messages [1989, Washington Post]

Walker Amazing Machine

Walker obtained from the KGB a pocket machine to read the connections of rotors of KL-7







Modern Cryptanalysis

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

Linear and differential cryptanalysis usually 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) ?

LOW DATA CRYPTANALYSIS NEW!

Two Worlds:

- **The “approximation” cryptanalysis:**
  - Linear, differential, approximation, attacks etc..
  - based on probabilistic characteristics
    - true with some probability.
  - consequently, the security will grow exponentially with the number of rounds, and so does the number of required plaintexts in the attacks
    - main limitation in practice.
- **The “exact algebraic” approach:**
  - Write equations to solve, true with probability 1.

    => Low data complexity

Algebraic Attacks on Block Ciphers

1. Write +
2. Solve [key recovery].

Algebraic Attacks on Block Ciphers

Gröbner Bases:

- Optimising the expansion step 2. at high degree.
- Mostly the dense case is understood and implemented.
- Then either AES-128 is broken at up to say 4 [Gwenolé Ars thesis: maybe it is?]. AND if not at this degree, it must be secure (!).

Fast Algebraic Attacks [will just explain]:

- Avoid expansion, start with BIGGER initial systems but never allow any expansion or increase in the degree.
- Sparse case ! Essential problems: preserve sparsity.

2. Fast Algebraic Attacks On Block Ciphers

Fast Algebraic Attacks on Block Ciphers

Definition [informal on purpose] Methods to lower the degree of equations that appear throughout the computations… [e.g. max deg in F4]

(more generally need to substantially lower the memory requirements of algebraic attacks compared to their running time).

⇒ Very rich galaxy of attacks to be studied in the next 20 years…

How to lower the degree ?

- by having several P/C pairs (bigger yet much easier !)
- by CPA, CPCA, etc…
- by fixing internal variables (Guess-then-Algebraic).
- by finding [approximate] equations on bigger blocks
  - by interpolation [cf. W. Meier's talk]
  - by guessing equations that have strong bias
    - Linear-Algebraic or Bi-Linear-Algebraic Cryptanalysis
    - Differential-Algebraic.
- by clever choice of representation
- by introducing new variables (oh yes !)
- by having a larger key
- new tricks to be invented ?

cumulative effect !!!

3. Solving Methods…

3.3. ElimLin – The Most Surprising.

Complete description:

- Find linear equations in the linear span.
- Substitute, and repeat.

Amazingly powerful…

3.3. ElimLin – Remark:

In a way it is:

Doing things which Gröbner bases usually ignore or do not care about at ”degree 1.05” …

(very small number of higher-degree monomials).

3.4. ANF-to-CNF - The Outsider

Before we did try,

we actually never believed it could work…

☺ ☺ ☺

Convert MQ to a SAT problem.

(both are NP-hard problems)

3.4. ANF-to-CNF - The Outsider

Principle 1:

each monomial = one dummy variable.

$$a = wxyz$$

$$\Updownarrow$$

$$a \iff (w \wedge x \wedge y \wedge z)$$

$$\Updownarrow$$

$$(w \vee \bar{a})(x \vee \bar{a})(y \vee \bar{a})(z \vee \bar{a})(a \vee \bar{w} \vee \bar{x} \vee \bar{y} \vee \bar{z})$$

d+1 clauses for each degree d monomial

Also

Principle 2:

Handling XORs – Not obvious. Long XORs known to be hard problems for SAT solvers.

$$a \oplus b \oplus c \oplus d = 0$$

$$\Updownarrow$$

$$(\bar{a} \vee b \vee c \vee d)(a \vee \bar{b} \vee c \vee \bar{d})(a \vee b \vee \bar{c} \vee d)(a \vee b \vee c \vee \bar{d})$$

$$(\bar{a} \vee \bar{b} \vee \bar{c} \vee d)(\bar{a} \vee \bar{b} \vee c \vee \bar{d})(\bar{a} \vee b \vee \bar{c} \vee \bar{d})(a \vee \bar{b} \vee \bar{c} \vee \bar{d})$$

- Split longer XORs in several shorter with more dummy variables.
- About 4 h clauses for a XOR of size h.

ANF-to-CNF

This description is enough to produce a working version.

Space for non-trivial optimisations. See:

Gregory V. Bard, Nicolas T. Courtois and Chris Jefferson:

“Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers”.

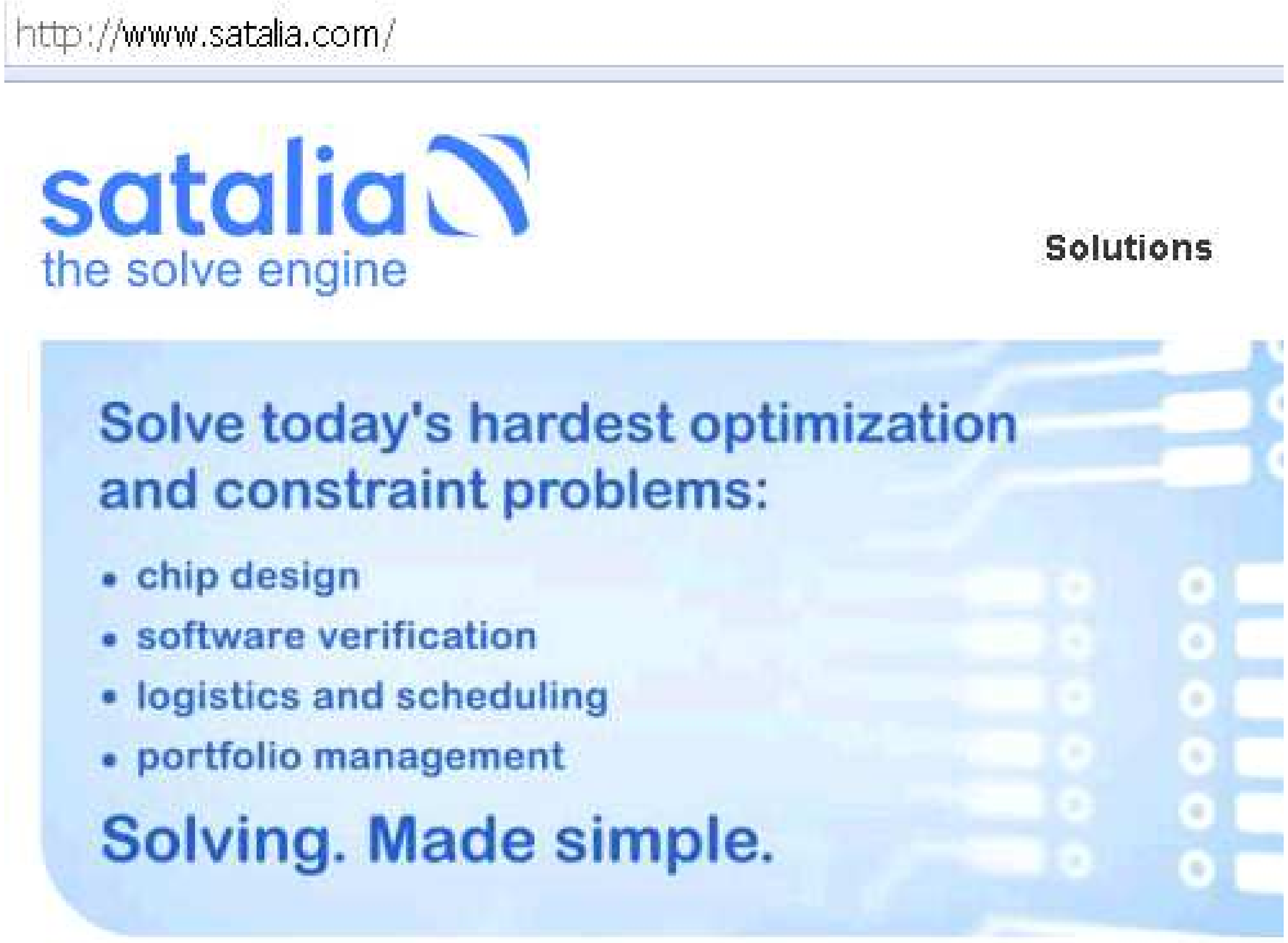
Ready Software

Several ready programs to perform this conversion are made available on this web page:

www.cryptosystem.net/aes/tools.html

SAT Solvers in the Cloud

UCL spin-off company



http://www.satalia.com/

satalia

the solve engine

Solutions

Solve today's hardest optimization and constraint problems:

- chip design
- software verification
- logistics and scheduling
- portfolio management

Solving. Made simple.

solving SAT problems on demand…

commercial but also for free…

Solving SAT

What are SAT solvers?

Heuristic algorithms for solving SAT problems.

- Guess some variables.
- Examine consequences.
- If a contradiction found, I can add a new clause saying "In this set of constraints one is false".

Very advanced area of research.

Introduction for "dummies":

Gregory Bard PhD thesis.

MiniSat 2.0.

Winner of SAT-Race 2006 competition.

An open-source SAT solver package,
by Niklas Eén, Niklas Sörensson,

More recent version [2012]:
CryptoMiniSat 2.92.
improved by Mate Soos,
added also some linear algebra…

Ready Software for Windows

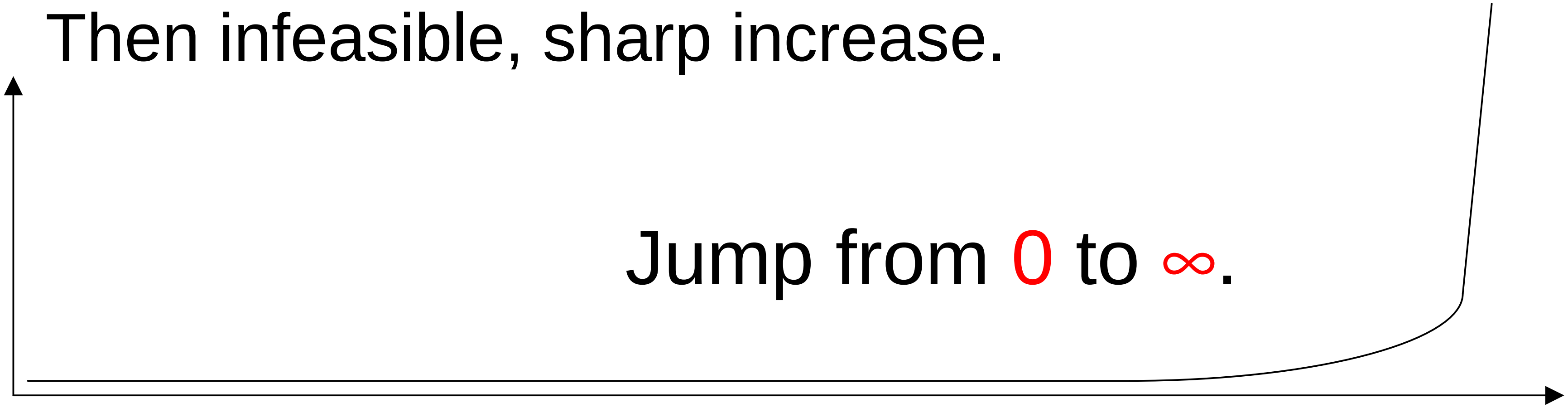
Ready programs:

www.cryptosystem.net/aes/tools.html

ANF-to-CNF + MiniSat 2.0.

Gives amazing results in algebraic cryptanalysis of just any (not too complex/not too many rounds) cipher, cf. (VSH). Also for random sparse MQ.

- Certain VERY large systems solved in seconds on PC (thousands of variables !).
- Few take a couple hours/days…
- Then infeasible, sharp increase.

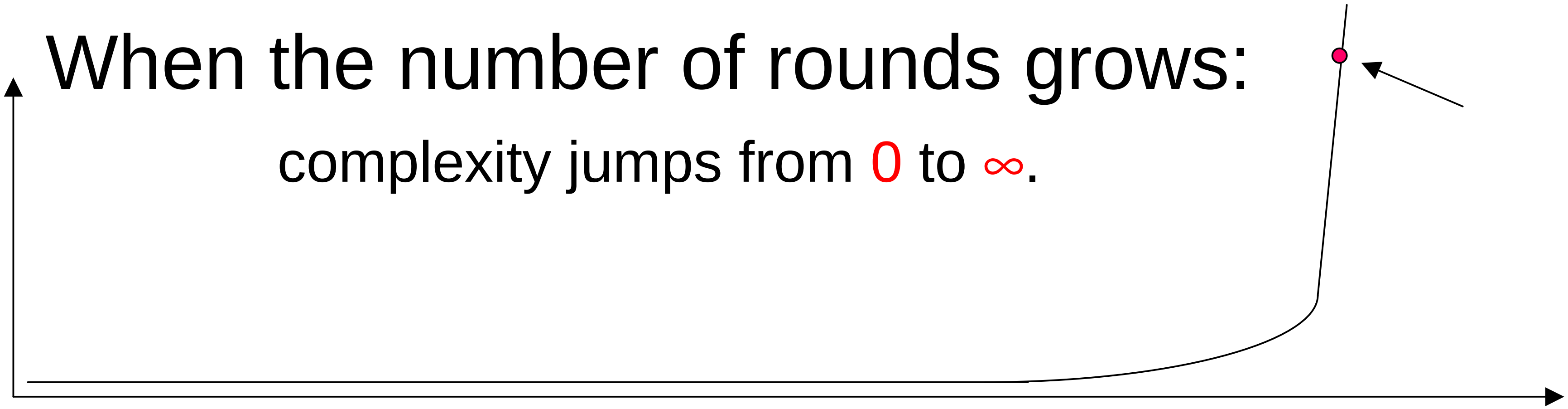


Jump from 0 to ∞.

What Are the Limitations of Algebraic Attacks ?

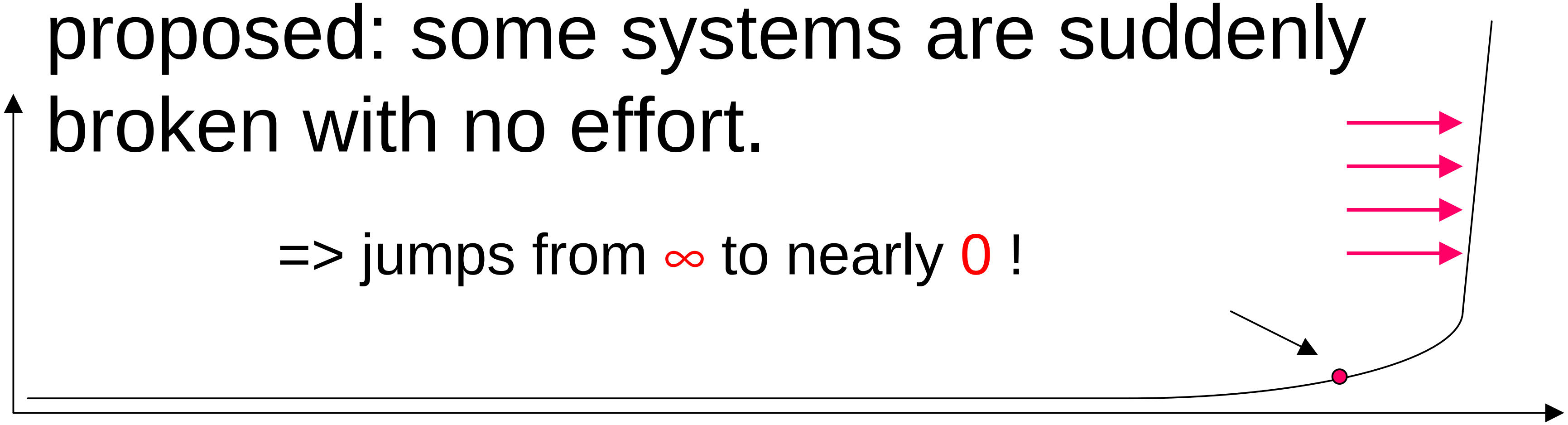
- When the number of rounds grows:

  complexity jumps from 0 to ∞.



- With new attacks and new “tricks” being proposed: some systems are suddenly broken with no effort.

  => jumps from ∞ to nearly 0 !



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: (IOH)

Algebraic I/O relations.

Theorem [Courtois-Pieprzyk]:

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

=>3 for DES.

Idea 2: (VSH)

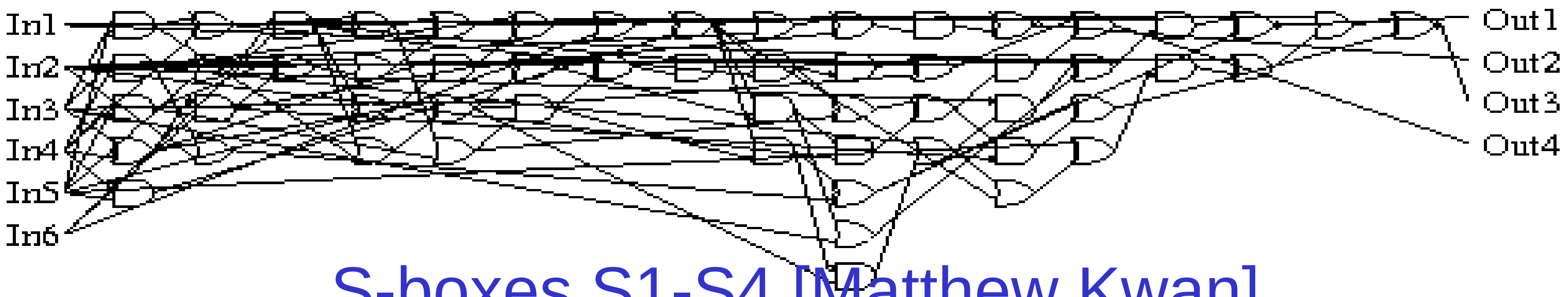
DES has been designed to be implemented in hardware.

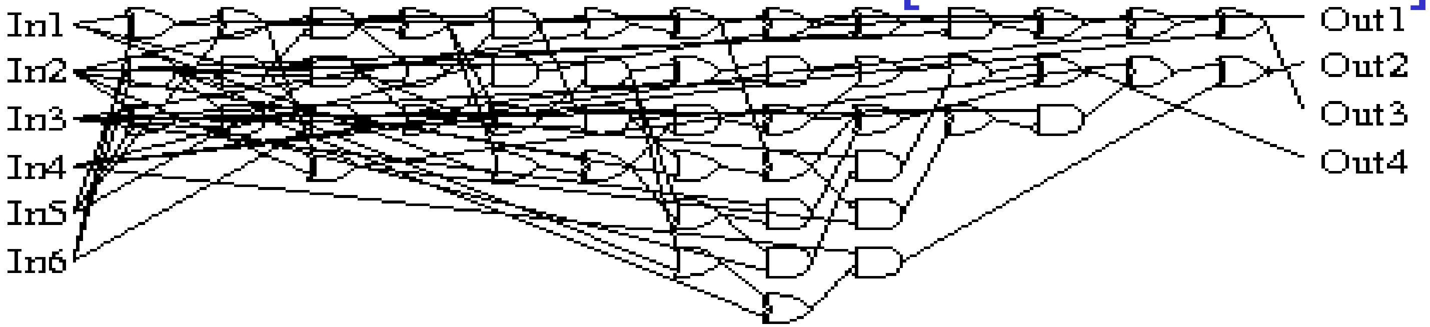
=> Very-sparse quadratic equations at the price of adding some 40 new variables per S-box.

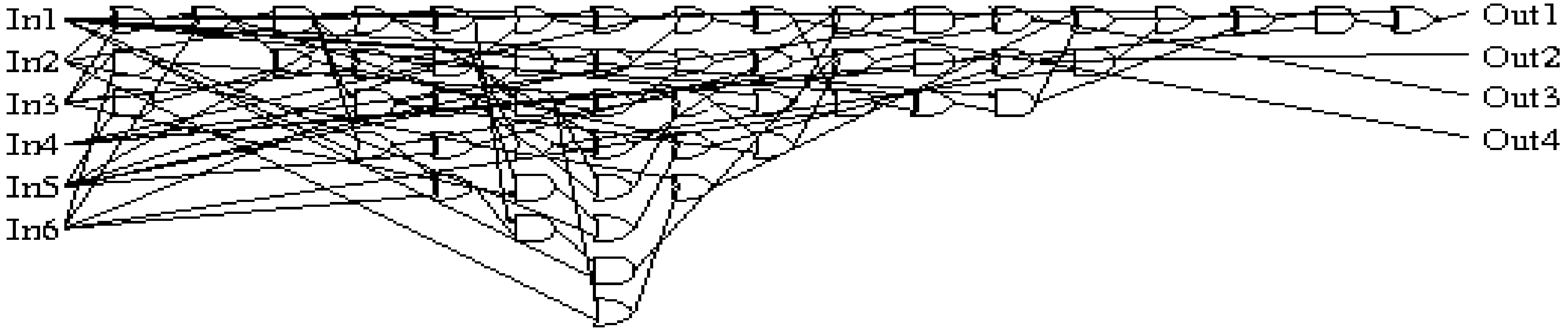
Results ?

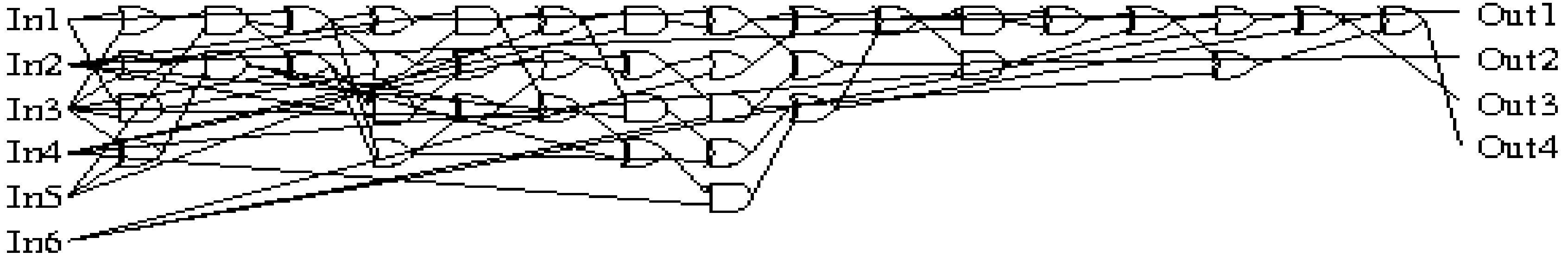
Both Idea 1 (IOH) and Idea 2 (VSH)
(and some 20 other I have tried…)
can be exploited in working
key recovery attacks.

S-boxes S1-S4 [Matthew Kwan]

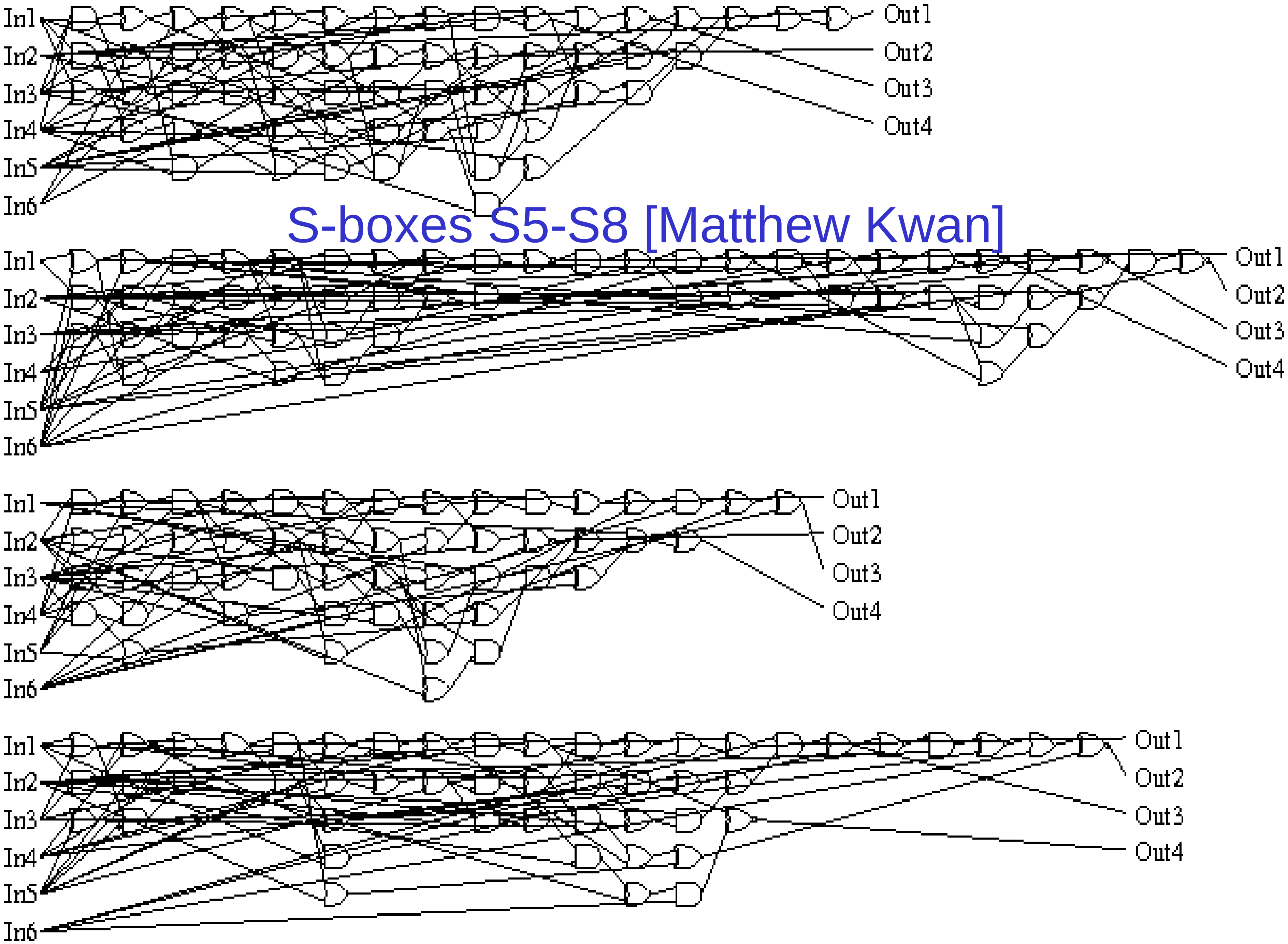








S-boxes S5-S8 [Matthew Kwan]



I / O Degree

Consider function $f : GF(2)^n \rightarrow GF(2)^m$, $f(x) = y$, 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.

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/

What Can Be Done ?

Idea 1 (Cubic IOH) + ElimLin:

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

- When 23 variables fixed, takes 173 s.
- Magma crashes > 2 Gb of RAM.

Idea 2 (VSH^{40}) + ANF-to-CNF + MiniSat 2.0.:

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

- Fix 20 variables takes 68 s.
- Magma crashes with > 2 Gb.

And GOST?

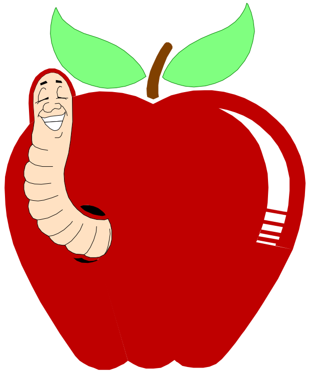
Essentially the same software methods…

well, actually with a lot of non-trivial super-compact representation and circuit optimisation work, cf. our paper at http://2012.sharcs.org/record.pdf.

… allow also to break up to 8 rounds of GOST…

Can we hope to break 32 rounds?

4. Self Similarity

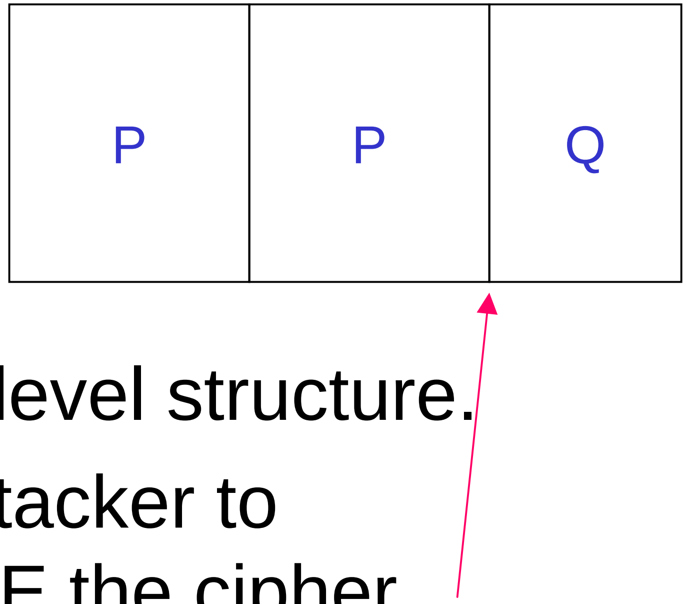


or What's Wrong With Some Ciphers

KEY IDEA

REDUCE the complexity.
For example:

REDUCE the number of rounds.



How? Use self-similarity and high-level structure.

Magic process which allows the attacker to guess/determine values INSIDE the cipher.

We now call it Algebraic Complexity Reduction
[Courtois 2011]

4.1. Crypto-1 Cipher

Waste of Silicon

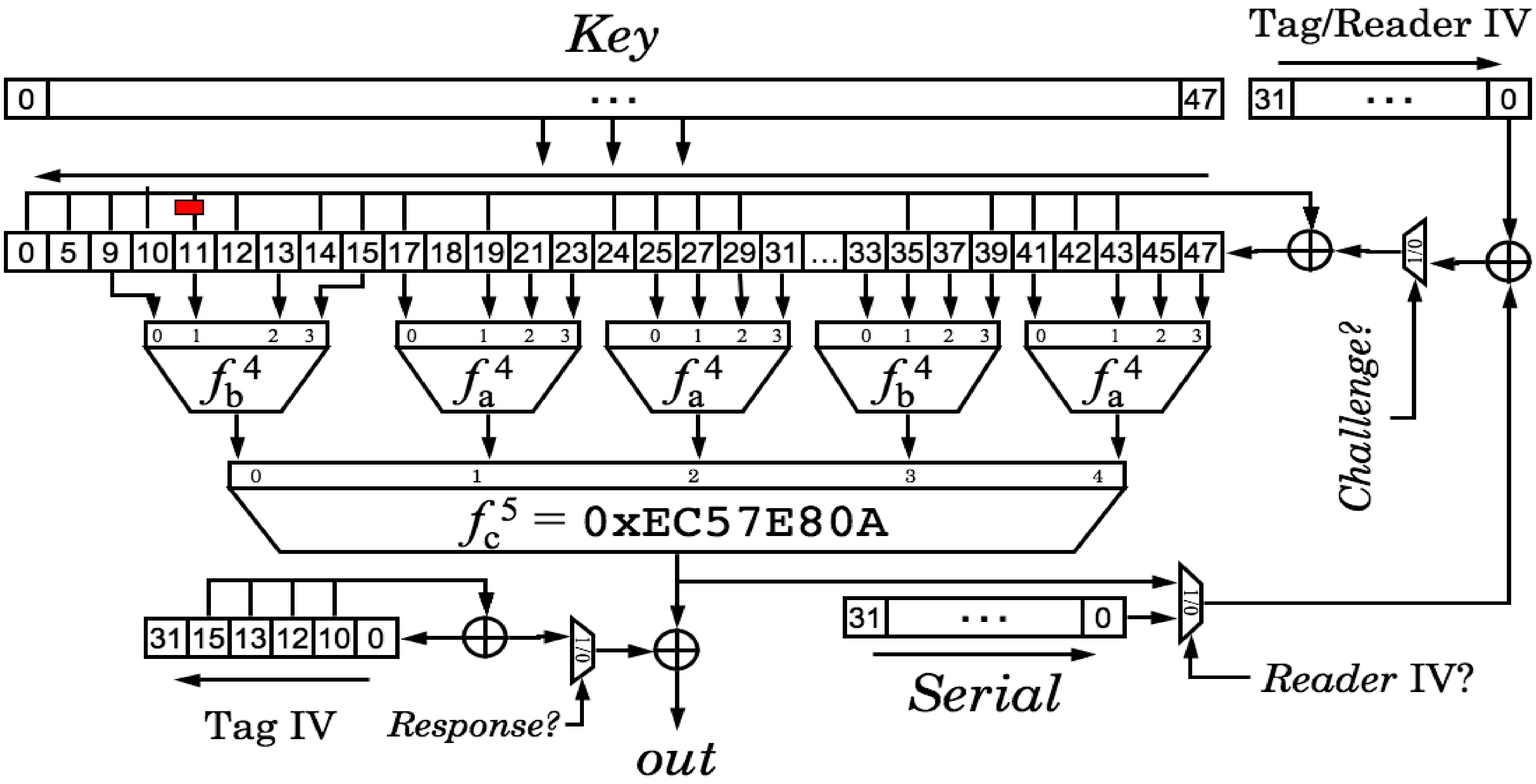
MiFare was manufactured by Philips, now NXP, and licensed to Infineon.

BUT, even a hardware or software designer would NOT notice how weak the cipher is.

Camouflage?

Due to a combination with another terrible weakness half of the silicon is wasted…

Crypto1 Cipher



f_a^4 = 0x9E98 = (a+b)(c+1)(a+d)+(b+1)c+a

f_b^4 = 0xB48E = (a+c)(a+b+d)+(a+b)cd+b

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

Waste of Silicon

Internal bits are computed 2-3 times.

One could save half of the gates!

“Courtois Dark Side” Attack on MiFare Classic

Cf. eprint.iacr.org/2009/137. Basic Facts:

It is a multiple differential attack.

Form of multiple “self-similarity” as well..

I exhibit a differential that

- holds simultaneously for 256 differentials this works with probability of about 1/17.
- for 8 differentials the probability is about 0.75 (!!).

Both are differences on 51 bits of the state of the cipher.

A VERY STRONG property(!).

Summary

- We broke >1 billion smart cards covering 70 % of the contactless badge/ticketing market.
- Our attack require more than 10 times less data than the Dutch attacks about which there were 10 000 press reports…

- Security of many buildings (banks, military, UK Cabinet Office) is badly compromised.
- Security of many transport [metro,bus] and parking cards worldwide is badly compromised.
- Property and important assets [e;g. government and financial data] are directly under threat.

4.3. Self-Similarity and KeeLoq

KeeLoq

- Designed in the 80's by Willem Smit.
- In 1995 sold to Microchip Inc for more than 10 Million of US$.

??

How Secure is KeeLoq

According to Microchip, KeeLoq should have ``a level of security comparable to DES''. Yet faster.

Miserably bad cipher, main reason:

its periodic structure: cannot be defended. The complexity of most attacks on KeeLoq does NOT depend on the number of rounds of KeeLoq.

Notation

f_k() – 64 rounds of KeeLoq

g_k() – 16 rounds of KeeLoq, prefix of f_k().

We have: E_k = g_k o f^8_k.

528 = 16+8*64 rounds.

4.4. Sliding Properties of KeeLoq

[and one simple attack from FSE 2008]

Sliding Attacks – 2 Cases

- **Complete periodicity [classical].**

P	P	P

- **Incomplete periodicity [new] – harder.**

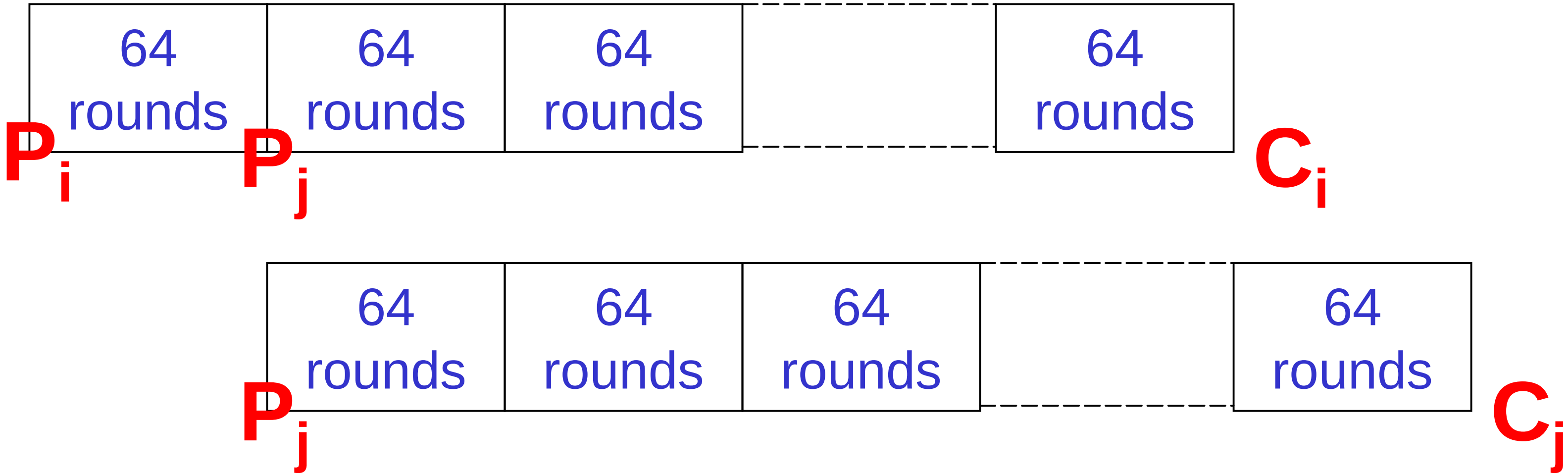
P	P	P	Q

  – **KeeLoq: Q is a functional prefix of P. Helps a lot.**

Sliding Attacks

Classical Sliding Attack [Grossman-Tuckerman 1977]:

- Take $2^{n/2}$ known plaintexts (here n=32, easy !)
- We have a “slid pair” (P_i,P_j) s.t.

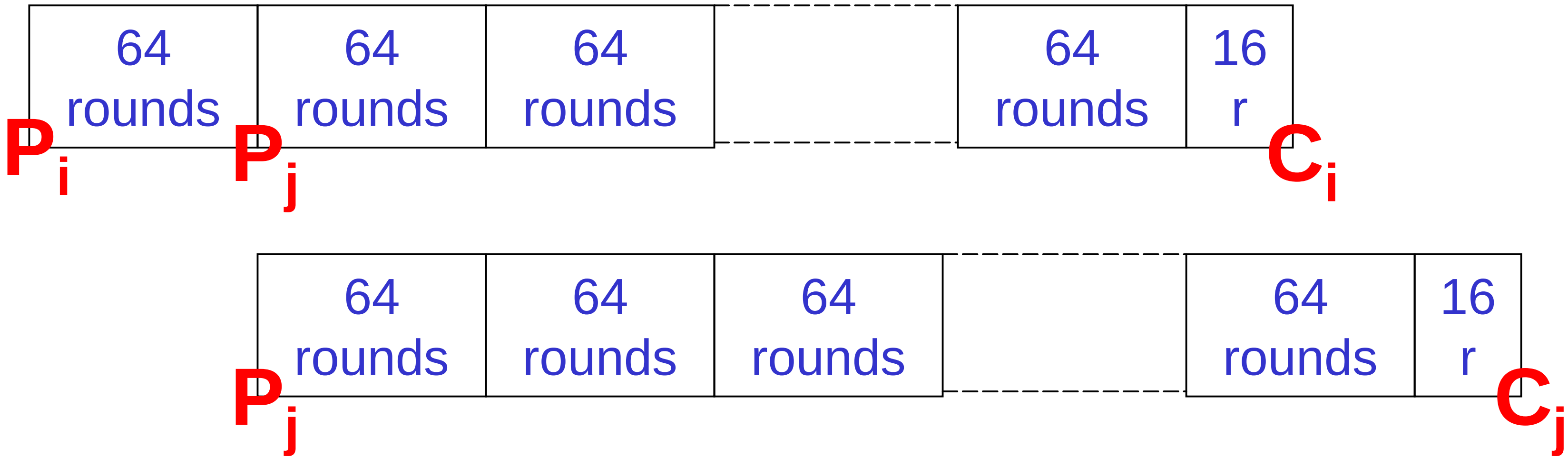


Gives an unlimited number of other sliding pairs !!!

KeeLoq and Sliding

Apply Classical Sliding? Attack 1.

- Take $2^{n/2}$ known plaintexts (here $n=32$, easy !)
- We have a "slid pair" (P_i,P_j) s.t.

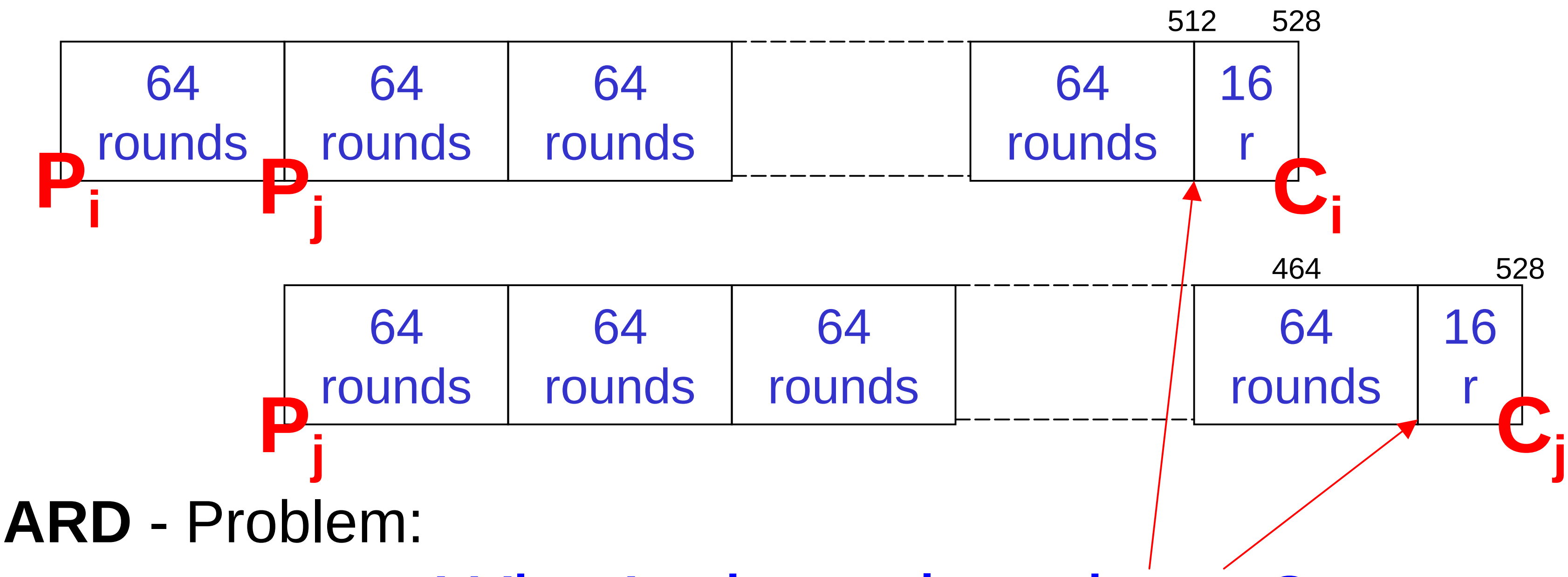


Classical sliding fails – because of the "odd" 16 rounds:

Classical Sliding –Not Easy

Classical Sliding Attack [Grossman-Tuckerman 1977]:

- Take $2^{n/2}$ known plaintexts (here n=32, easy !)
- We have a "slid pair" (P_i,P_j).

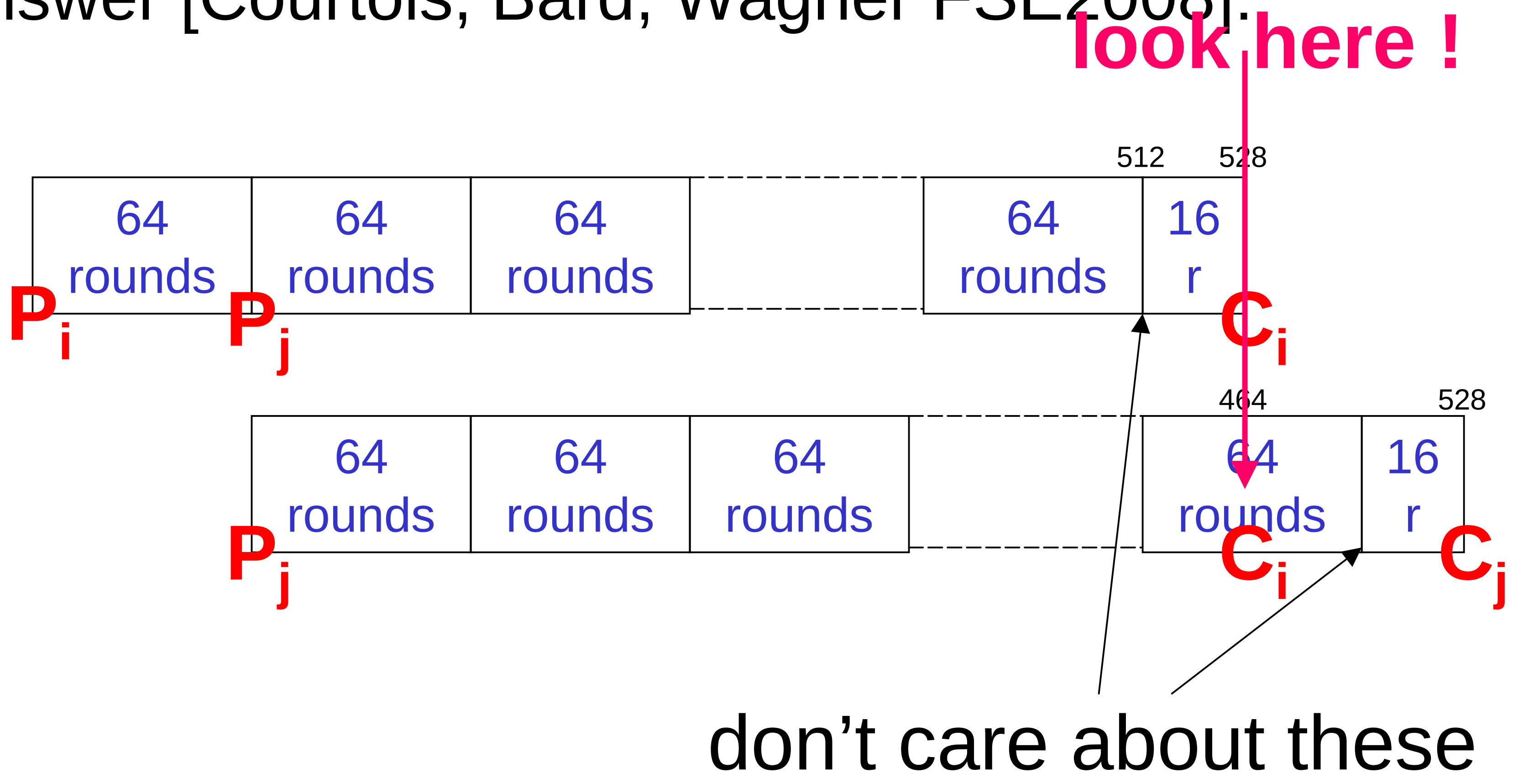


HARD - Problem:

What's the values here ?

Algebraic Sliding

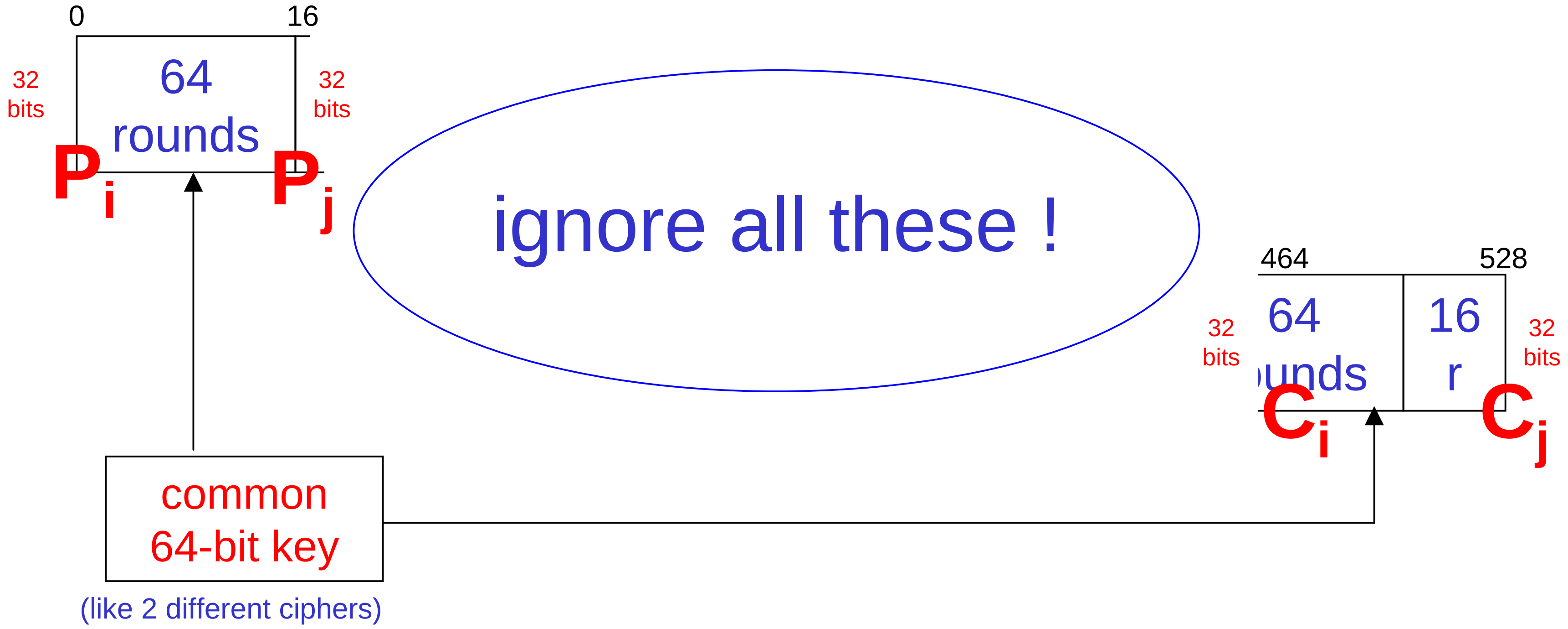
Answer [Courtois, Bard, Wagner FSE2008]:



Algebraic Attack [FSE 2008]

We are able to use C_i,C_j directly !

Write and merge 2 systems of equations:



System of Equations

64-bit key. Two pairs on 32 bits.
Just enough information.

Attack:

- Write an MQ system.
  - Gröbner Bases methods – miserably fail.
- Convert to a SAT problem
  - [Cf. Courtois, Bard, Jefferson, eprint/2007/024/].
- Solve it.
  - Takes 2.3 seconds on a PC with MiniSat 2.0.

Attack Summary:

Given about 2^{16} KP.

We try all 2^{32} pairs (P_i,P_j).

- If OK, it takes 2.3 seconds to find the 64-bit key.
- If no result - early abort.

Total attack complexity about 2^{64} CPU clocks which is about 2^{53} KeeLoq encryptions.

4.6. Snow 2.0. Cipher

ISO

- Less than 10 crypto algorithms were ever standardized by ISO. E.g. AES.
- All in ISO 18033.
  - Snow 2.0. is an international standard for stream cipher encryption.
  - In 2010 the Russian National Standard GOST was also submitted to ISO 18033 to become an international standard.

I / O Degree (a.k.a. [Graph] Alg. Immunity)

Consider function $f : GF(2)^n \rightarrow GF(2)^m$, $f(x) = y$, 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

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

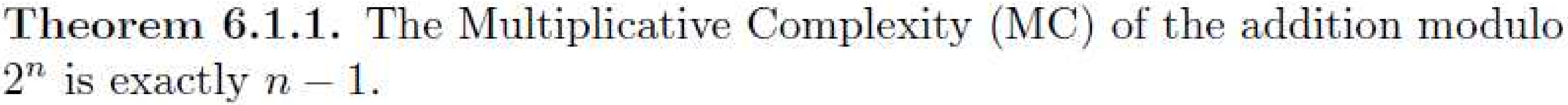
that holds with certainty for every couple (x, y) such that $y = f(x)$.

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 I/O Degree = 2

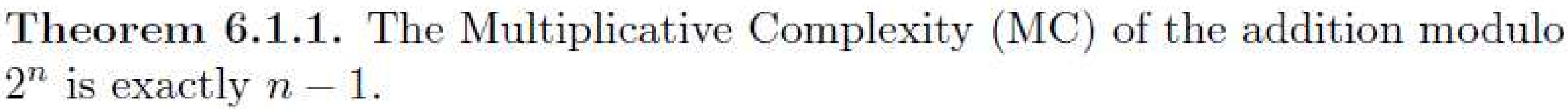
Quadratic. More importantly:
Quadratic I/O without extra variables

(the c_i can be all eliminated)

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

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

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:

$$\boxed{xy + (x + y)c = (x + c)(y + c) - c^2}$$

1x each

$$x_0y_0$$

$$x_1y_1 + (x_1 + y_1)c_1$$

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

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

Conditional A.I. = Conditional I/O Degree

Already exploited by Krause, Armknecht, Fischer and Meier [FSE 2007 and ICALP 2007]

Definition: Let us assume $n > m$. Given some fixed output y, a y-conditional I/O equation for S is a nonzero algebraic equation $r_y(x) = 0$ that holds with probability 1 for every x such that $S(x) = y$.

Given some fixed output y, let d be the minimum degree of a y-conditional I/O equation. The conditional algebraic immunity CAI of S is the minimum of d over all y in $GF(2)^m$.

Conditional Describing Degree

Definition: Given some fixed output y, let d be the minimum degree such that the equation $S(x) = y$ is entirely defined by conditional I/O equations of degree at most d. The minimal d over all y in $GF(2)^m$ is called conditional describing degree (CDD) of S.

This paper:

For + mod 2^n: We show that:

- The Conditional Describing Degree is 1 (!)
- Is it trivial? Well, we know that for minus mod 2^n: consider $x-y=0$.
  - Where (x,y) is the input, 0 is the fixed output.
- NEW: Holds also for + mod 2^n: consider $x+y=111111\ldots111$.

This paper:

For + mod 2^n:

- The Conditional Describing Degree is 1 (!)
- So what?
  - View it as follows: fix n linear equations, get 2n!
- Amplification…

This paper:

Larger Blocks of Snow 2.0.

- However some equations can be more interesting than others.
  - How to generate (lots of) extra degree falls elsewhere, because of the structure of Snow?
  - This is not wishful thinking. We constructed such an attack a particularly good one.

4.7. High-Level Attacks on Snow 2.0.

[Courtois-Debraize ICICS 2008]

Moreover:

- If I have to assume that the output for whole 32-bit + mod 2^n: is one specific value – this will happen with VERY LOW probability.
- We can do much better:

We present a LARGE family of outputs, not only 00000 or 111 for which the + mod 2^n: can be partly linearized.

Interest: we want to fix some WELL CHOSEN bits, determine other.
How? Structure of Snow dictates that.

*BTW: Link to LC

- Is it LC with multiple approximations?
- Not at all, all the equations hold simultaneously.

- Find 1 linear equation true with probability 50 % – trivial, no interest.
- Find 10 that simultaneously hold for 50 % o inputs of this S-box/operation. Very strong and helps AC a lot.

Conditional algebraic attacks:

Amplification:

- given n linear assumptions, get C*n consequences.
  - Find attacks that maximize C!
  - A precise measure of “structural” algebraic vulnerability.

- C=2 for + mod 2^n.
- C=4 for Snow 2.0. Keystream generator.
  - Non-trivial result and method…

Amplification=4 or How to Linearize Snow?

Fix to 0.

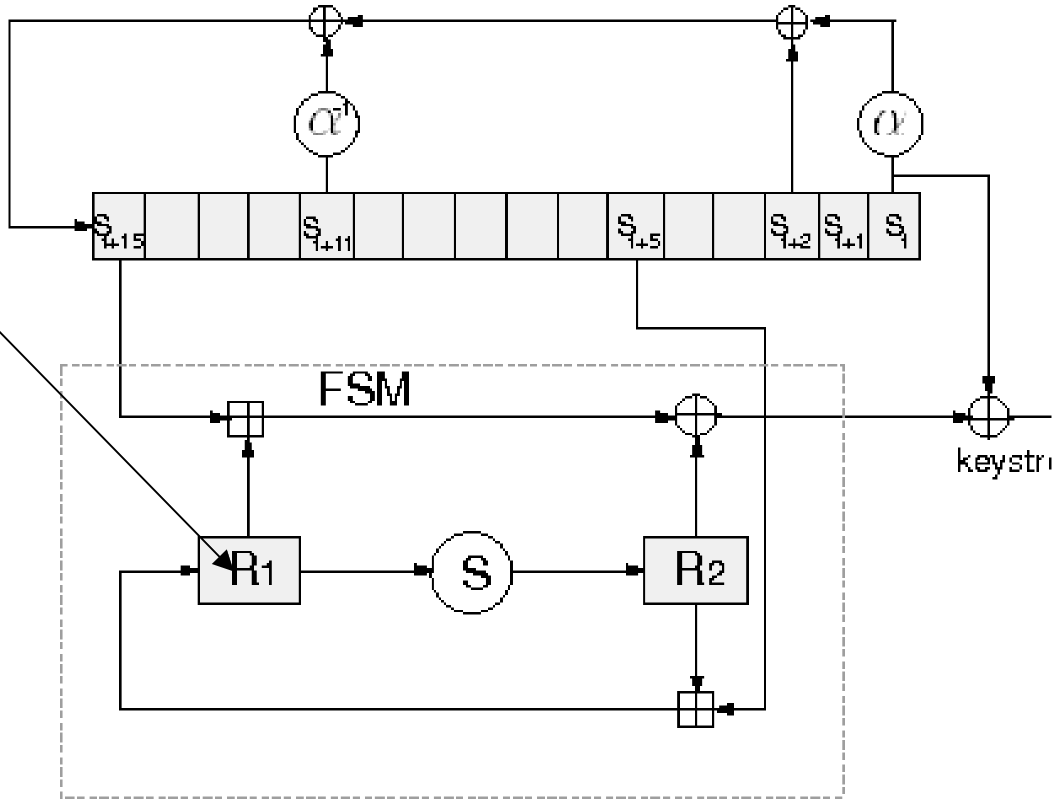
For 9 consecutive steps.

Linearizes both +!

And the S-box layer

n -> 4n equations.

Seems optimal.



5. GOST Cipher

GOST 28148-89

- The Official Encryption Standard of Russian Federation.
- Developed in the 1970s, or the 1980s,
  - First "Top Secret" algorithm.
  - Downgraded to "Secret" in 1990.
- Declassified in 1994.

Why Declassified

- 1994:
  - By mistake???
  - No country ever declassified their national algorithm.

Applications of GOST

– Much cheaper to implement than DES, AES and any other known cipher… (details later).

– Widely implemented and used:
  - Crypto ++,
  - Open SSL,
  - RSA Labs, Etc.
  - Central Bank of Russia,
  - other very large Russian banks..

GOST vs. DES

We hear that: “GOST 28147 “was a Soviet alternative to the United States standard algorithm, DES”

- ???? this is just wrong:
- very long key, 256 bits, military-grade
  - in theory secure for 200 years…
  - not a commercial algorithm for short-term security such as DES…

Can GOST be Used to Encrypt Secret documents?

United States DES
can be used ONLY for unclassified documents.

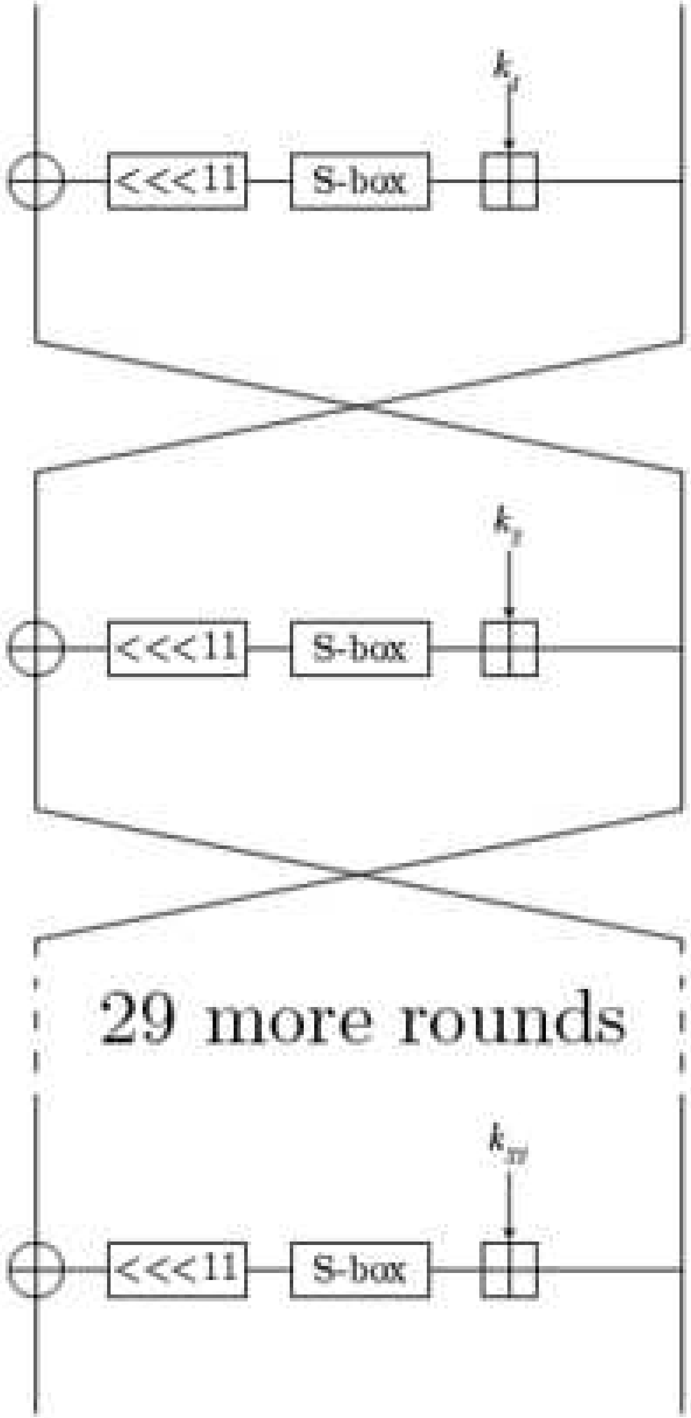
In contrast,

GOST "does not place any limitations on the secrecy level of the protected information".

193.166.3.2/pub/crypt/cryptography/papers/gost/russian-des-preface.ps.gz

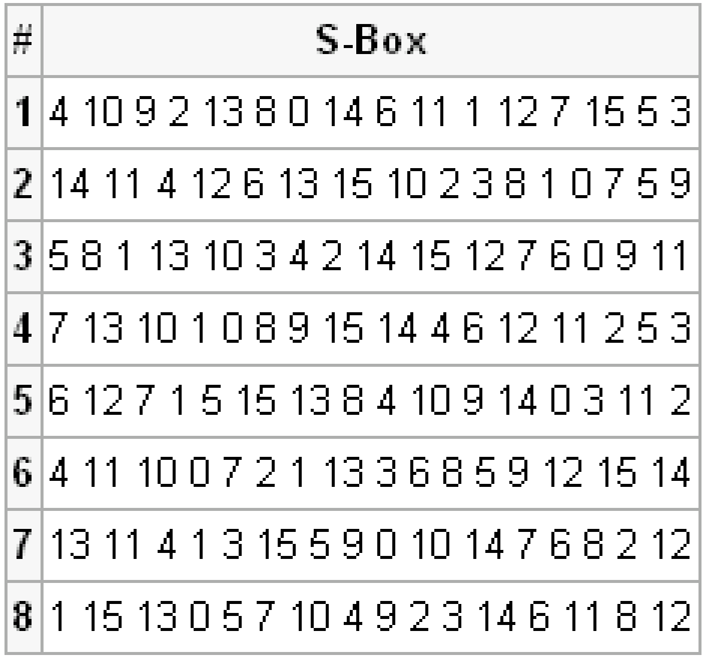
GOST

- Key = 2^{256} initial settings.
- S-boxes = 2^{512} possibilities.
  - But if bijective 2^{354} possibilities.
- Total 2^{610} (or 2^{768}).
  - Compare to 2^{151} possibilities with FIALKA.



GOST Boxes

- 8 secret S-boxes. (354 bits of info)
  - Central Bank of Russia uses these:
- Secret S-boxes are the equivalent of secret rotors in FIALKA



#	S-Box
1	4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3
2	14 11 4 12 6 13 15 10 2 3 8 1 0 7 5 9
3	5 8 1 13 10 3 4 2 14 15 12 7 6 0 9 11
4	7 13 10 1 0 8 9 15 14 4 6 12 11 2 5 3
5	6 12 7 1 5 15 13 8 4 10 9 14 0 3 11 2
6	4 11 10 0 7 2 1 13 3 6 8 5 9 12 15 14
7	13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12
8	1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12

- Our attacks work for any S-boxes but they must be known.
  - there are methods about how to recover the secret S-boxes…

Analysis of GOST

- It was analysed by Schneier, Biham, Biryukov, Dunkelman, Wagner, Pieprzyk, Gabidulin,…
- Nobody found an attack…

*Claims on GOST

Wikipedia April 2011:

Cryptanalysis of GOST

Compared to DES, GOST has a very simple round function. However, the designers of GOST attempted to offset the simplicity of the round function by specifying the algorithm with 32 rounds and secret S-boxes.

Another concern is that the avalanche effect is slower to occur in GOST than in DES. This is because of GOST's lack of an expansion permutation in the round function, as well as its use of a rotation instead of a permutation. Again, this is offset by GOST's increased number of rounds.

There is not much published cryptanalysis of GOST, but a cursory glance says that it seems secure (Schneier, 1996).

The large number of rounds and secret S-boxes makes both linear and differential cryptanalysis difficult. Its avalanche effect may be slower to occur, but it can propagate over 32 rounds very effectively.

[Biryukov, Wagner, Eurocrypt 2000]

“Even after considerable amount of time and effort, no progress in cryptanalysis of the standard was made in the open literature”

More [Biryukov, Wagner, Eurocrypt 2000]

“GOST looks like a cipher that can be made both arbitrarily strong or arbitrarily weak depending on the designer's intent since some crucial parts of the algorithm are left unspecified.”

5.2. GOST on the International Stage

Consensus on GOST Security [2010]

Axel Poschmann, San Ling, and Huaxiong Wang:
256 Bit Standardized Crypto for 650 GE – GOST Revisited,
In CHES 2010

"Despite considerable
cryptanalytic efforts
spent in the past 20 years,
GOST is still not broken."

Security + Implementation Or Why GOST is Very Competitive

Same paper: Axel Poschmann, San Ling, and Huaxiong Wang: 256 Bit Standardized Crypto for 650 GE – GOST Revisited, In CHES 2010

- GOST-PS, fully Russian standard compliant variant using the S-boxes taken from PRESENT cipher:
  - only 651 GE
- The Russian Central Bank version is called GOST-FB,
  - it requires 800 GE
- AES-128
  - requires 3400 GE for a much lower security level!
- DES
  - requires also about 4000 GE…
- PRESENT: 1900 GE for 128-bit version.

in terms of cost/security level claimed GOST is probably strictly the best symmetric cipher known…

GOST and International Standards Organization [ISO]

ISO

- Less than 10 crypto algorithms were ever standardized by ISO. E.g. AES.
- All in ISO 18033.
  - Four 64-bit block ciphers:
    - e.g. TDES
  - Only three 128-bit block ciphers:
    - e.g. AES

GOST in ISO

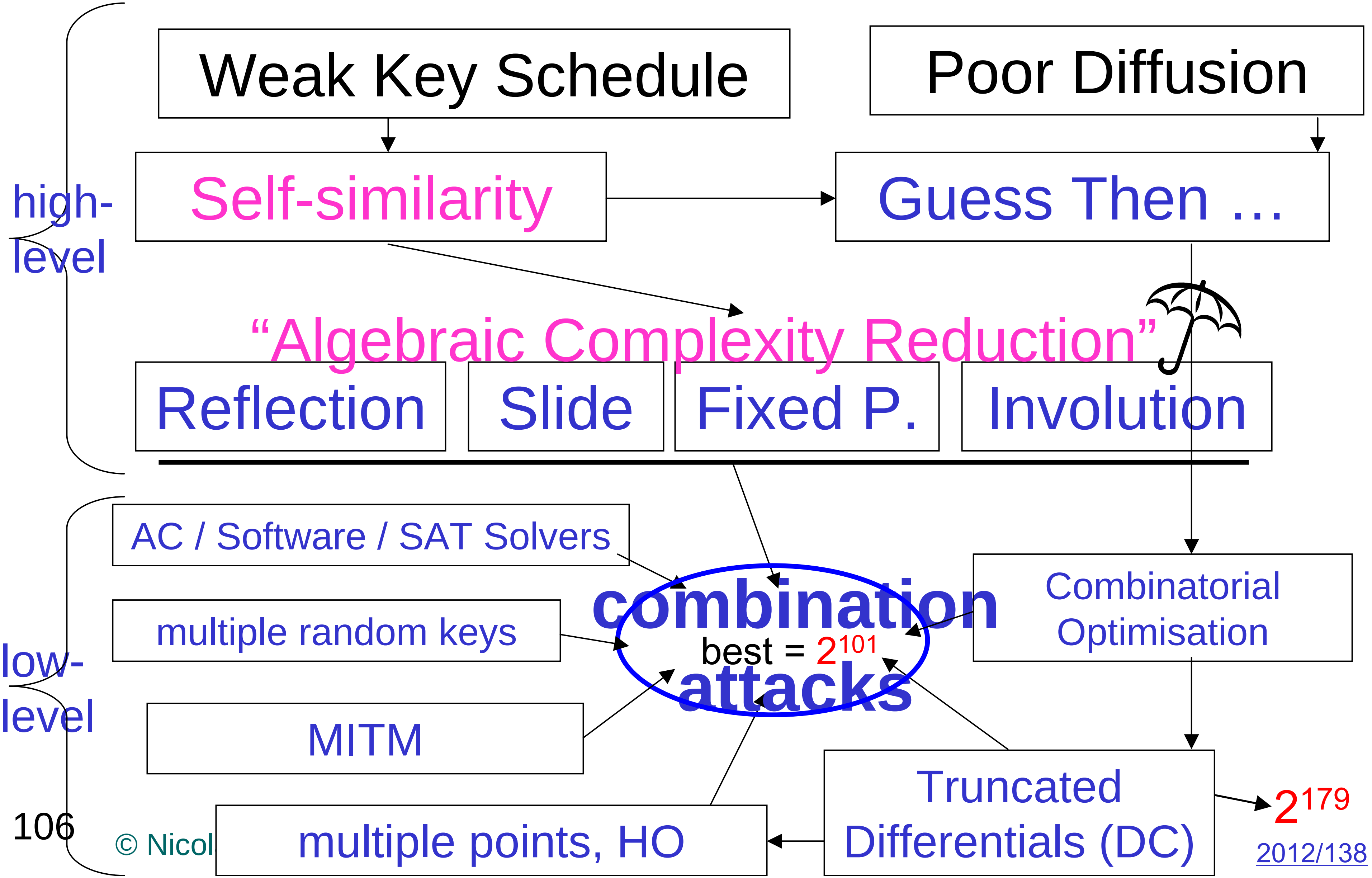
- In 2010 GOST was also submitted to ISO 18033 to become an international standard.
- In the mean time GOST was broken.
- Two attacks were published in early 2011:
  - One by Takanori Isobe [FSE 2011].
  - One by Nicolas Courtois [eprint/2011/211].

Future of GOST in ISO

- Our report [eprint/2011/211] was officially submitted to ISO.
- It says: […] to standardize GOST now would be really dangerous and irresponsible […]
- But Why?
  - Half-broken in very serious sense
  - Really broken in academic sense

What's Wrong? >50 distinct attacks… Best = 2^{101}

cf. 2011/626



high-level

Weak Key Schedule

Poor Diffusion

Self-similarity

Guess Then …

"Algebraic Complexity Reduction"

Reflection

Slide

Fixed P.

Involution

low-level

AC / Software / SAT Solvers

multiple random keys

MITM

multiple points, HO

combination attacks

best = 2^{101}

Combinatorial Optimisation

Truncated Differentials (DC)

2^{179}

2012/138

☂

6. Algebraic Complexity Reduction

[Black Box] Reduction Paradigm

Black-box
- high-level
- guess and determine methods
- which transform
- an attack … into another…

Reductions

- Given 2^X KP for the full 32-round GOST.
- Obtain Y KP for 8 rounds of GOST.
- This valid with probability 2^{-Z}.
- For a proportion 2^{-T} of GOST keys.

Some 40 distinct reductions of this type with a large variety of X,Y, Z, T can be found in eprint/2011/626

Example

- Given 2^{32} KP for the full 32-round GOST.
- Obtain 4 KP for 8 rounds of GOST.
- This valid with probability 2^{-128}.

Is Algebraic Complexity Reduction Already Known?

There exists many known attacks which enter the framework of Algebraic Complexity Reduction:

- Slide attacks
- Fixed Point Attacks
- Cycling Attacks
- Involution Attacks
- Guessing [Conditional Algebraic Attacks]
- Etc..

What's New?

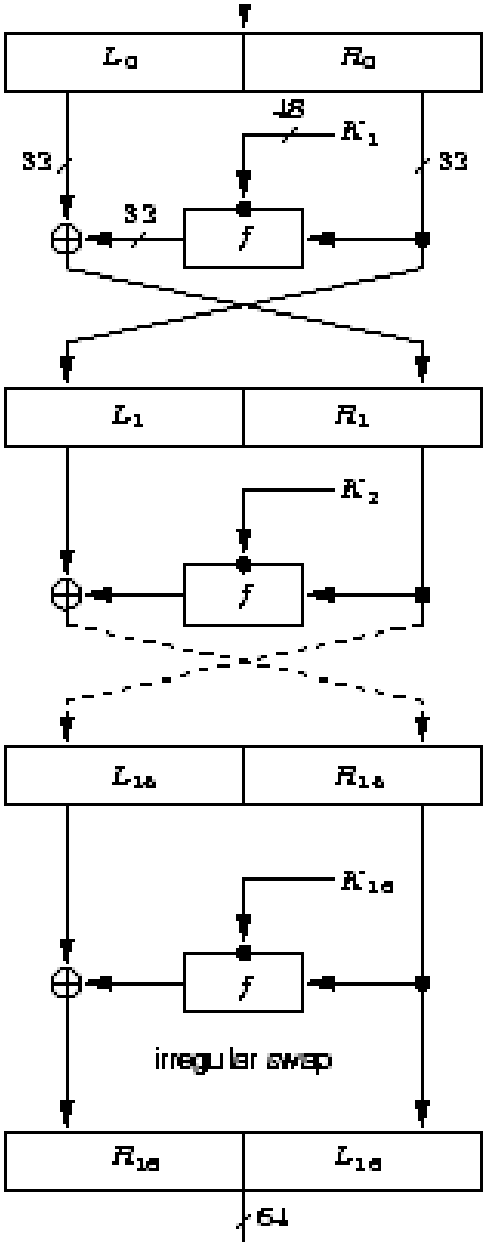
Slide / Fixed Point / Cycling / Guessing / Etc..

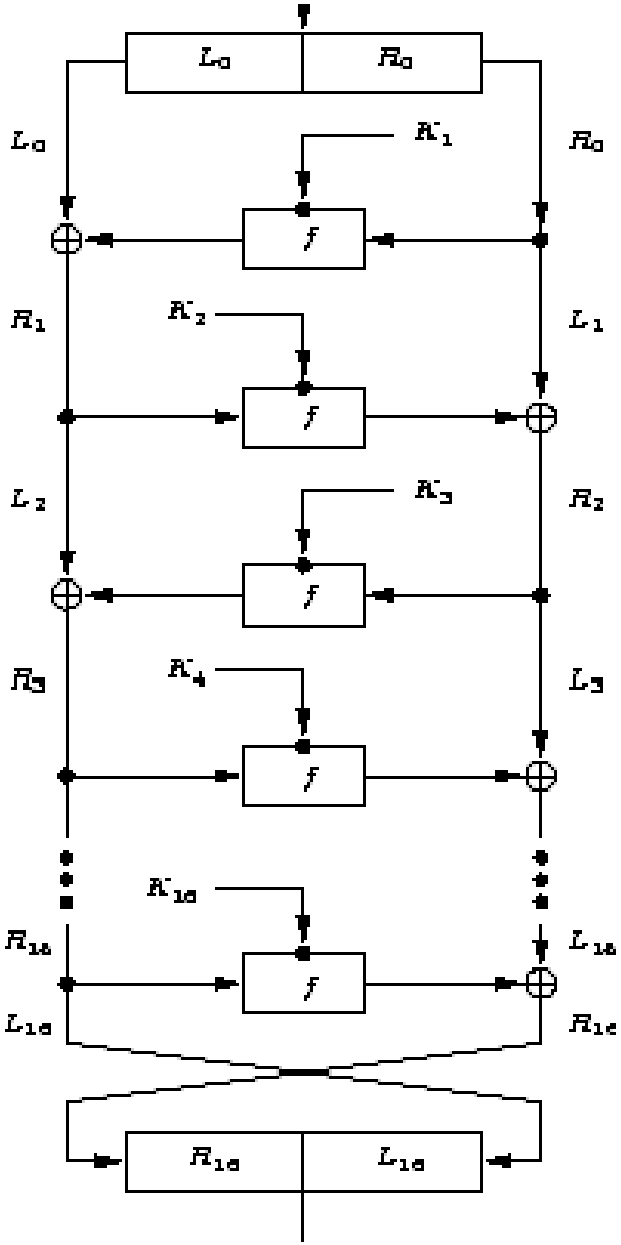
WHAT'S NEW?

- There are now many completely new attacks which are exactly none of the above [though similar or related].
- Many new attacks are possible and many of these attacks were <u>never studied</u> because they generate only a few known plaintexts, and only in the last 5 years it became possible to design an appropriate last step for these attacks which is a low-data complexity key recovery attack [e.g. algebraic, MITM].

Feistel Schemes

2x Same

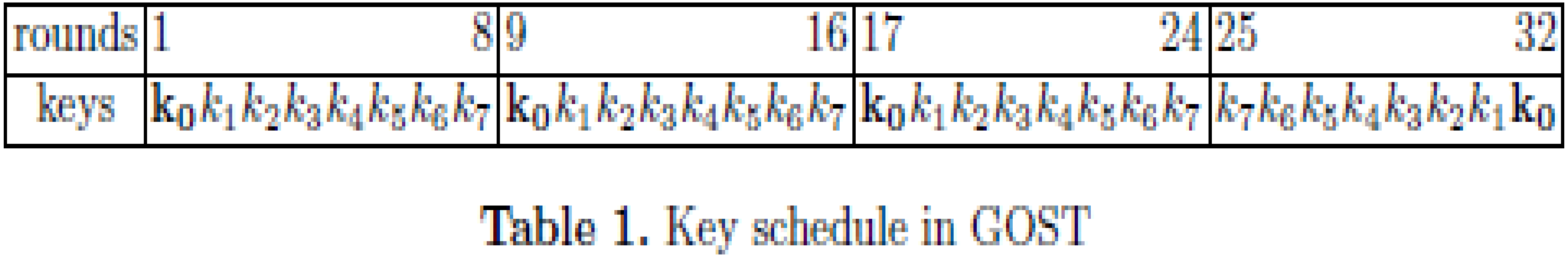

L_0
R_0
48
K_1
33
33
33
f
L_1
R_1
K_2
f
L_{16}
R_{16}
K_{16}
f
irregular swap
R_{16}
L_{16}
64


L_0
R_0
L_0
K_1
R_0
f
R_1
K_2
L_1
f
L_2
K_3
R_2
f
R_3
K_4
L_3
f
K_{16}
R_{16}
L_{16}
f
L_{16}
R_{16}
R_{16}
L_{16}

6.2. Structure of GOST

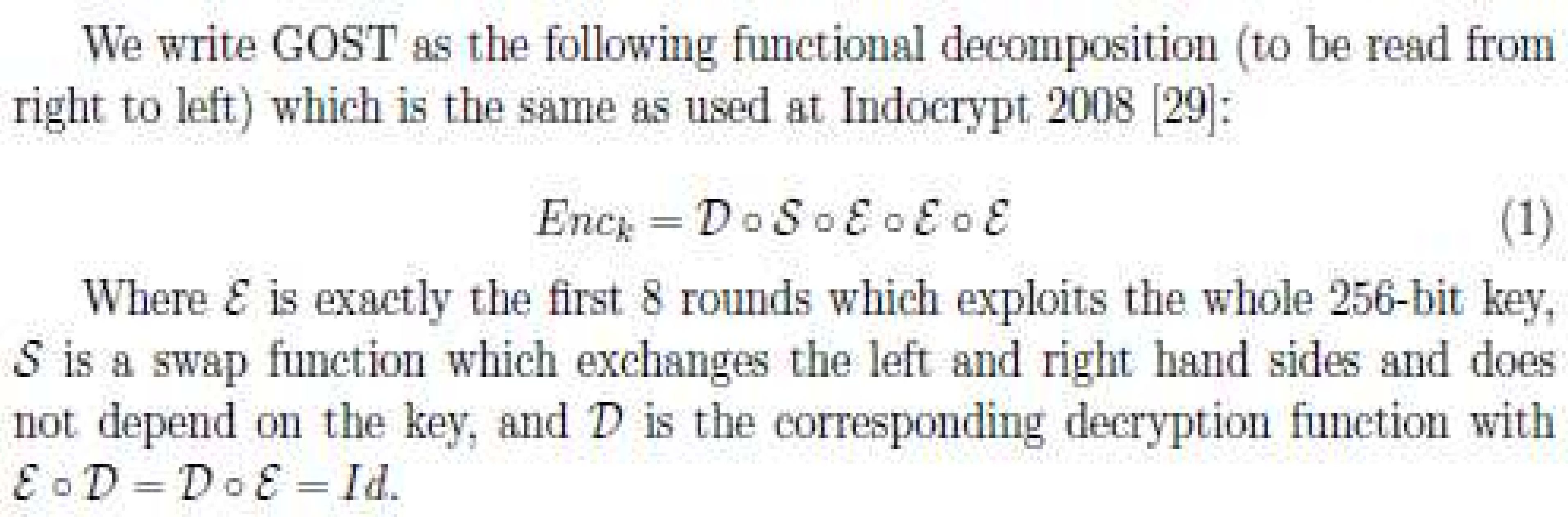
$$Enc_k = \underleftarrow{\mathcal{D} \circ \mathcal{S} \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E}}$$

Self-Similar Key Schedule
Periodic Repetition + Inversed Order



rounds	1 8	9 16	17 24	25 32
keys	$\mathrm{k}_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$	$\mathrm{k}_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$	$\mathrm{k}_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$	$k_7 k_6 k_5 k_4 k_3 k_2 k_1 \mathrm{k}_0$

Table 1. Key schedule in GOST

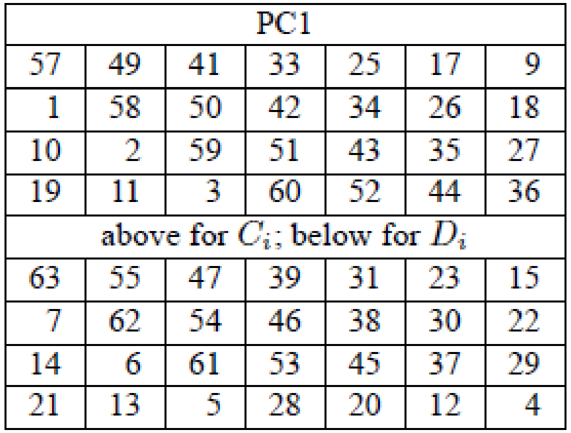


We write GOST as the following functional decomposition (to be read from right to left) which is the same as used at Indocrypt 2008 [29]:

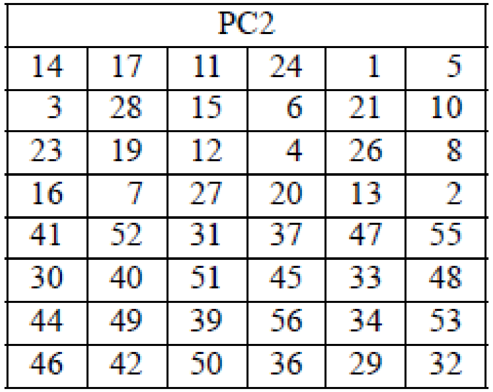
$$Enc_k = \mathcal{D} \circ \mathcal{S} \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E} \qquad (1)$$

Where $\mathcal{E}$ is exactly the first 8 rounds which exploits the whole 256-bit key, $\mathcal{S}$ is a swap function which exchanges the left and right hand sides and does not depend on the key, and $\mathcal{D}$ is the corresponding decryption function with $\mathcal{E} \circ \mathcal{D} = \mathcal{D} \circ \mathcal{E} = Id$.

*Compare: DES



PC1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
above for C_i; below for D_i						
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

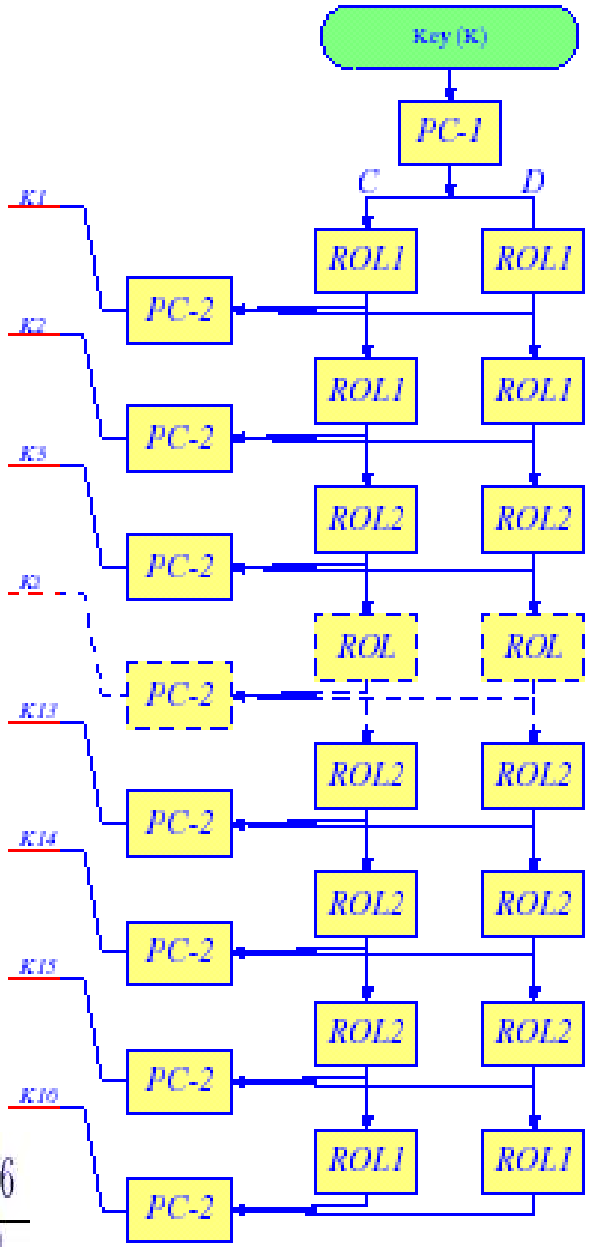


PC2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

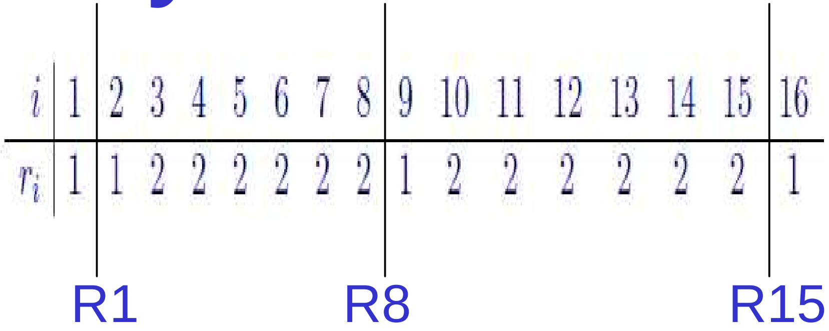
16*48 subsets of 56 bits.

```
1: $K \xrightarrow{\text{PC1}} (C, D)$
2: for $i = 1$ to 16 do
3:     $C \leftarrow \text{ROL}_{r_i}(C)$
4:     $D \leftarrow \text{ROL}_{r_i}(D)$
5:     $K_i \leftarrow \text{PC2}(C, D)$
6: end for
```

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
r_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1



Fixed Points: DES Key Schedule



i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
r_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

R1 R8 R15

- Can DES key be periodic?
- After step 1= key for R1
- After step 8=key for R8
- After step 15=key for R15
- We have a pattern G
  of length 7 which repeats twice.
- Unhappily G = + 13 mod 28 (and not 14)
- Does NOT have many fixed points.

Last 16 Rounds of GOST

$$Enc_k = \boxed{\mathcal{D} \circ \mathcal{S} \circ \mathcal{E}} \circ \mathcal{E} \circ \mathcal{E}$$

$$\longleftarrow$$

“Theorem Which Won World War 2”,

[I. J. Good and Cipher A. Deavours, afterword to: Marian Rejewski, "How Polish Mathematicians Deciphered the Enigma", Annals of the History of Computing, 3 (3), July 1981, 229-232]

P and

Q^{-1} o P o Q

have the same cycle structure

Last 16 Rounds of GOST

$$Enc_k = \boxed{\mathcal{D}\circ\mathcal{S}\circ\mathcal{E}}\circ\mathcal{E}\circ\mathcal{E}$$

$$\xleftarrow{\hspace{6cm}}$$

“Theorem Which Won World War 2”,

$\Rightarrow$ Has exactly 2^{32} fixed points (order 1) and 2^{64}-2^{32} points of order 2.

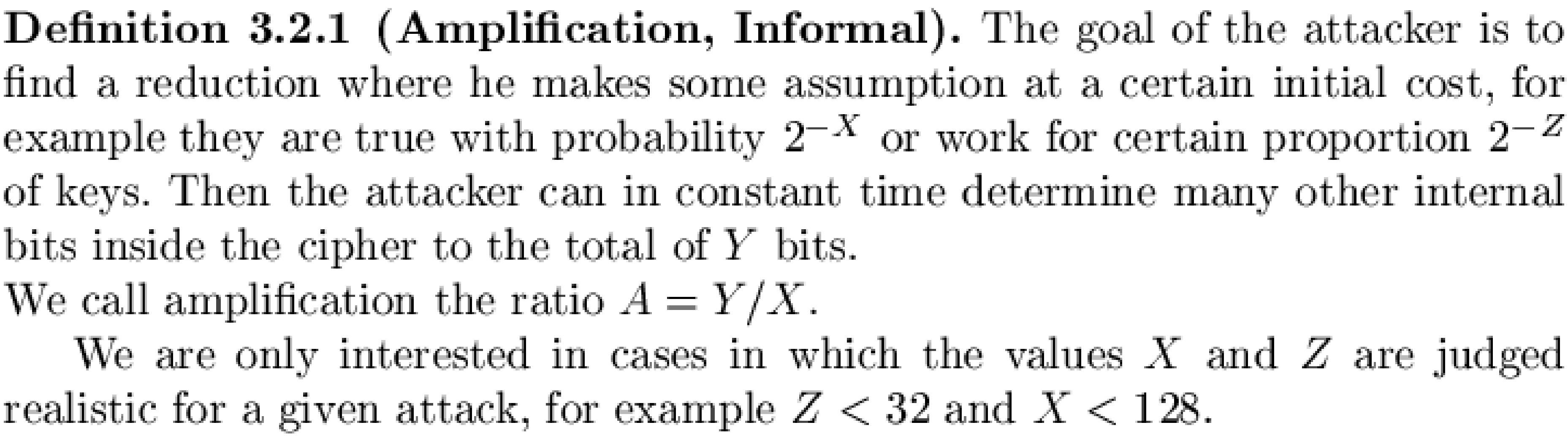
$\Rightarrow$ A lot of fixed points (very few for DES).

6.3. Complexity Reduction ☂ in Guess-Then-Determine attacks

Reason: Self-Similarity

6.3.1. Guess-Then-Determine: Amplification

Amplification



Definition 3.2.1 (Amplification, Informal). The goal of the attacker is to find a reduction where he makes some assumption at a certain initial cost, for example they are true with probability 2^{-X} or work for certain proportion 2^{-Z} of keys. Then the attacker can in constant time determine many other internal bits inside the cipher to the total of Y bits.
We call amplification the ratio $A = Y/X$.

We are only interested in cases in which the values X and Z are judged realistic for a given attack, for example $Z < 32$ and $X < 128$.

Killer examples:

- Slide attacks – unlimited.
- Weak Key Family 3 in GOST –
  VERY large => attack on GOST with 2^{159} per key

6.4. Complexity Reduction: First Example:

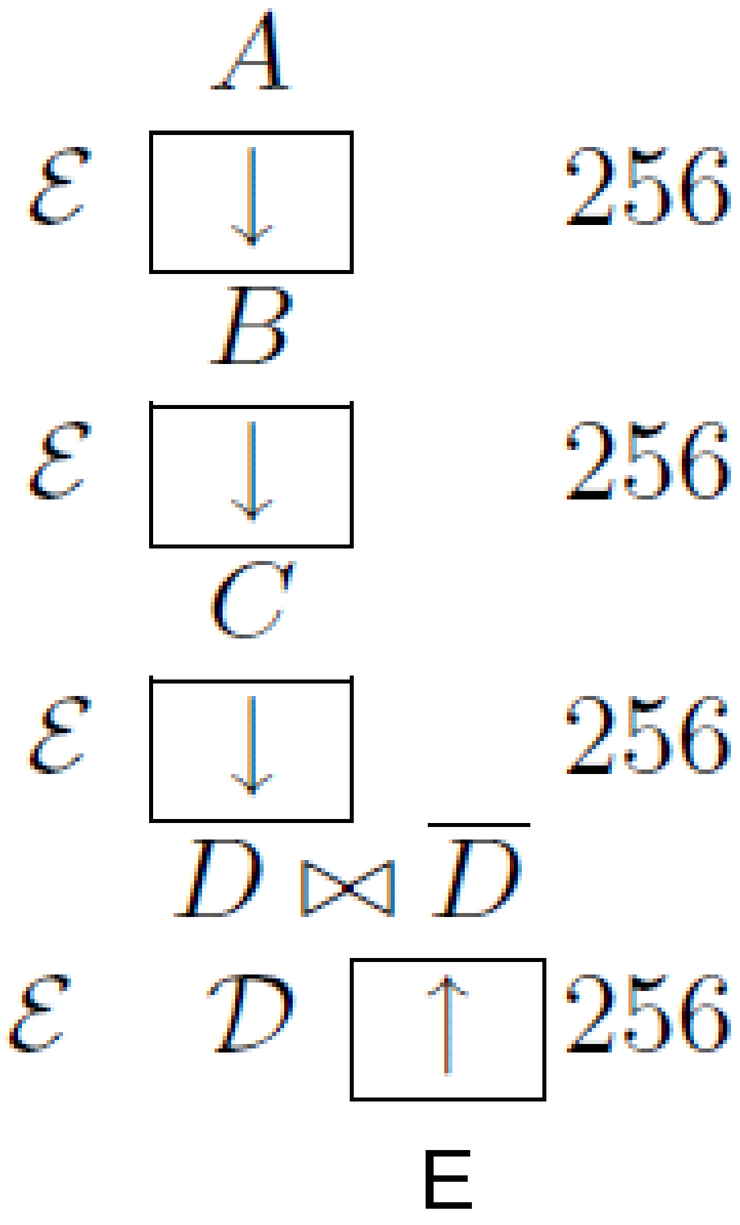
Relaxing the Requirements of A Sliding Attack

Black Box Reduction:
Pseudo-Sliding Attack

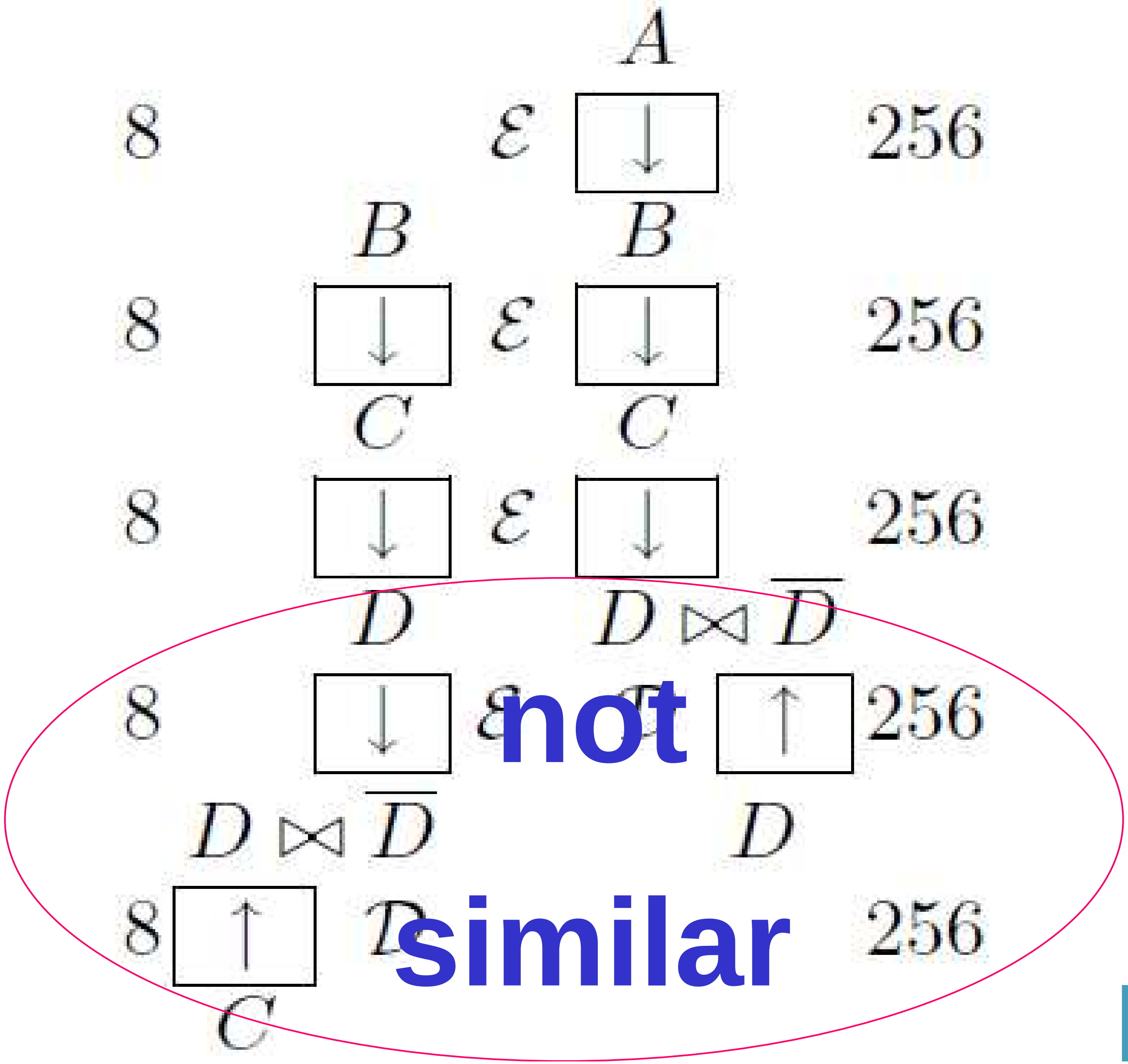
[Cryptologia Jan 2012]

One Encryption

$$Enc_k = \mathcal{D} \circ \mathcal{S} \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E}$$


A
$\mathcal{E}$ $\downarrow$ 256
B
$\mathcal{E}$ $\downarrow$ 256
C
$\mathcal{E}$ $\downarrow$ 256
$D \bowtie \overline{D}$
$\mathcal{E}$ $\mathcal{D}$ $\uparrow$ 256
E

Two Encryptions with A Slide

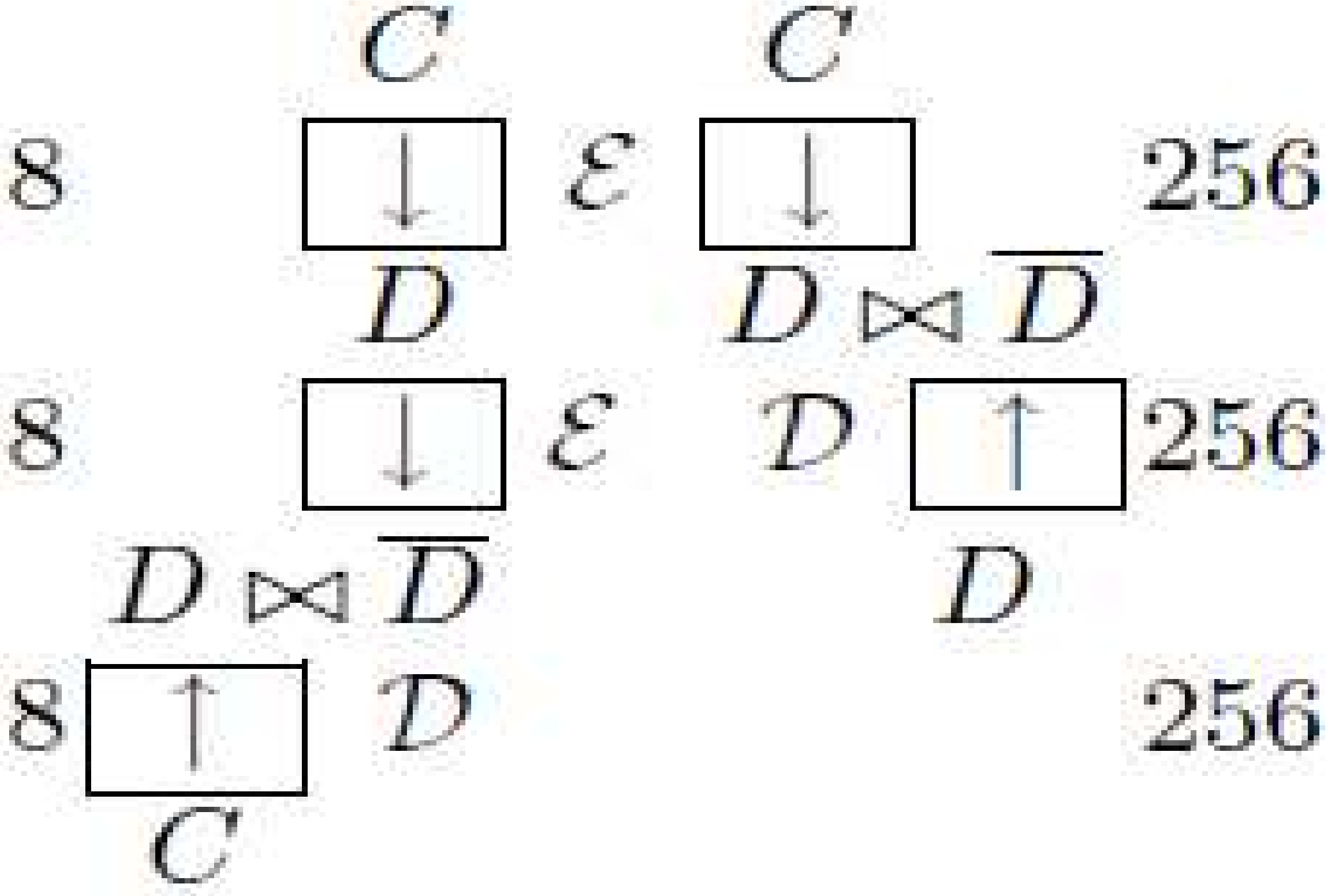


not similar

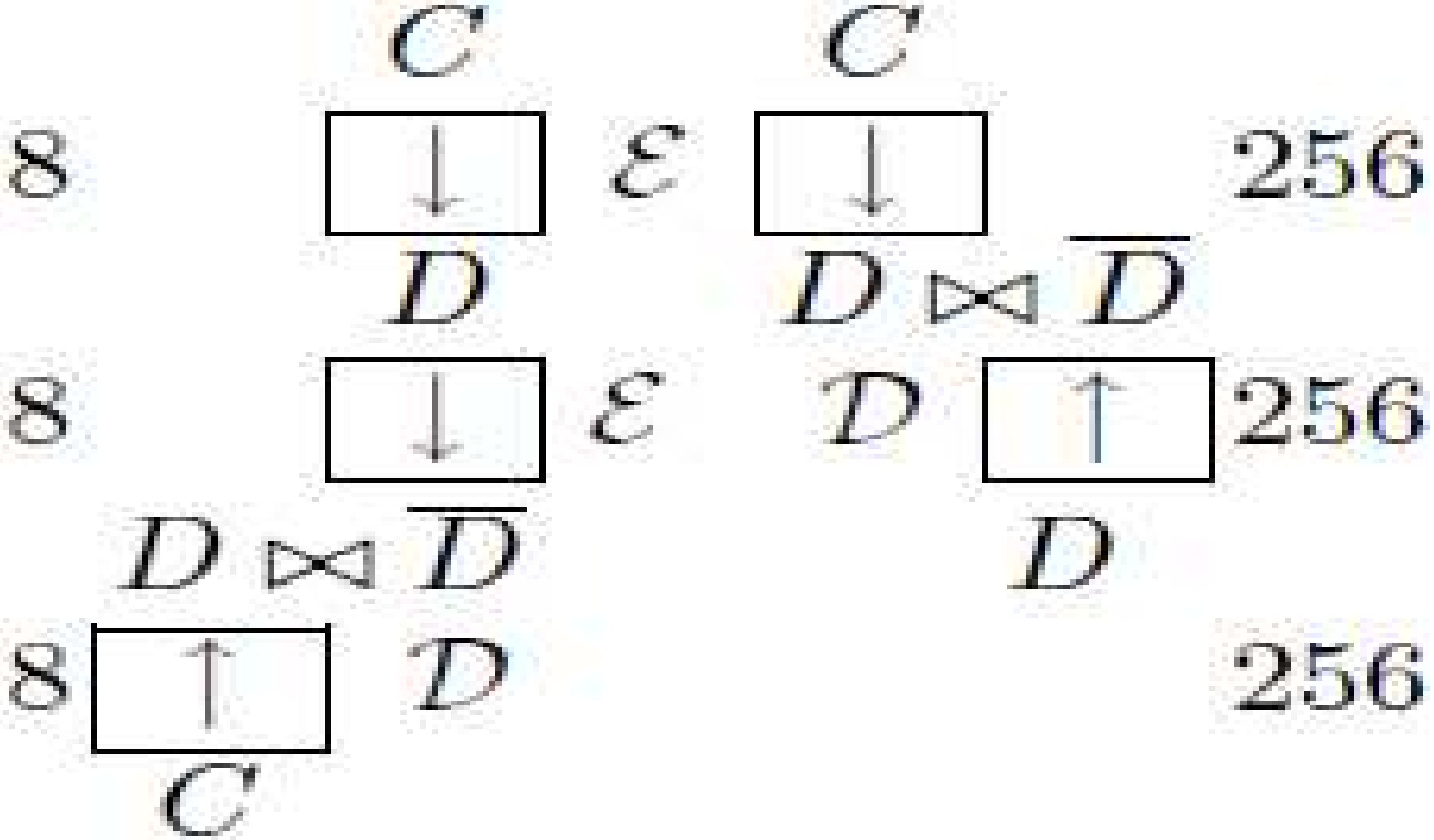
Assumptions

We proceed as follows. We consider plaintexts with a very peculiar property:

Assumption 1 (Assumption W). Let A be such that $\mathcal{E}(D) = \overline{D}$ where D is defined as $D = \mathcal{E}^3(A)$.



GOST, Self-Sir



Fact 2 (Property W). Given 2^{64} KP there is on average one value A which satisfies the Assumption. For 63% of all GOST keys at least one such A exists.

Remark: For the remaining 37 % of keys this attack fails. However many other attacks still work, see [12].

Reduction

New Attack on GOST

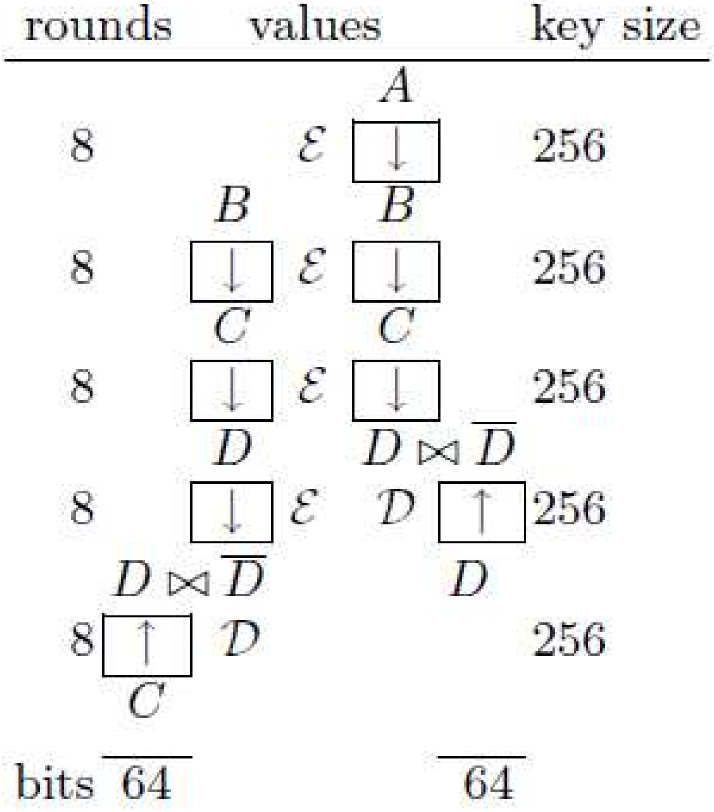
Fact 3 (Consequences of Property W). If A satisfies the Assumption W above and defining $B = \mathcal{E}(A)$ and $C = \mathcal{E}(B)$ we have:

1. $Enc_k(A) = D$. This is illustrated on the right hand side of Fig. 1.
2. $Enc_k(B) = C$ This can be seen on the left hand side of Fig. 1.

2^{64} KP

guess A,B

correct $P=2^{-128}$



$P=2^{-128}$

=>

4 pairs

for 8 rounds

Fig. 1. A black-box "Algebraic Complexity Reduction" from 32 to 8 rounds of GOST

Final Key Recovery 8R

4 Pairs, 8 rounds.

The key is found within

2^{110} GOST computations.

Overall Attack

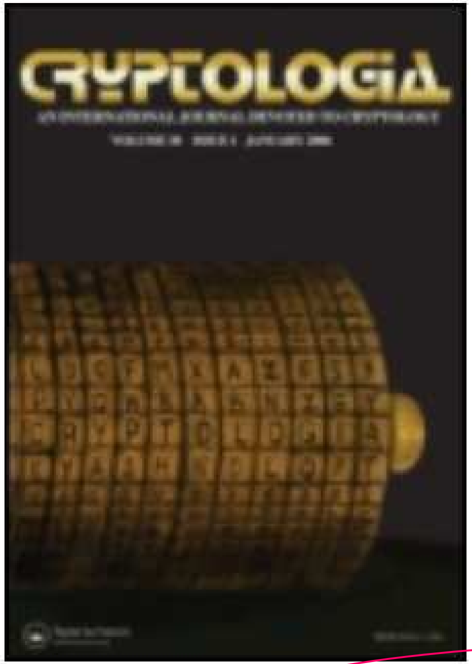
$2^{128+110}$ GOST computations.

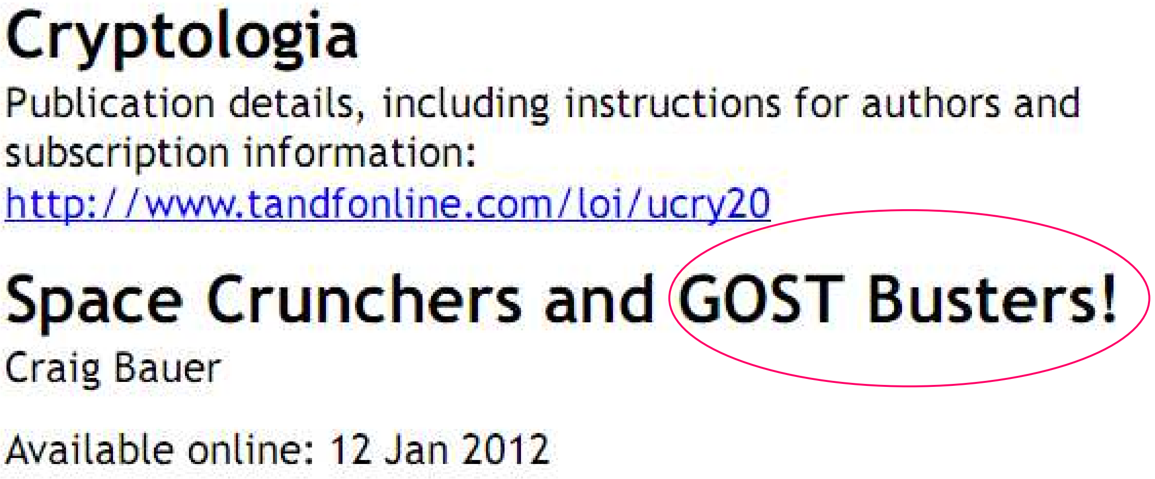
2^{17} times faster than brute force.

Not the best attack yet.

Cryptologia [Jan 2012]

Editorial:





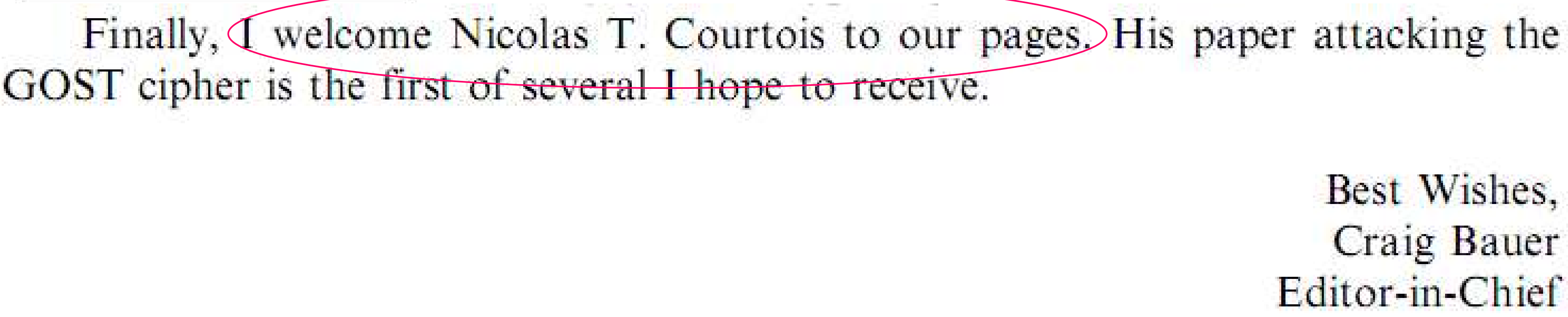
Cryptologia

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucry20

Space Crunchers and GOST Busters!

Craig Bauer

Available online: 12 Jan 2012



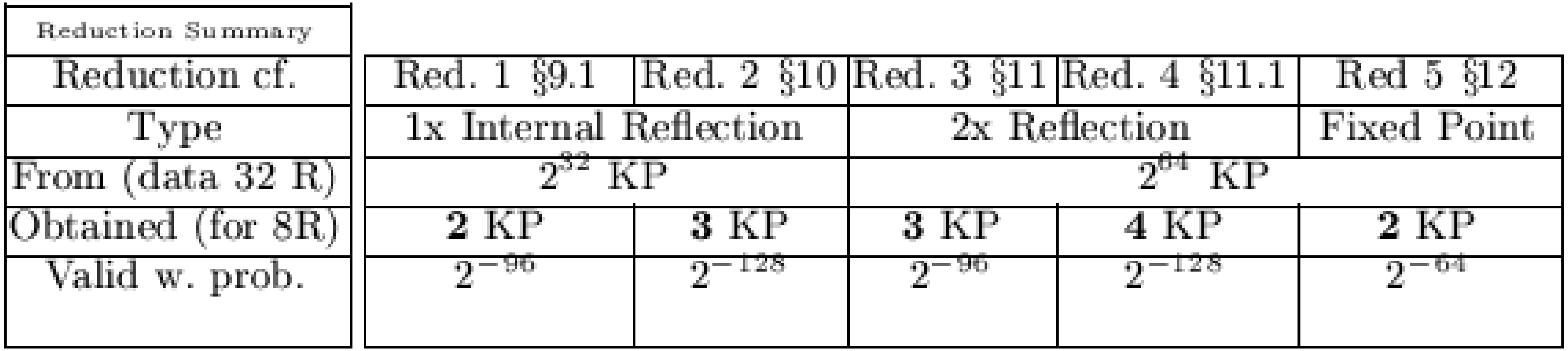
Finally, I welcome Nicolas T. Courtois to our pages. His paper attacking the GOST cipher is the first of several I hope to receive.

Best Wishes,
Craig Bauer
Editor-in-Chief

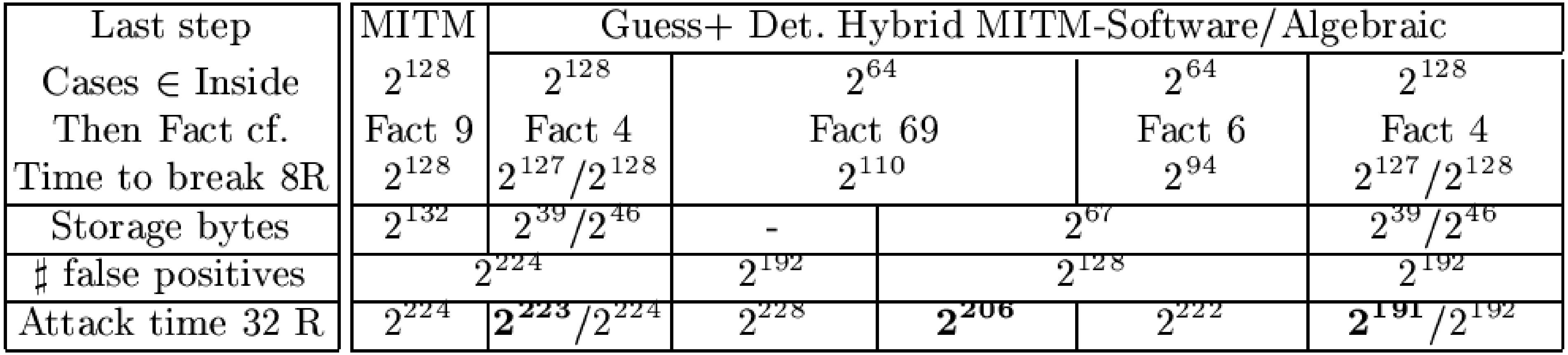
6.5. More Single Key Attacks…

Many more single-key attacks on full 32-round GOST…

cf. eprint.iacr.org/2011/626/



Reduction Summary					
Reduction cf.	Red. 1 §9.1	Red. 2 §10	Red. 3 §11	Red. 4 §11.1	Red 5 §12
Type	1x Internal Reflection		2x Reflection		Fixed Point
From (data 32 R)	2^{32} KP		2^{64} KP		
Obtained (for 8R)	**2 KP**	**3 KP**	**3 KP**	**4 KP**	**2 KP**
Valid w. prob.	2^{-96}	2^{-128}	2^{-96}	2^{-128}	2^{-64}



Last step	MITM	Guess+ Det. Hybrid MITM-Software/Algebraic				
Cases ∈ Inside	2^{128}	2^{128}	2^{64}		2^{64}	2^{128}
Then Fact cf.	Fact 9	Fact 4	Fact 69		Fact 6	Fact 4
Time to break 8R	2^{128}	$2^{127}/2^{128}$	2^{110}		2^{94}	$2^{127}/2^{128}$
Storage bytes	2^{132}	$2^{39}/2^{46}$	-	2^{67}		$2^{39}/2^{46}$
♯ false positives	2^{224}		2^{192}	2^{128}		2^{192}
Attack time 32 R	2^{224}	$\mathbf{2^{223}}/2^{224}$	2^{228}	$\mathbf{2^{206}}$	2^{222}	$\mathbf{2^{191}}/2^{192}$

Science ≠ Politics

Main paper was submitted to Asiacrypt 2011.

One referee wrote: “I think that the audiences of Asiacrypt will not feel it is interesting.”

=>however about half of papers accepted at this Asiacrypt are about things about which nobody ever heard, not even professional cryptologists (say JH42, Armadillo,theory, incremental research, things which would interest very few people)…, not to say it would interest anybody in the industry or government circles…

=>HOW many times it ever happened at Asiacrypt that a military-grade cipher, and an official government standard of a major country, used by large banks, implemented in SSL, was broken, while being in the process of being standardized by ISO to become a global industrial standard? Not many times.

⇒ impacting potentially all of: national critical infrastructures, key financial systems and even ordinary computer software
  - ⇒ It could be worth tens of billions of dollars to fix problems due to GOST..
  - ⇒ For now nothing bad happened, just some bad press.
    - ⇒ BUT: Is GOST really broken?

Science ≠ Politics

But is GOST really so bad?

When it was submitted to ISO, and only then, suddenly some cryptanalysts tried to break it… And succeeded.

And there is now more than 50 attacks… Academic attacks.

We do in “the West” ☺ put VERY HIGH super-paranoid requirements on security of ciphers…

- ⇒ It is debatable whether the Russian designers of GOST ever thought that it should not have attacks faster than 2^{256}…
- ⇒ Remember that GOST can have a secondary key: secret S-boxes.

Even today, in spite of all our 20+ attacks, GOST is better than any comparable cipher:

Look at the (best attack) / (implementation cost) ratio

cf. Poschmann et al CHES 2010

- – Key schedule could be easily fixed to avoid academic shortcut attacks…
- – GOST-P is even better (better S-box <= PRESENT: new ISO standard).

6.6. Black Box Reduction: Reflection Attack

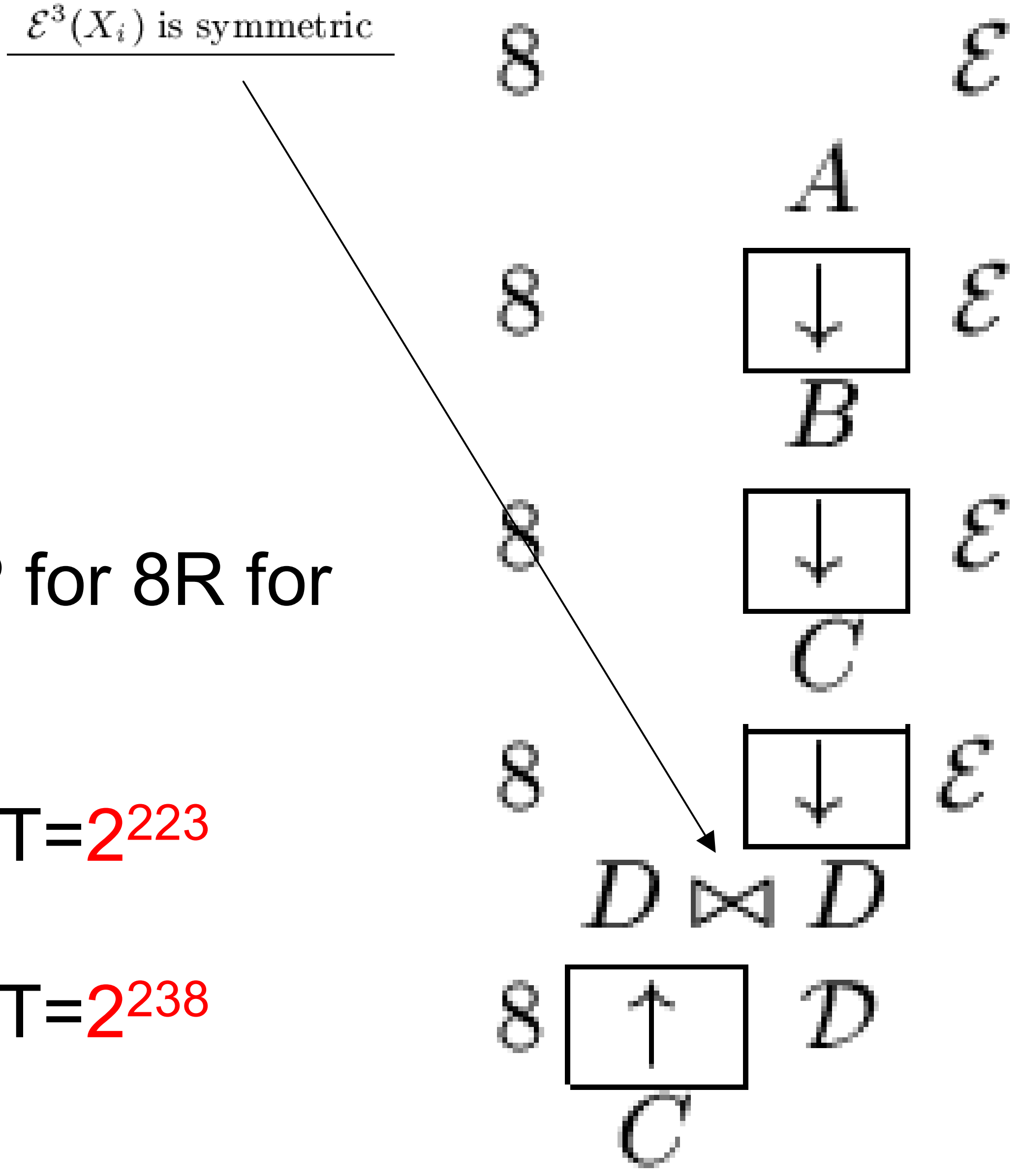
Reflection – Happens 2^{32} Times - KPA

- guess A det C
  info=64 cost=2^{-32}
- guess B
  info=64+64 cost=2^{-64}
- [guess D
  info=64 cost=2^{-32}]

Summary: we get 2/3 KP for 8R for the price of $2^{-96}/2^{-128}$.

break 8R 2KP 2^{127}
=> break 32R D=2^{32} T=2^{223}

break 8R 3KP 2^{110}
=> break 32R D=2^{32} T=2^{238}



6.7. Double Reflection Attack

2x Reflection, Happens About Once:

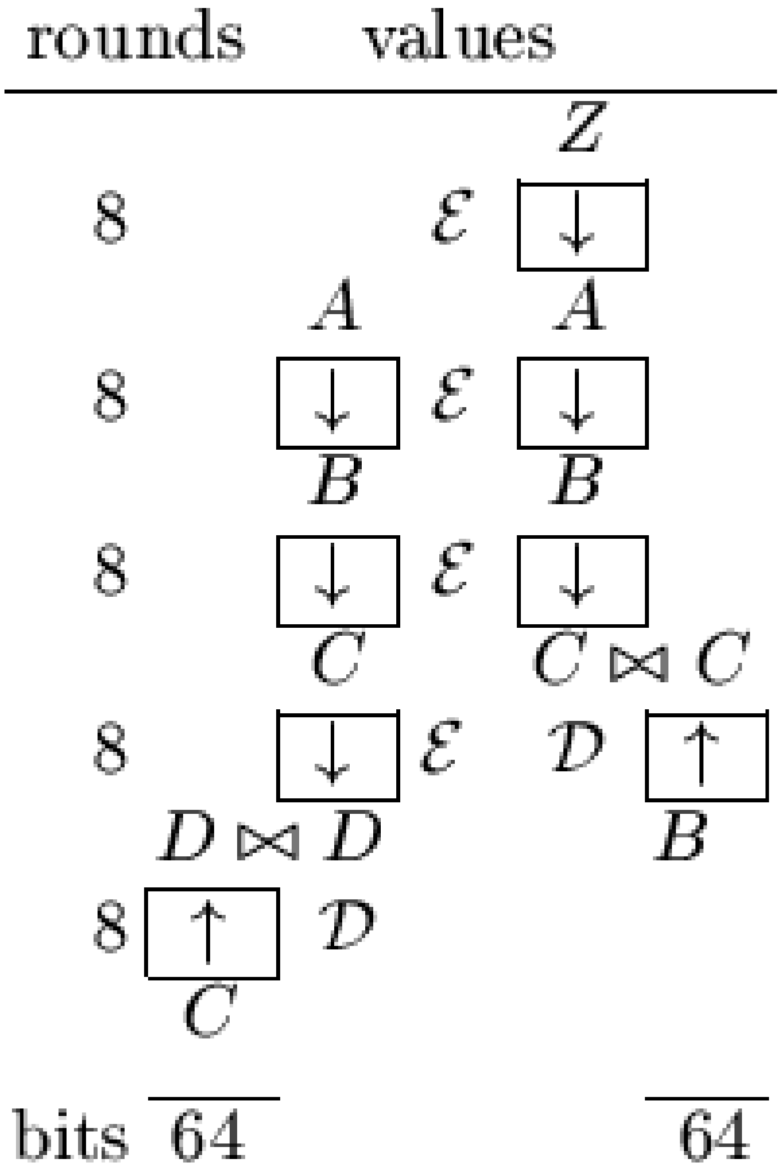
$\mathcal{E}^2(X_i)$ symmetric
$\mathcal{E}^3(X_i)$ symmetric

- guess C det A
  info=64 cost=2^{-32}
- guess B det Z
  info=64+64+64 cost=2^{-64}
- [guess D
  info=64 cost=2^{-32}]

Summary: we get 3/4 KP for
8R for the price of $2^{-96}/2^{-128}$

break 8R 3KP 2^{110}
=> break 32R D=2^{64} T=2^{206}

break 8R 4KP 2^{94}
=> break 32R D=2^{64} T=2^{222}



Other Attacks?

Best single key attack:

$D=2^{64}$ $T=2^{179}$

Nicolas Courtois: An Improved Differential Attack on Full GOST, March 2012, eprint.iacr.org/2012/138.

However ciphers are NEVER used with single keys in the real life… On the contrary.

NEW!

7. Multiple Random Key Scenario

“stronger, more versatile
and MORE practical
than any known
single key attack”

???

7.1. One Triple Reflection Attack

3x Reflection, Weak Keys 2^{-64}

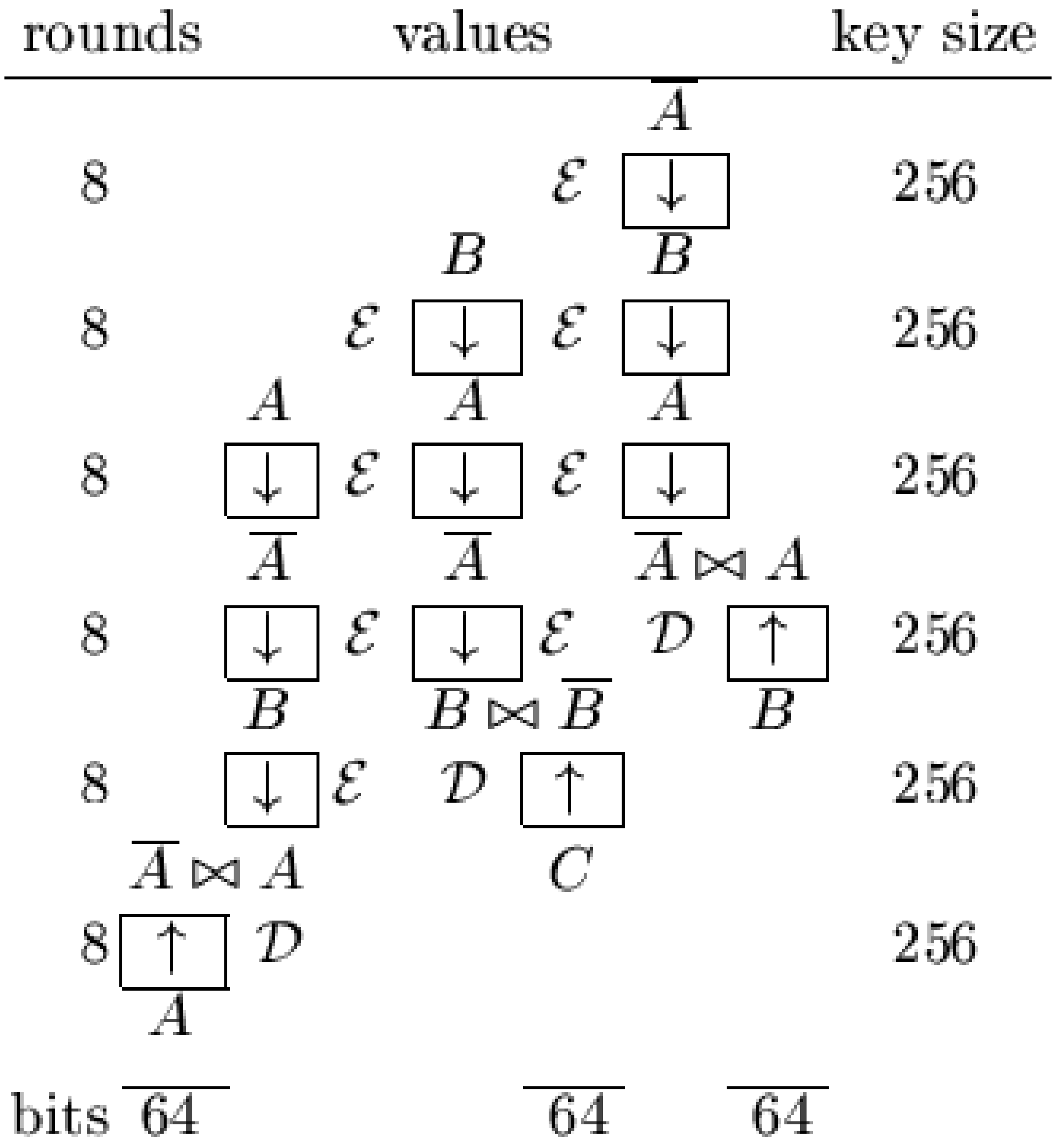
$$\mathcal{E}^2(\overline{A}) = A$$

$$\mathcal{E}(A) = \overline{A}$$

No guessing =>

Very high amplification.

All data obtained
nearly “for free”.


rounds
values
key size
$\overline{A}$
8 $\mathcal{E}$ ↓ 256
B B
8 $\mathcal{E}$ ↓ $\mathcal{E}$ ↓ 256
A A A
8 ↓ $\mathcal{E}$ ↓ $\mathcal{E}$ ↓ 256
$\overline{A}$ $\overline{A}$ $\overline{A} \bowtie A$
8 ↓ $\mathcal{E}$ ↓ $\mathcal{E}$ $\mathcal{D}$ ↑ 256
B $B \bowtie \overline{B}$ B
8 ↓ $\mathcal{E}$ $\mathcal{D}$ ↑ 256
$\overline{A} \bowtie A$ C
8 ↑ $\mathcal{D}$ 256
A
bits $\overline{64}$ $\overline{64}$ $\overline{64}$

7.2. Combined Attacks: DC + Algebraic Complexity Reduction

two totally unrelated families of attacks…
…until December 2012

NEW!

New Combined Attacks

New attacks from November 2012 combine ALL of truncated differentials, fixed points, advanced MITM, software/SAT solvers and reflection in ONE single attack. Example:

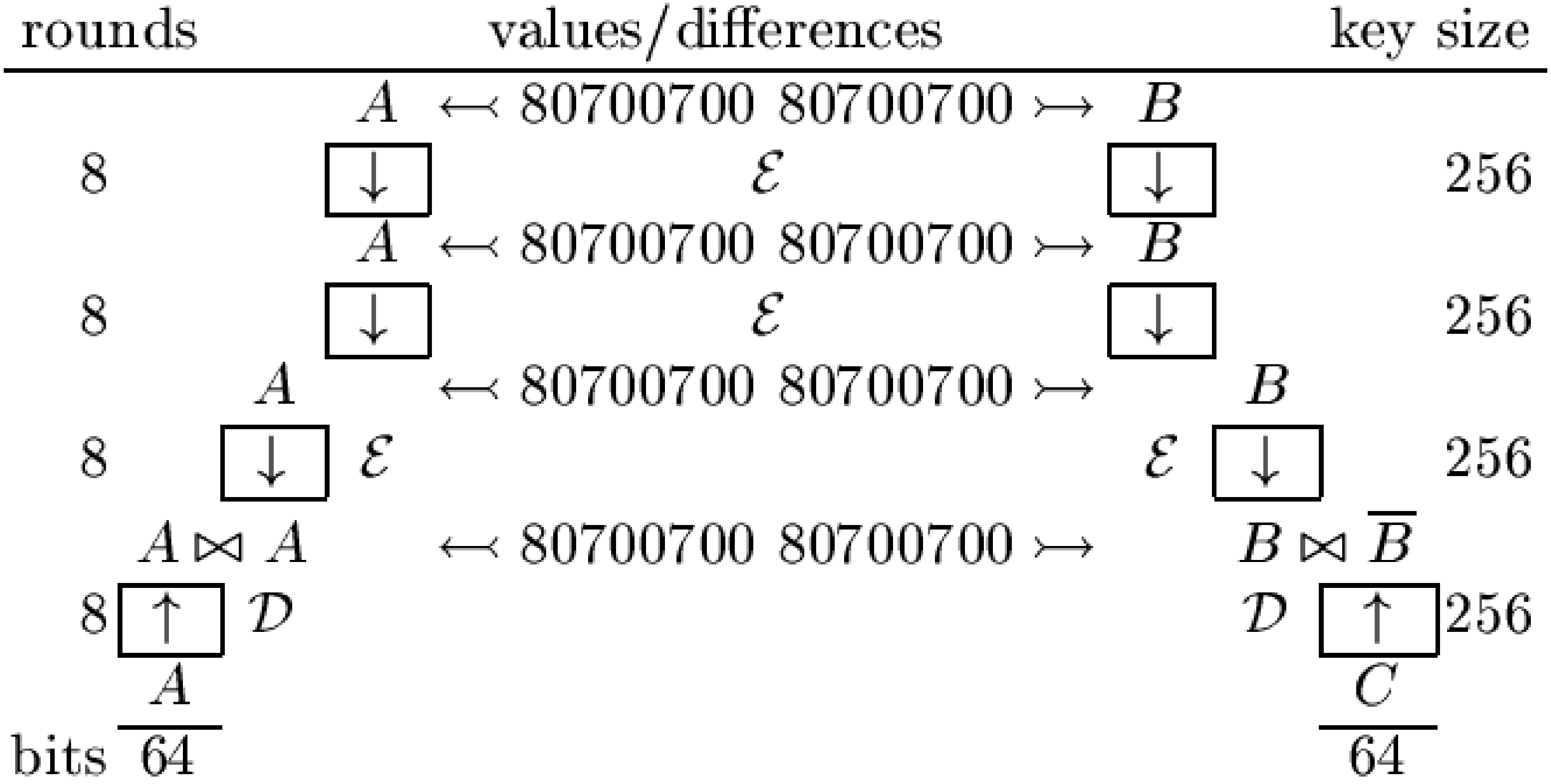
Family 5.3. Fact 47 Section 19.5.

Given 2^{52} devices with random keys on 256 bits and 2^{32} ACP (Adaptively Chosen Plaintexts), we can recover one GOST key in time of 2^{139}.

Total data = 2^{84}. Mostly used to reject keys which do not satisfy our conditions.

Combined DC+Algebraic Complexity Reduction

3 KP for 8R obtained. Time(8R)= 2^{110}.


rounds
values/differences
key size
$A \leftarrow$ 80700700 80700700 $\rightarrow B$
8
$\mathcal{E}$
256
$A \leftarrow$ 80700700 80700700 $\rightarrow B$
8
$\mathcal{E}$
256
$A \leftarrow$ 80700700 80700700 $\rightarrow B$
8
$\mathcal{E}$
$\mathcal{E}$
256
$A \bowtie A \leftarrow$ 80700700 80700700 $\rightarrow B \bowtie \overline{B}$
8
$\mathcal{D}$
$\mathcal{D}$
256
A
C
bits 64
64

8. Multiple-Point Events and Bicliqes

NEW!

Attacks with Multiple Fixed Points and Bicliques

New attacks with multiple related encryptions
+ additional well-chosen properties,
as usual.

A form of advanced
higher-order differential attack.

Greatly decreases the cost of making
assumptions such as A=B' etc.

Single Key Approximate Multiple Fixed Points

=> all
8 points
share
the same
50 bits!

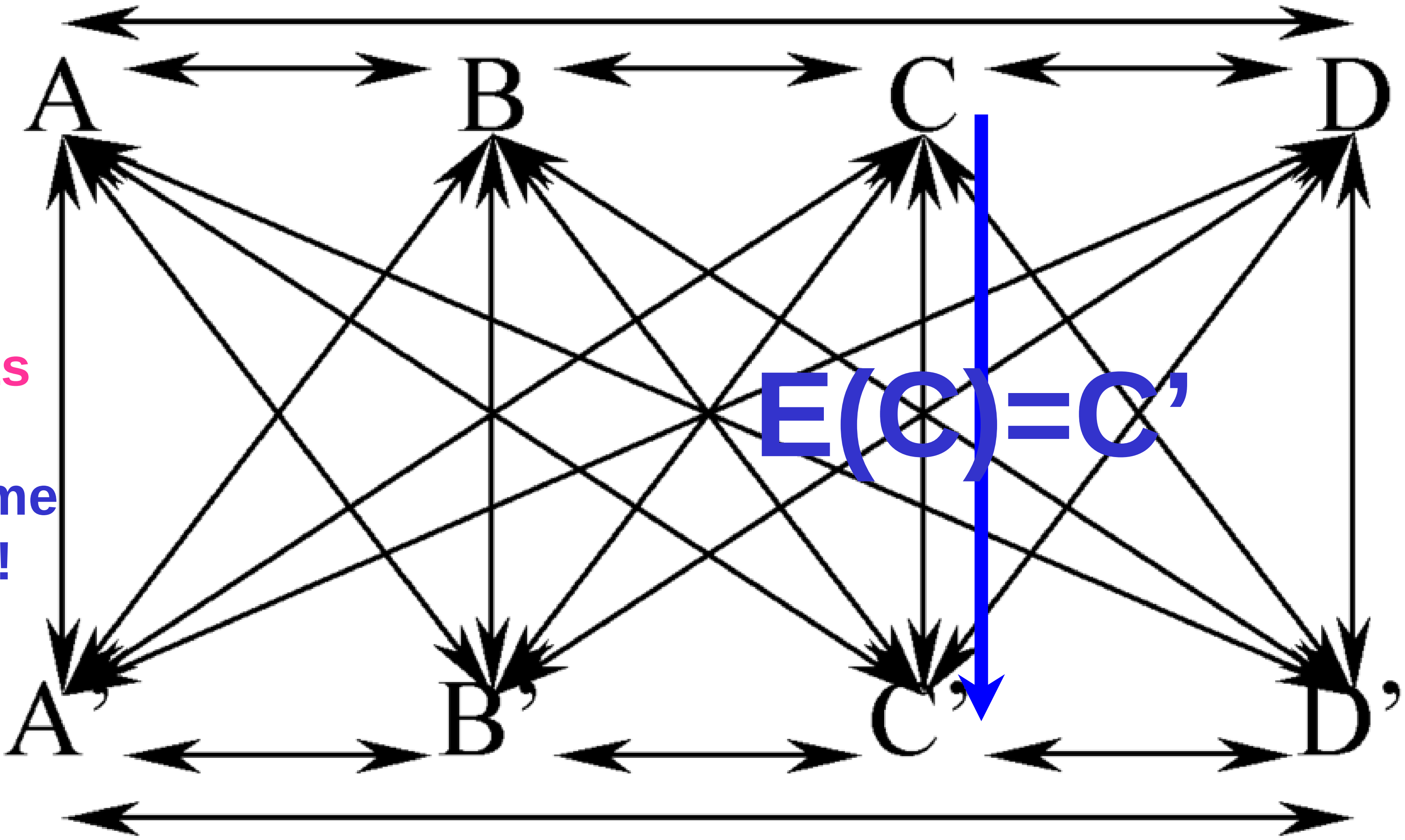


Fig. 18. An approximate fixed point biclique with $k = 4$

Attacks with Multiple Fixed Points and Bicliques

Example:

Family 8.4. Fact 73 Section 22.6.

Given 2^{79} devices with random keys on 256 bits and 2^{32} CP per key we can recover one GOST key in time of 2^{101}.

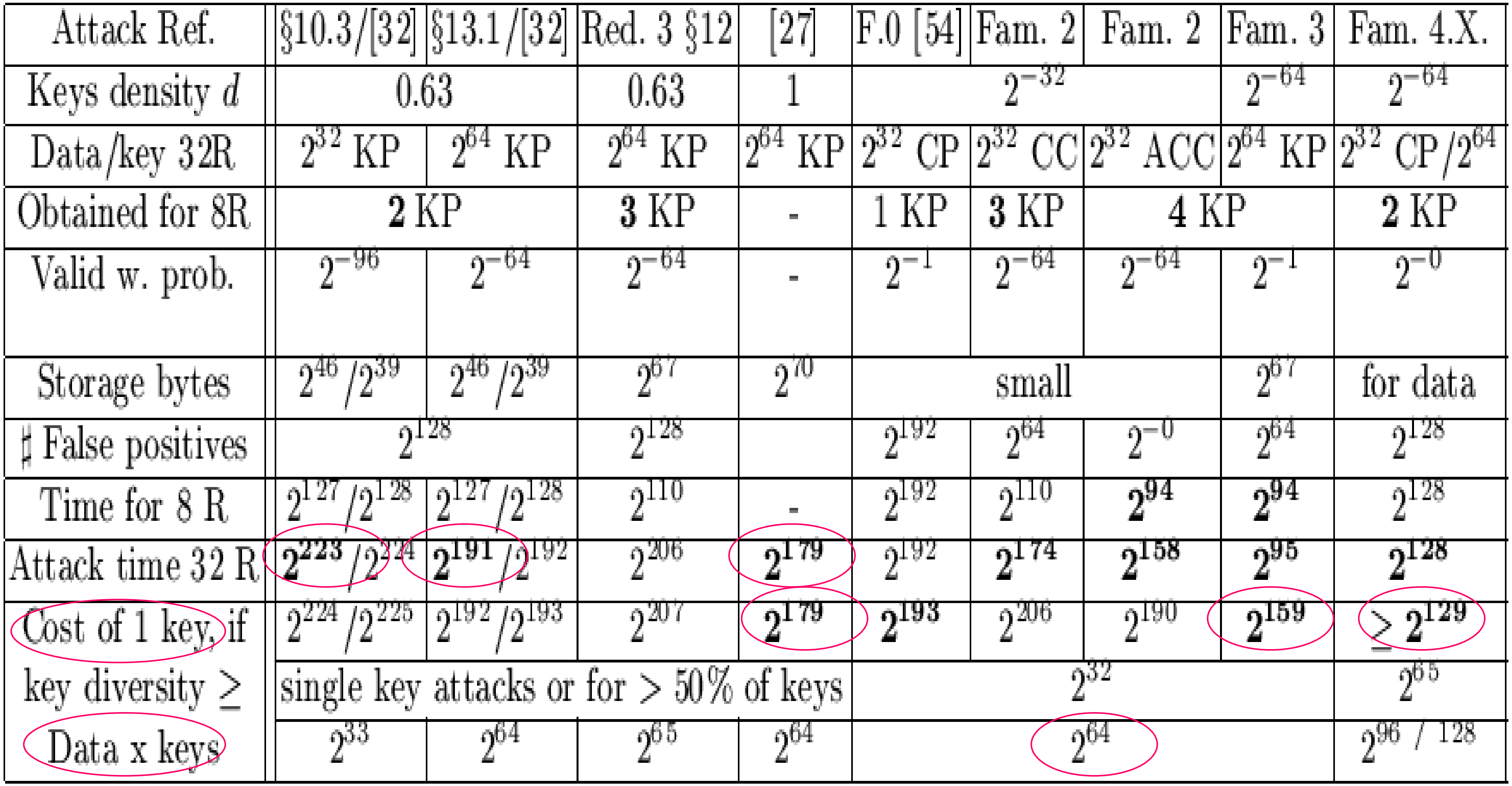
=> Nearly feasible (for a large intelligence agency).

=> Further improvements expected…

8.2. Summary

The Multiple Key Scenario (1)

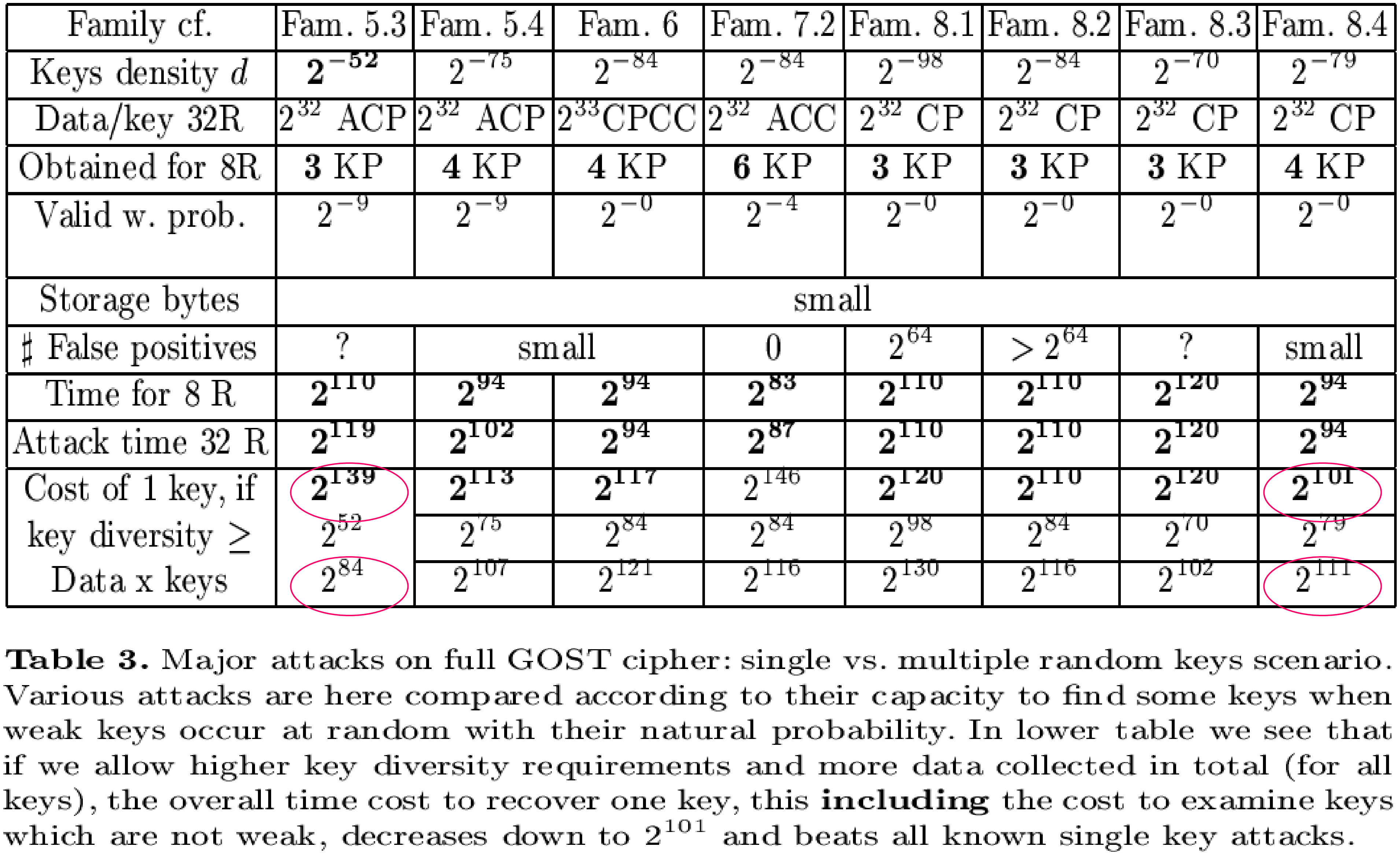
cf. eprint.iacr.org/2011/626/



Attack Ref.	§10.3/[32]	§13.1/[32]	Red. 3 §12	[27]	F.0 [54]	Fam. 2	Fam. 2	Fam. 3	Fam. 4.X.
Keys density d	0.63		0.63	1	2^{-32}			2^{-64}	2^{-64}
Data/key 32R	2^{32} KP	2^{64} KP	2^{64} KP	2^{64} KP	2^{32} CP	2^{32} CC	2^{32} ACC	2^{64} KP	2^{32} CP/2^{64}
Obtained for 8R	**2** KP		**3** KP	-	1 KP	**3** KP	**4** KP		**2** KP
Valid w. prob.	2^{-96}	2^{-64}	2^{-64}	-	2^{-1}	2^{-64}	2^{-64}	2^{-1}	2^{-0}
Storage bytes	$2^{46}/2^{39}$	$2^{46}/2^{39}$	2^{67}	2^{70}	small			2^{67}	for data
♯ False positives	2^{128}		2^{128}		2^{192}	2^{64}	2^{-0}	2^{64}	2^{128}
Time for 8 R	$2^{127}/2^{128}$	$2^{127}/2^{128}$	2^{110}	-	2^{192}	2^{110}	$\mathbf{2^{94}}$	$\mathbf{2^{94}}$	2^{128}
Attack time 32 R	$\mathbf{2^{223}}/2^{224}$	$\mathbf{2^{191}}/2^{192}$	2^{206}	$\mathbf{2^{179}}$	2^{192}	$\mathbf{2^{174}}$	$\mathbf{2^{158}}$	$\mathbf{2^{95}}$	$\mathbf{2^{128}}$
Cost of 1 key, if	$2^{224}/2^{225}$	$2^{192}/2^{193}$	2^{207}	$\mathbf{2^{179}}$	$\mathbf{2^{193}}$	2^{206}	2^{190}	$\mathbf{2^{159}}$	$\geq \mathbf{2^{129}}$
key diversity $\geq$	single key attacks or for > 50% of keys				2^{32}				2^{65}
Data x keys	2^{33}	2^{64}	2^{65}	2^{64}	2^{64}				2^{96} / 128

The Multiple Key Scenario (2)

cf. eprint.iacr.org/2011/626/



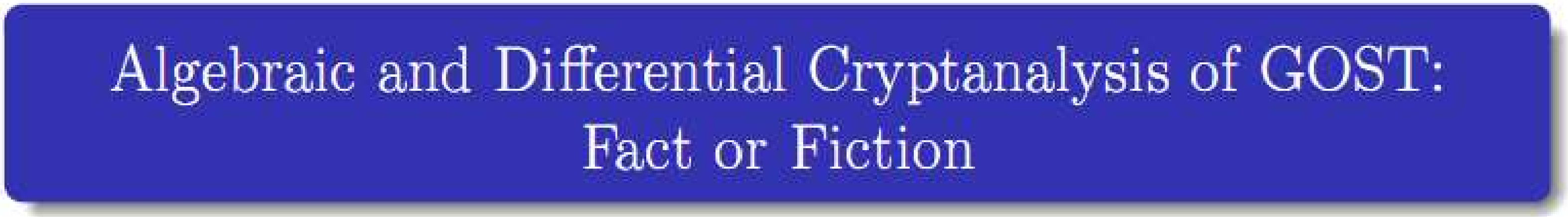
Family cf.	Fam. 5.3	Fam. 5.4	Fam. 6	Fam. 7.2	Fam. 8.1	Fam. 8.2	Fam. 8.3	Fam. 8.4
Keys density d	$\mathbf{2^{-52}}$	2^{-75}	2^{-84}	2^{-84}	2^{-98}	2^{-84}	2^{-70}	2^{-79}
Data/key 32R	2^{32} ACP	2^{32} ACP	2^{33}CPCC	2^{32} ACC	2^{32} CP	2^{32} CP	2^{32} CP	2^{32} CP
Obtained for 8R	**3** KP	**4** KP	**4** KP	**6** KP	**3** KP	**3** KP	**3** KP	**4** KP
Valid w. prob.	2^{-9}	2^{-9}	2^{-0}	2^{-4}	2^{-0}	2^{-0}	2^{-0}	2^{-0}
Storage bytes	small							
♯ False positives	?	small		0	2^{64}	$> 2^{64}$	?	small
Time for 8 R	$\mathbf{2^{110}}$	$\mathbf{2^{94}}$	$\mathbf{2^{94}}$	$\mathbf{2^{83}}$	$\mathbf{2^{110}}$	$\mathbf{2^{110}}$	$\mathbf{2^{120}}$	$\mathbf{2^{94}}$
Attack time 32 R	$\mathbf{2^{119}}$	$\mathbf{2^{102}}$	$\mathbf{2^{94}}$	$\mathbf{2^{87}}$	$\mathbf{2^{110}}$	$\mathbf{2^{110}}$	$\mathbf{2^{120}}$	$\mathbf{2^{94}}$
Cost of 1 key, if	$\mathbf{2^{139}}$	$\mathbf{2^{113}}$	$\mathbf{2^{117}}$	2^{146}	$\mathbf{2^{120}}$	$\mathbf{2^{110}}$	$\mathbf{2^{120}}$	$\mathbf{2^{101}}$
key diversity ≥	2^{52}	2^{75}	2^{84}	2^{84}	2^{98}	2^{84}	2^{70}	2^{79}
Data x keys	2^{84}	2^{107}	2^{121}	2^{116}	2^{130}	2^{116}	2^{102}	2^{111}

Table 3. Major attacks on full GOST cipher: single vs. multiple random keys scenario. Various attacks are here compared according to their capacity to find some keys when weak keys occur at random with their natural probability. In lower table we see that if we allow higher key diversity requirements and more data collected in total (for all keys), the overall time cost to recover one key, this **including** the cost to examine keys which are not weak, decreases down to 2^{101} and beats all known single key attacks.

8.3. Facts or Fictions?

July 2012

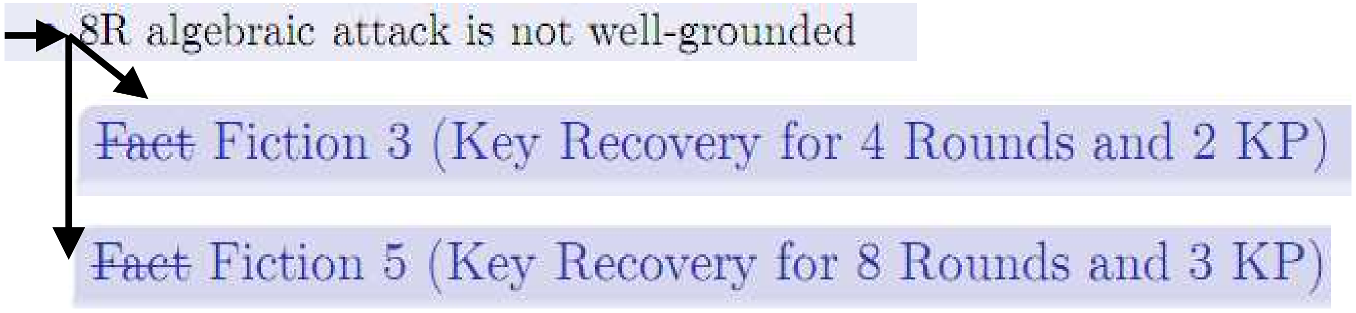
In CTCrypt 2012, workshop held in English, in Russia, July 2012.



Algebraic and Differential Cryptanalysis of GOST:
Fact or Fiction

https://www.tc26.ru/documentary%20materials/CTCrypt%202012/slides/CTCrypt_rudskoy_slides_final.pdf

A. Dmukh, V. Rudskoy



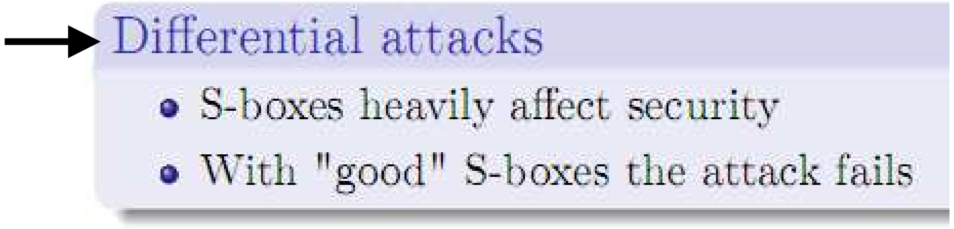
- 8R algebraic attack is not well-grounded

~~Fact~~ Fiction 3 (Key Recovery for 4 Rounds and 2 KP)

~~Fact~~ Fiction 5 (Key Recovery for 8 Rounds and 3 KP)

Easy: try CryptoMiniSat

See Cryptologia Jan 2013 and eprint/2011/626



Differential attacks

- S-boxes heavily affect security
- With "good" S-boxes the attack fails

Super naïve: it makes little sense to take our differential property optimised for one set of S-boxes and apply it to another set of S-boxes.
Another differential property is needed; carefully optimised for this another set of S-boxes…

9. GOST Hash

GOST Hash

Another Russian government standard: GOST-R-34.11-94

Obligatory part of Russian national Digital Signature standard.

Cf. Markus Michels, David Naccache, and Holger Petersen. GOST 34.10 - A brief overview of Russia’s DSA. Computers & Security, 15(8):725–732, 1996.

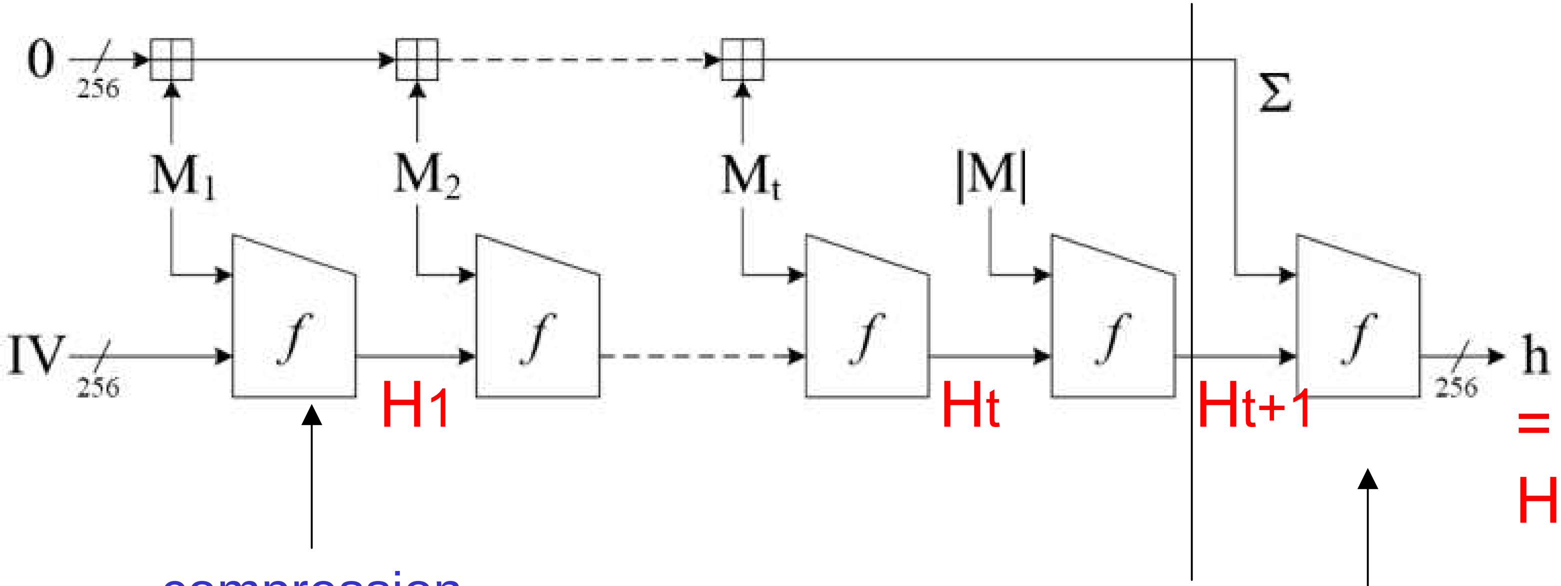
Lots of Applications of GOST Hash:

- Message authentication in (financial) networks.
- Legally binding contracts.
- Trust: electronic commerce (implemented in OpenSSL).

=> An attack on GOST Hash could be potentially much more serious than breaking GOST encryption…

High Level

Very special version of Merkle-Damgard + Len.



H1

Ht

Ht+1

h = Ht+2

compression function

512 → 256

security proof?

works the same way

collision => collision on the compression function

extra component

(not much stronger…)

Collisions on The Compression Function

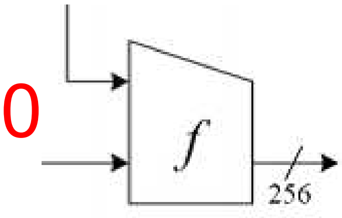
Sometimes called pseudo-collisions:

Because they may use intermediate values (IV or H_i) which will never occur in the real life…

“Certificational Weakness”:

- Any collision on this invalidates the security proof. But does not mean (yet) a real attack.
- Also because these conditions, again by the security proof are NECESSARY to develop collisions of the full hash process, this is a place to start working!

$\Delta\neq0$

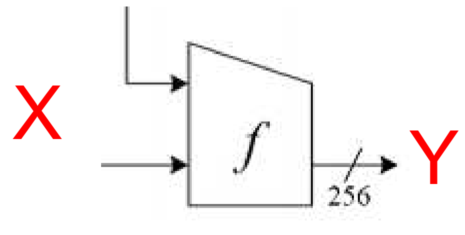


$\Delta=0$

$< 2^{128}$ time

Pre-Images on The Compression Function

given Y compute X



< 2^{255} time

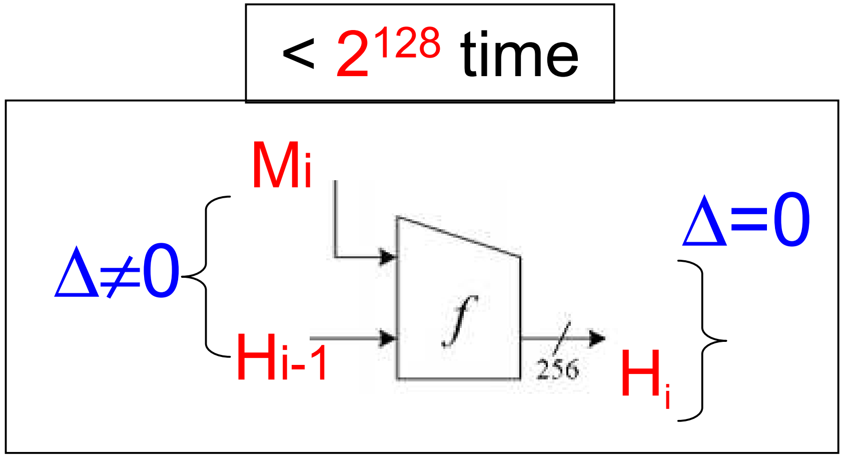
9.1.
How to Break GOST Compression

[Mendel-Pramstaller-Rechberger]
[Courtois- Mourouzis]

Collisions on Compression

Goal:

Pseudo-Collisions:

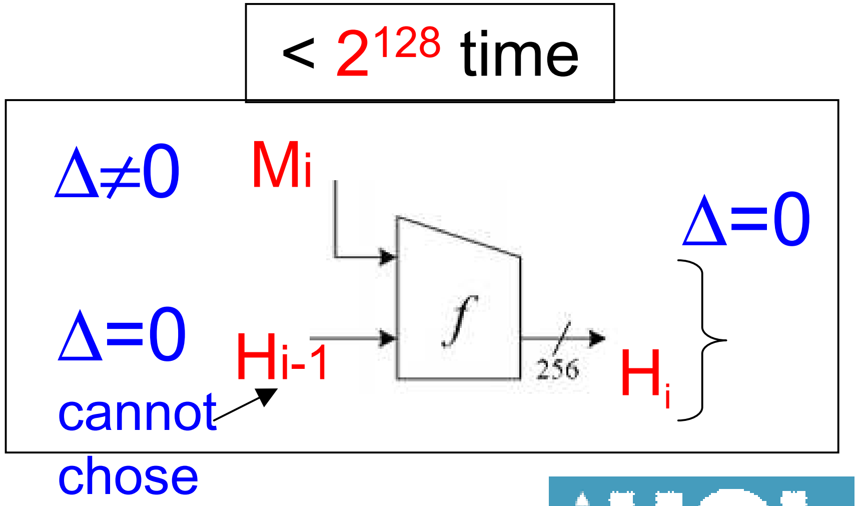


"Stronger" Collisions:

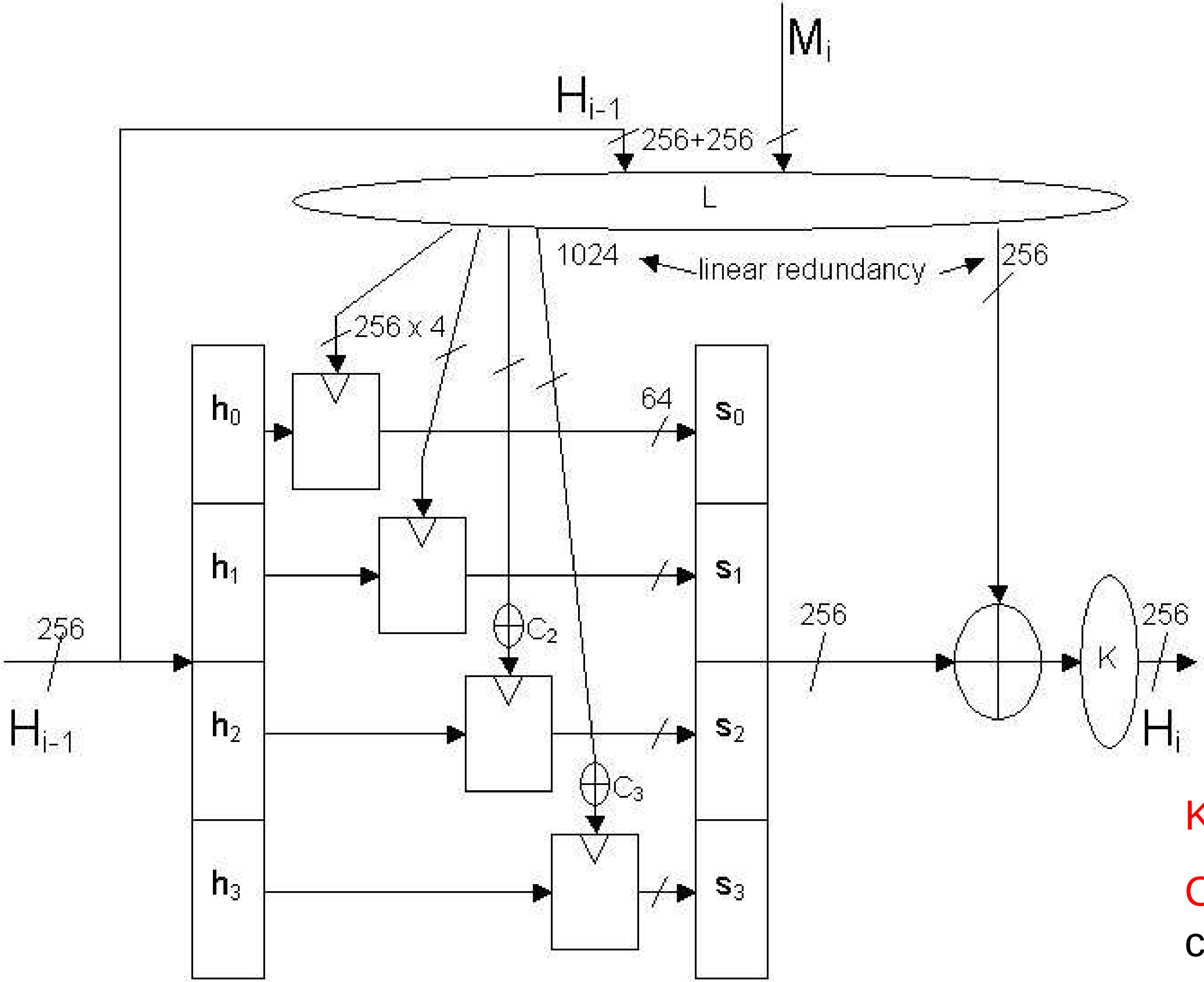
H_{i-1} is arbitrary fixed,

use just M_i to make

it collide nevertheless



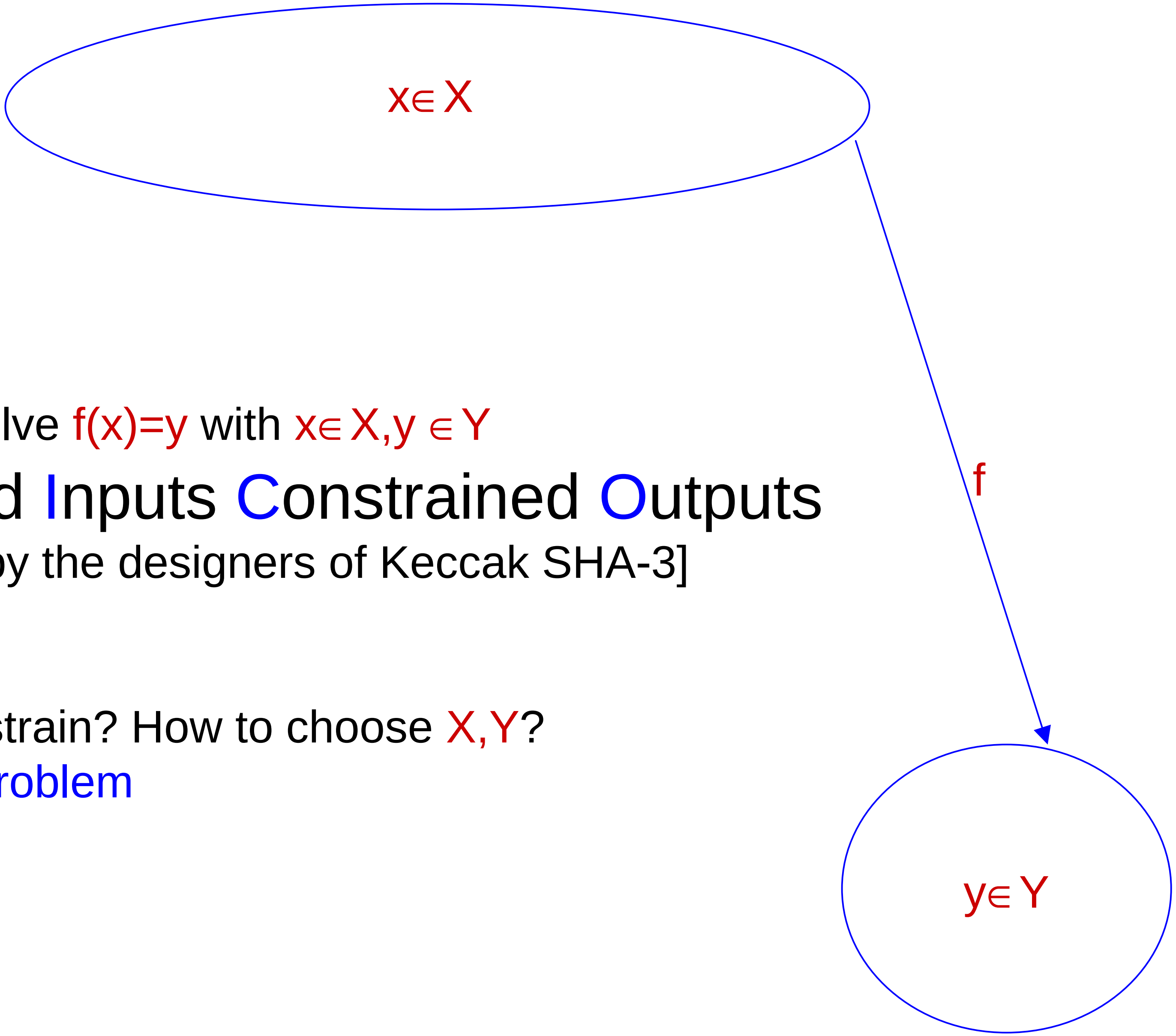
Inside



K,L are linear

C_1,C_2 are constants

CICO



CICO = Solve $f(x)=y$ with $x\in X, y \in Y$

Constrained Inputs Constrained Outputs

[term invented by the designers of Keccak SHA-3]

But how to constrain? How to choose X,Y?

“CICO Setup” problem

Key Idea

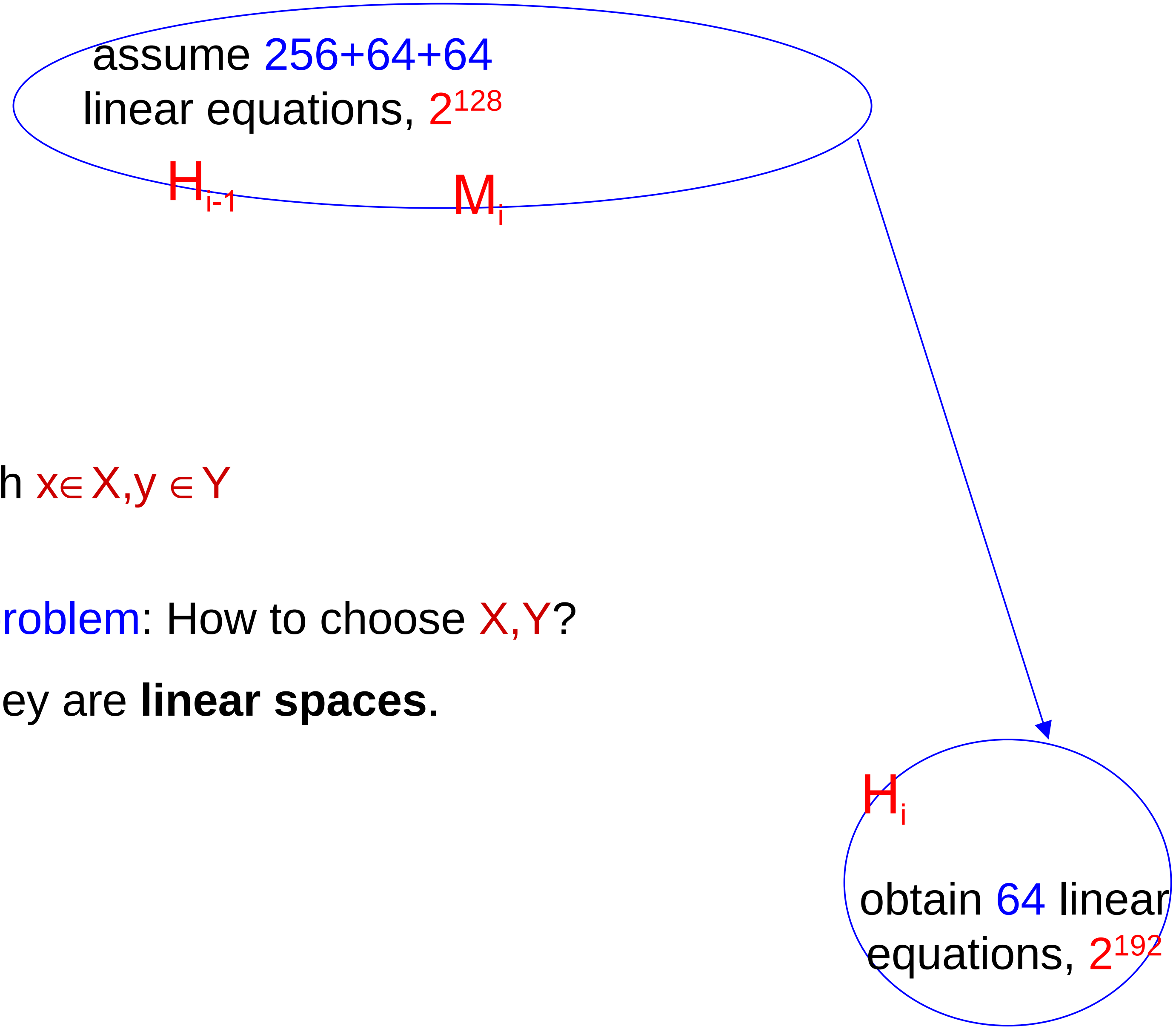
[Mendel-Pramstaller-Rechberger]
[Courtois- Mourouzis]

- Select a number of linear equations on the 512 outputs
- Which induces a smaller linear space for the 256-bit output.

Consequently both Ps.-collisions and preimage attacks are possible.

- For example if the output space is reduced to 2^{192} points, it is like breaking a hash function on 192 bits by brute force / collision search.
- This is if the input space is large enough…

Key Idea



assume 256+64+64 linear equations, 2^{128}

H_{i-1} M_i

H_i

obtain 64 linear equations, 2^{192}

CICO =

Solve $f(x)=y$ with $x \in X, y \in Y$

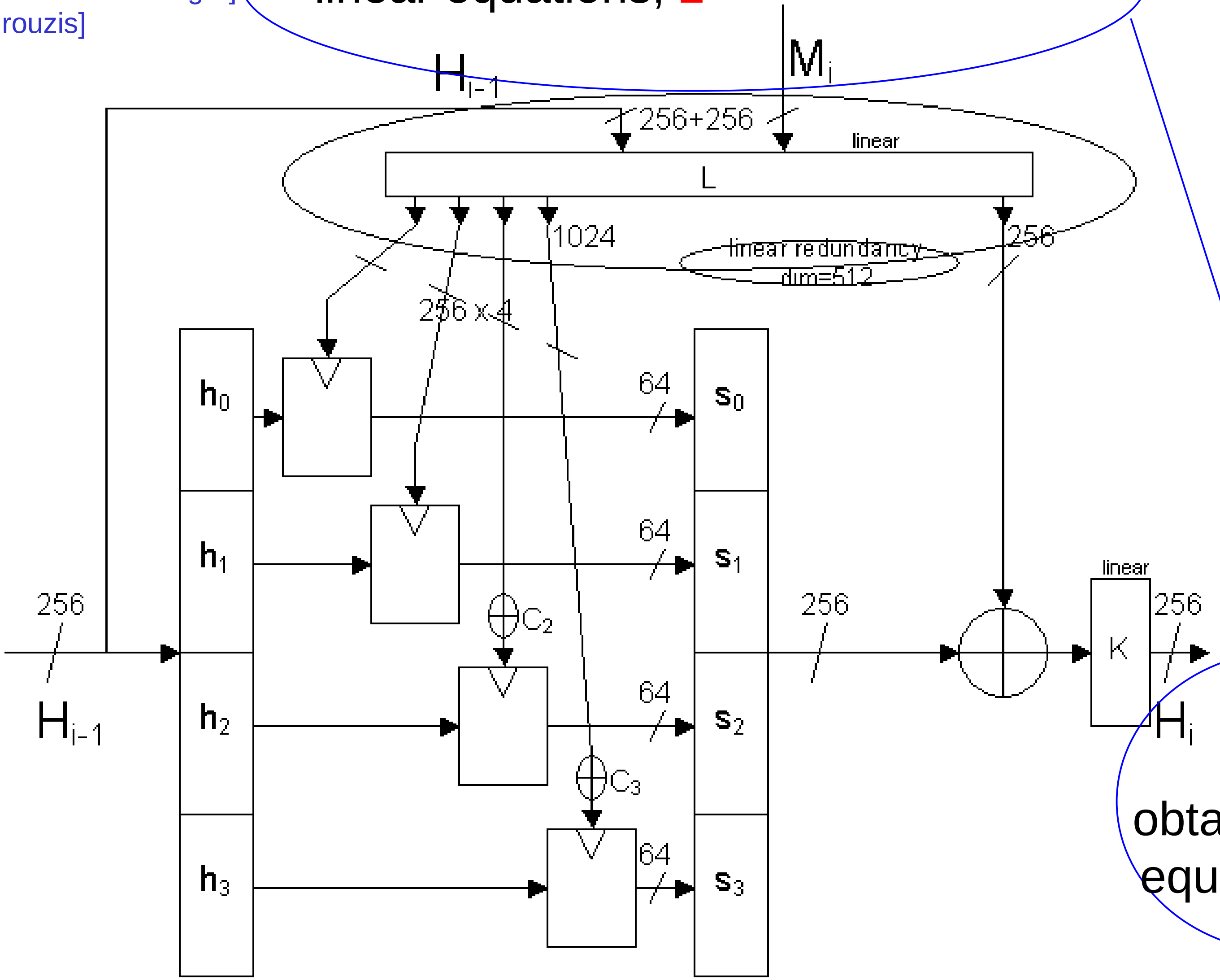
"CICO Setup" problem: How to choose X,Y?

Here they are **linear spaces**.

Attacks:

[Mendel-Pramstaller-Rechberger]
[Courtois- Mourouzis]

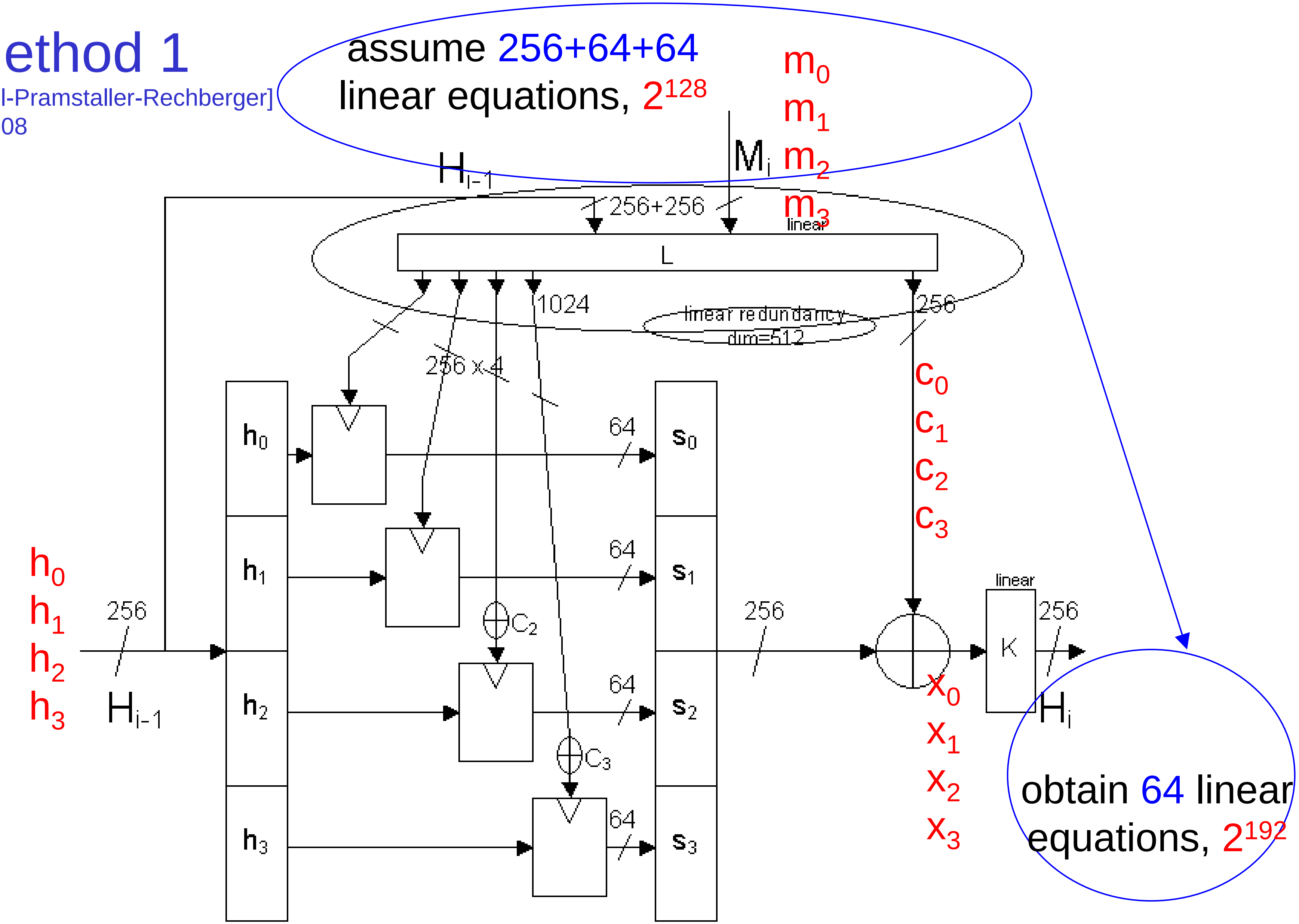
assume 256+64+64 linear equations, 2^{128}



obtain 64 linear equations, 2^{192}

Method 1

[Mendel-Pramstaller-Rechberger] FSE 2008



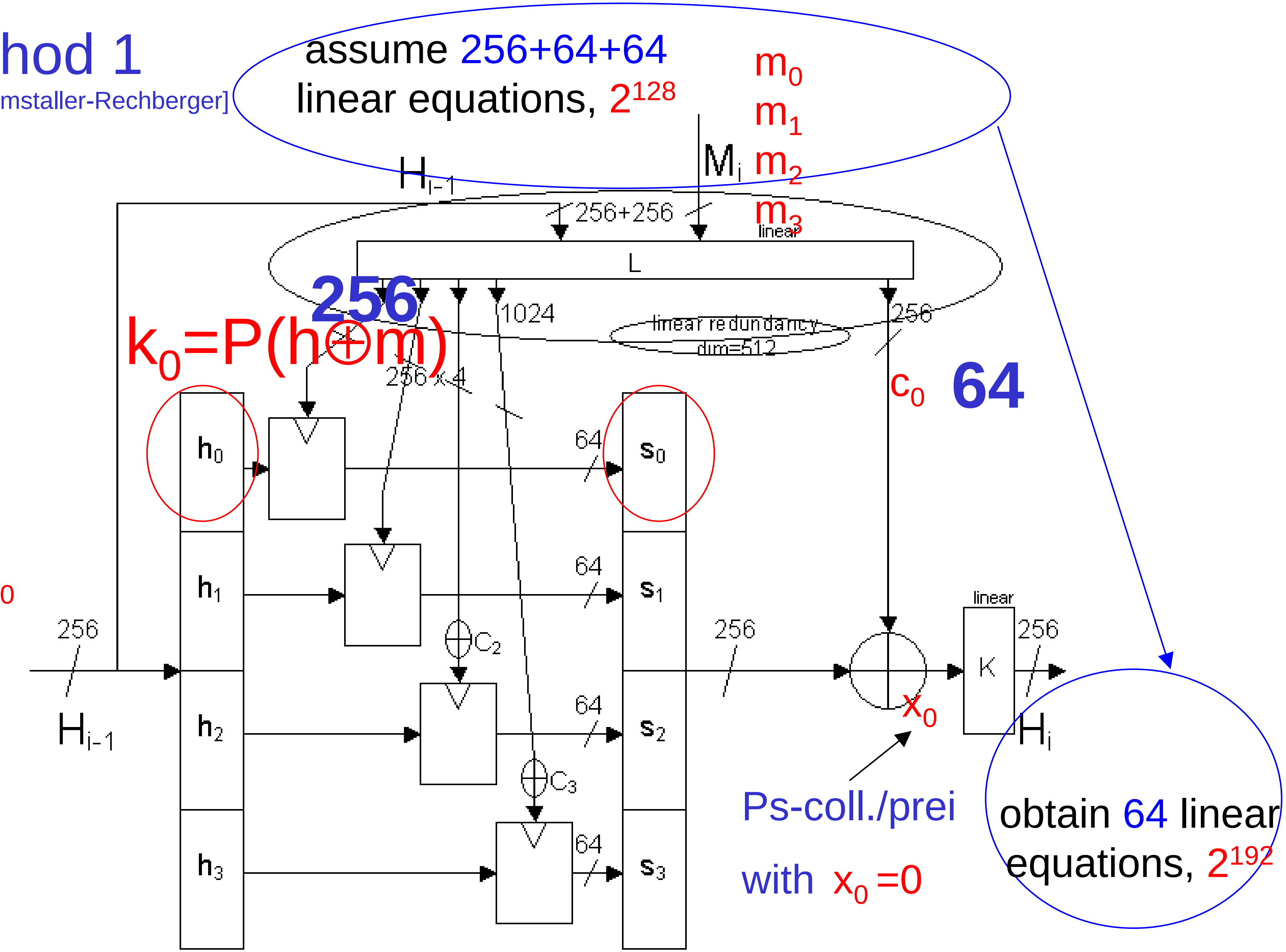
assume 256+64+64 linear equations, 2^{128}

obtain 64 linear equations, 2^{192}

Method 1

[Mendel-Pramstaller-Rechberger] FSE 2008

assume 256+64+64 linear equations, 2^{128}



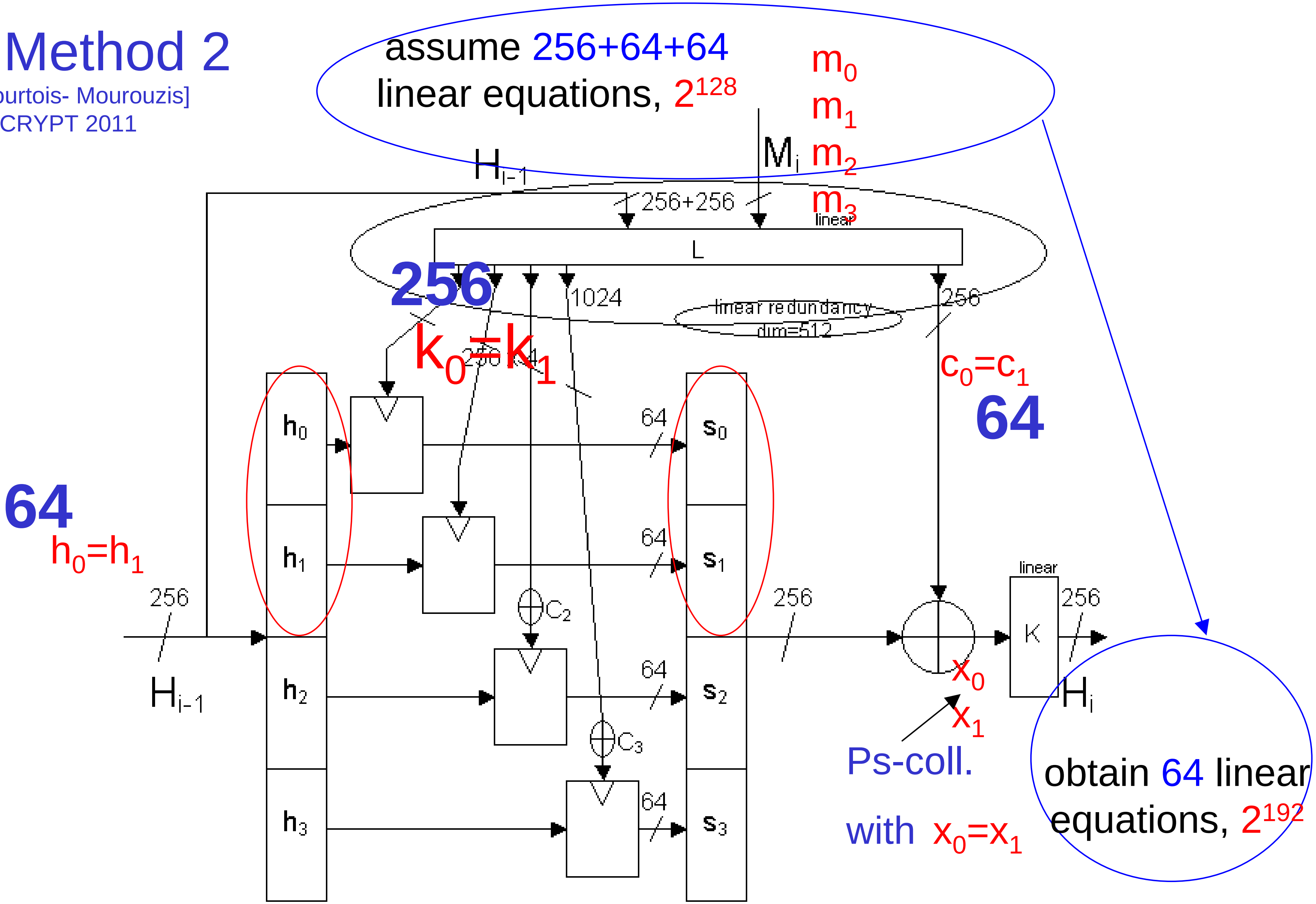
Ps-coll./prei

with x_0 =0

obtain 64 linear equations, 2^{192}

Method 2

[Courtois- Mourouzis]
SECRYPT 2011



assume 256+64+64 linear equations, 2^{128}

m_0
m_1
m_2
m_3

256
$k_0=k_1$

$c_0=c_1$
64

64
$h_0=h_1$

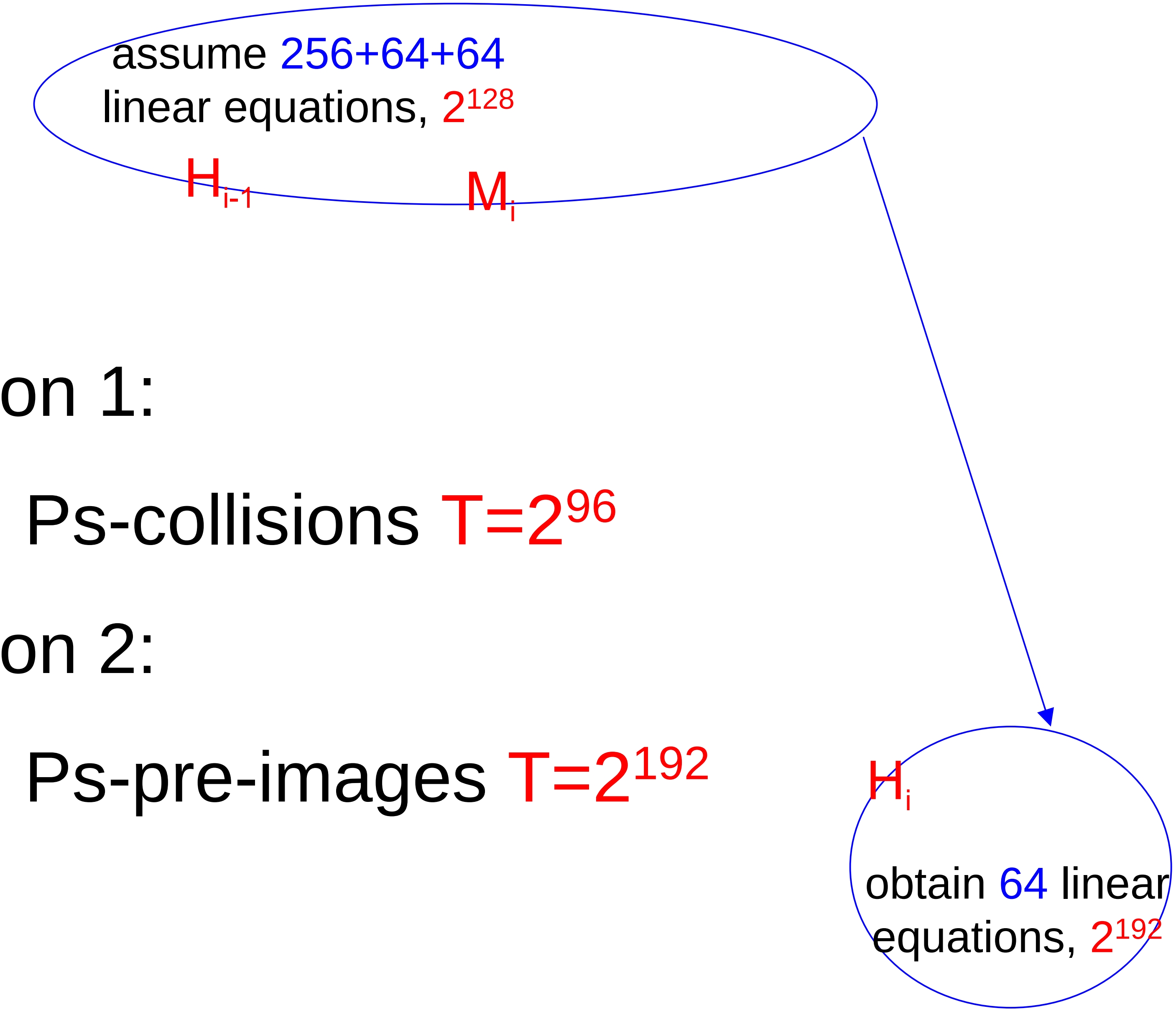
x_0
x_1

Ps-coll.

with $x_0=x_1$

obtain 64 linear equations, 2^{192}

Why Do This?



assume 256+64+64 linear equations, 2^{128}

H_{i-1} M_i

H_i

obtain 64 linear equations, 2^{192}

Application 1:

find Ps-collisions $T=2^{96}$

Application 2:

find Ps-pre-images $T=2^{192}$

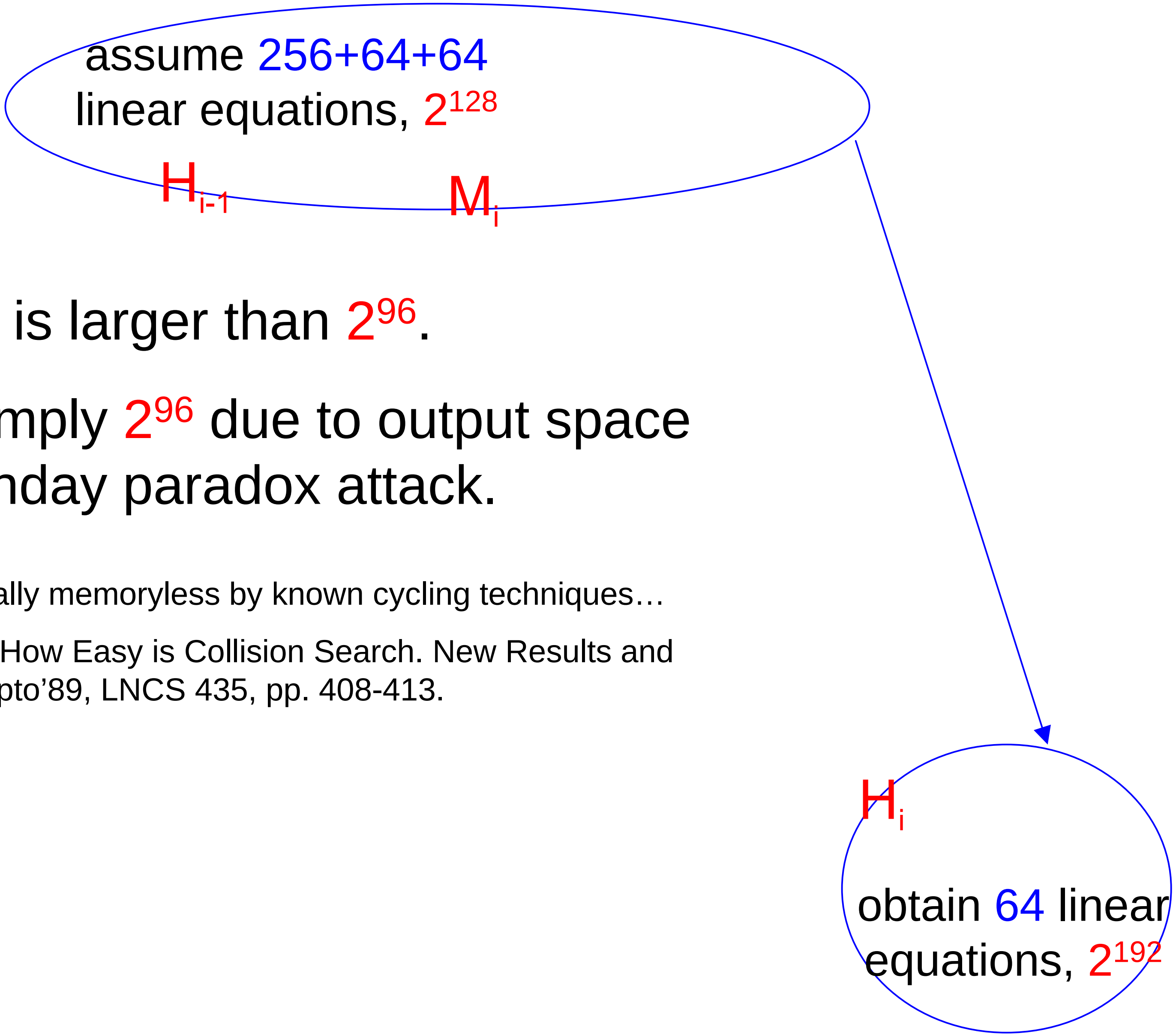
Pseudo-Collisions

$T= 2^{96} < 2^{128}$

Ps-Collisions

[Mendel-Pramstaller-Rechberger] FSE 2008 appendix

Also works with our Method 2!



assume 256+64+64 linear equations, 2^{128}

H_{i-1} M_i

H_i

obtain 64 linear equations, 2^{192}

Our input space is larger than 2^{96}.

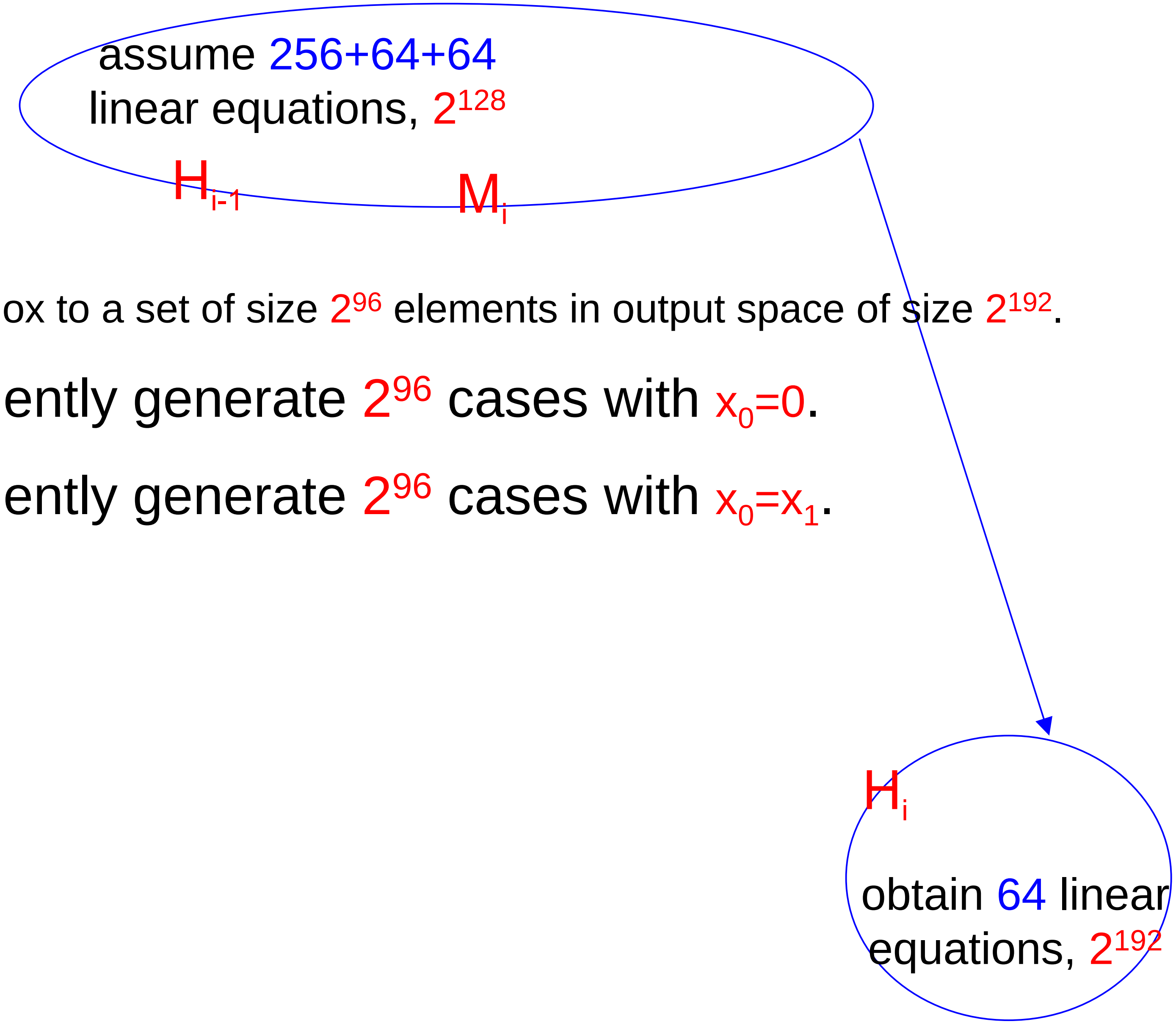
Complexity is simply 2^{96} due to output space size of 2^{192}. Birthday paradox attack.

Important: can be made totally memoryless by known cycling techniques…

Cf. Quisquater-Delescaille, How Easy is Collision Search. New Results and Applications to DES. In Crypto’89, LNCS 435, pp. 408-413.

Ps-Collisions

Easy, several methods



assume 256+64+64 linear equations, 2^{128}

H_{i-1} M_i

H_i

obtain 64 linear equations, 2^{192}

Apply birthday paradox to a set of size 2^{96} elements in output space of size 2^{192}.

Method 1: Efficiently generate 2^{96} cases with $x_0=0$.

Method 2: Efficiently generate 2^{96} cases with $x_0=x_1$.

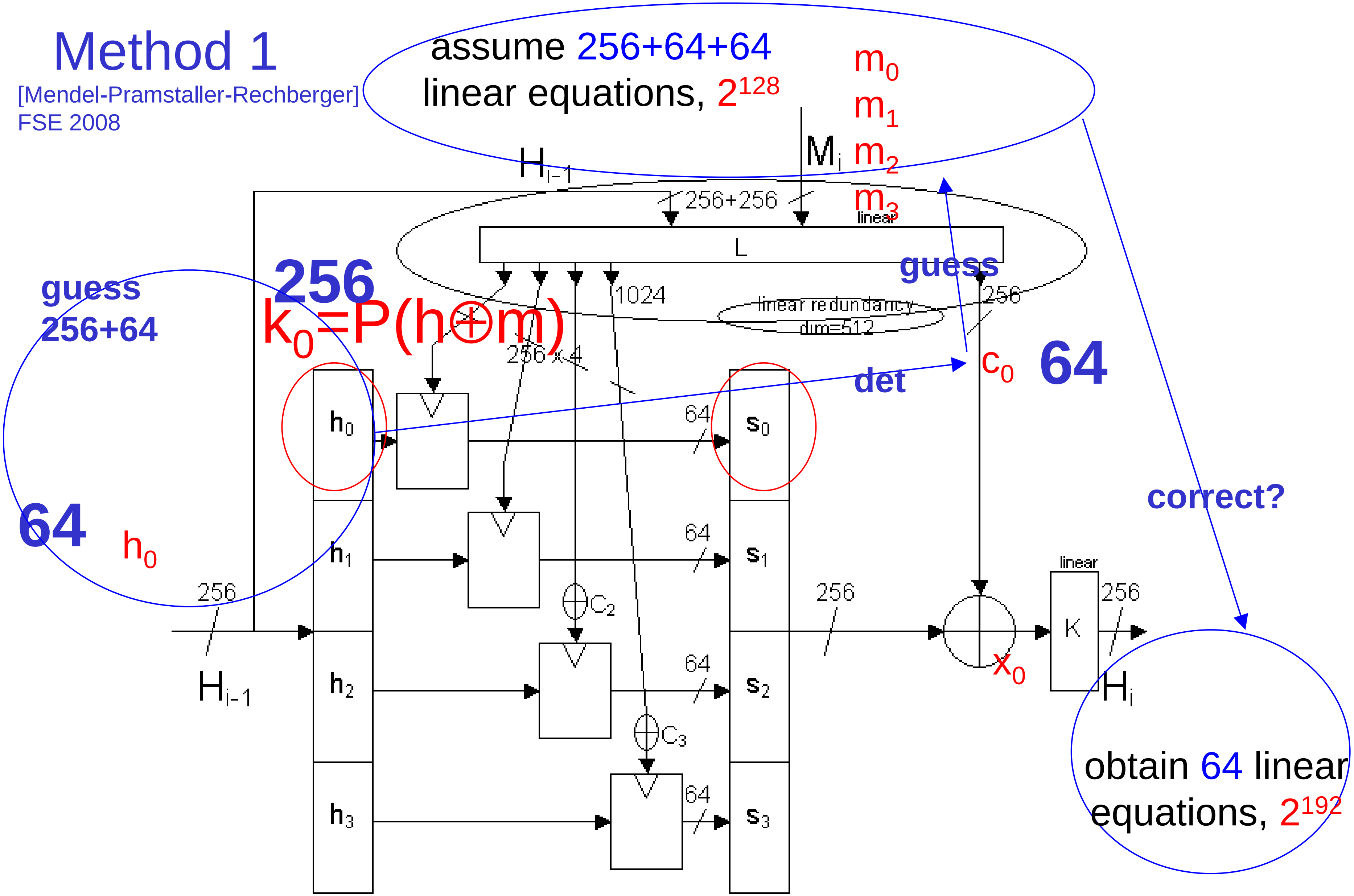
Ps-Pre-Images

$T=2^{192} < 2^{255}$

fewer methods

Method 1

[Mendel-Pramstaller-Rechberger] FSE 2008



assume 256+64+64 linear equations, 2^{128}

guess 256+64

$k_0=P(h\oplus m)$

guess

det

correct?

obtain 64 linear equations, 2^{192}

Pre-Images

[Mendel-Pramstaller-Rechberger] FSE 2008

with Method 1

assume 256+64+64 linear equations, 2^{128}

H_{i-1} M_i

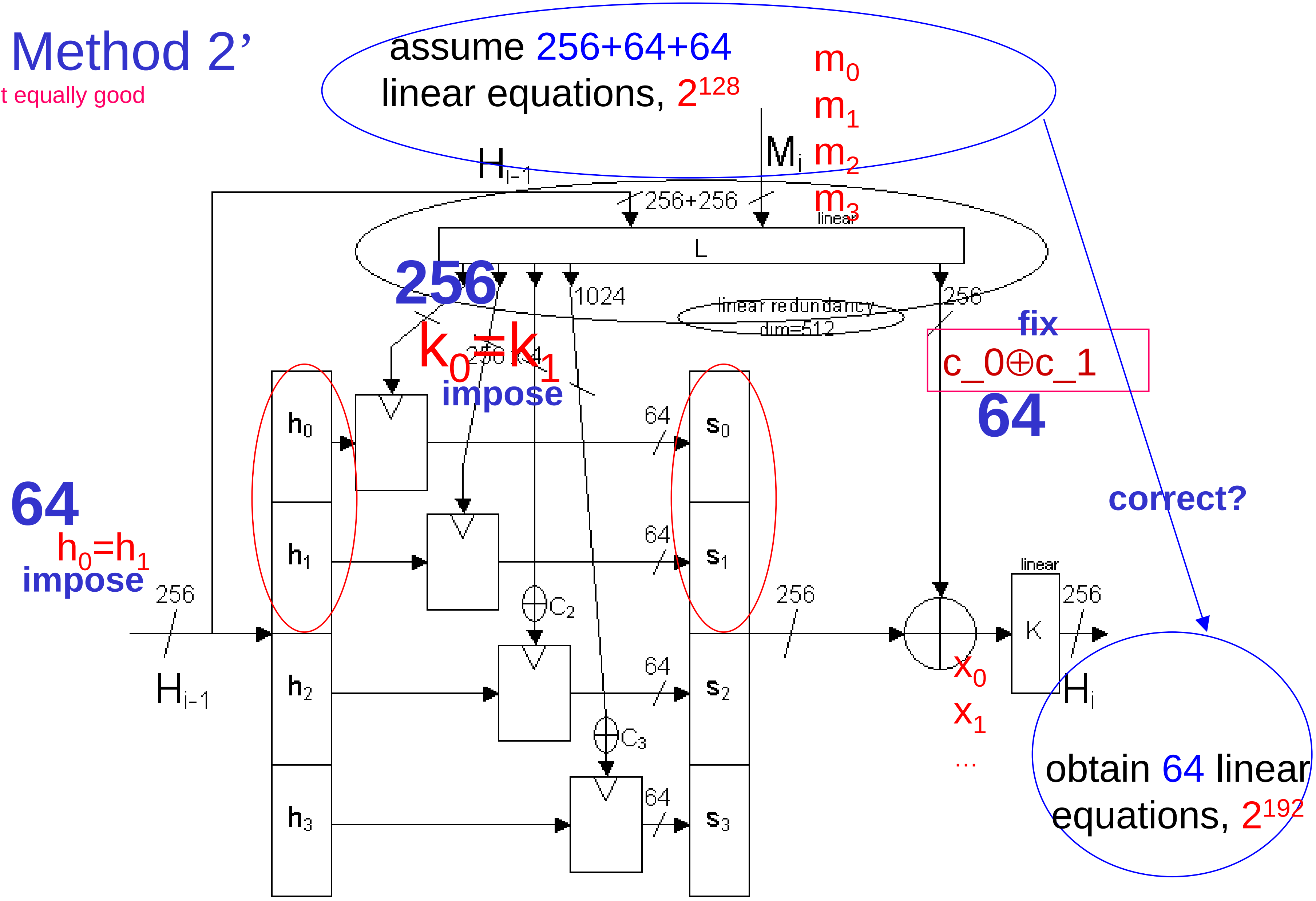
- With Method 1 we can first of all chose h_0,k_0 and compute c_0 which we need to obtain for a correct target value x_0.
- now the triple of values (h_0,k_0,c_0) determines 256+64+64 linear equations we fix for the inputs.
- Random input produces the output we want with probability 2^{-192}. Time complexity is simply 2^{192}.
- For every h_0,k_0 we can determine c_0 and explore the input space with 2^{128} points. In total we can explore $2^{256+64+128}$ possibilities, more than 2^{192} necessary.

H_i

obtain 64 linear equations, 2^{192}

Method 2’

not equally good



Pre-Images

with Method 2' [not so good]

assume 256+64+64 linear equations, 2^{128}

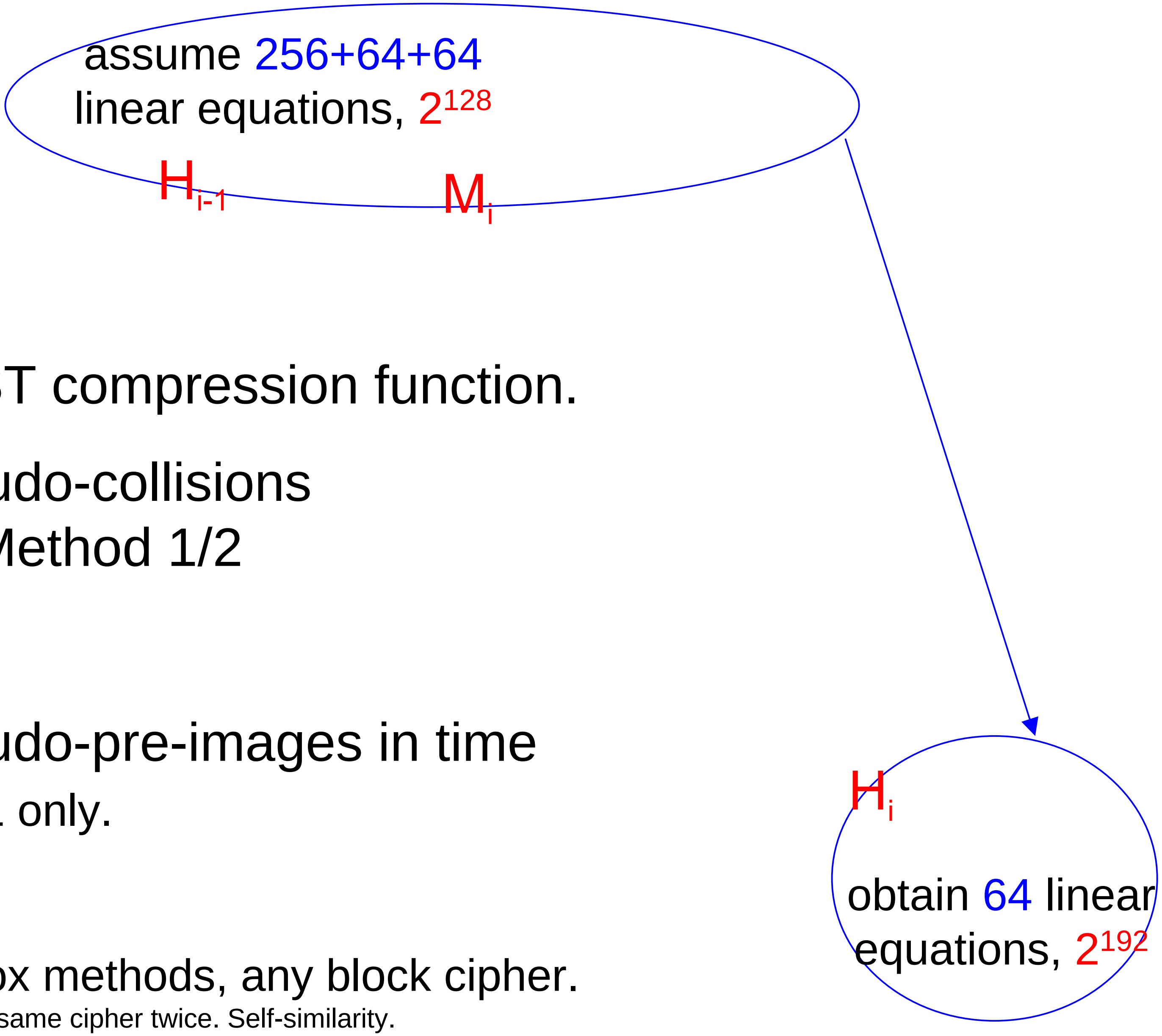
H_{i-1} M_i

H_i

obtain 64 linear equations, 2^{192}

- With Method 2' we can fix c_0⊕c_1 such that s_0⊕s_1=0 which we want to impose, 64 affine equations.
- Other 64+256 linear equations as in Method 2: $k_0=k_1$ and $h_0=h_1$.
- Now random input produces the output we want with probability 2^{-192}. Complexity is again 2^{192}.
- Problem: input space is only 2^{128}. Works with proba 2^{-64}.
- Six basic variants with 2 out of 4: Works with proba $2^{-61.4}$.
- Due to GOST complementation we get $2^{-60.4}$.
- This attack only works for some final outputs.

Conclusion


assume 256+64+64 linear equations, 2^{128}

obtain 64 linear equations, 2^{192}

For the GOST compression function.

We find pseudo-collisions in time 2^{96}. Method 1/2

We find pseudo-pre-images in time 2^{192}. Method 1 only.

100 % black-box methods, any block cipher.

In Method 2 needs to be same cipher twice. Self-similarity.

10. Diffusion in GOST

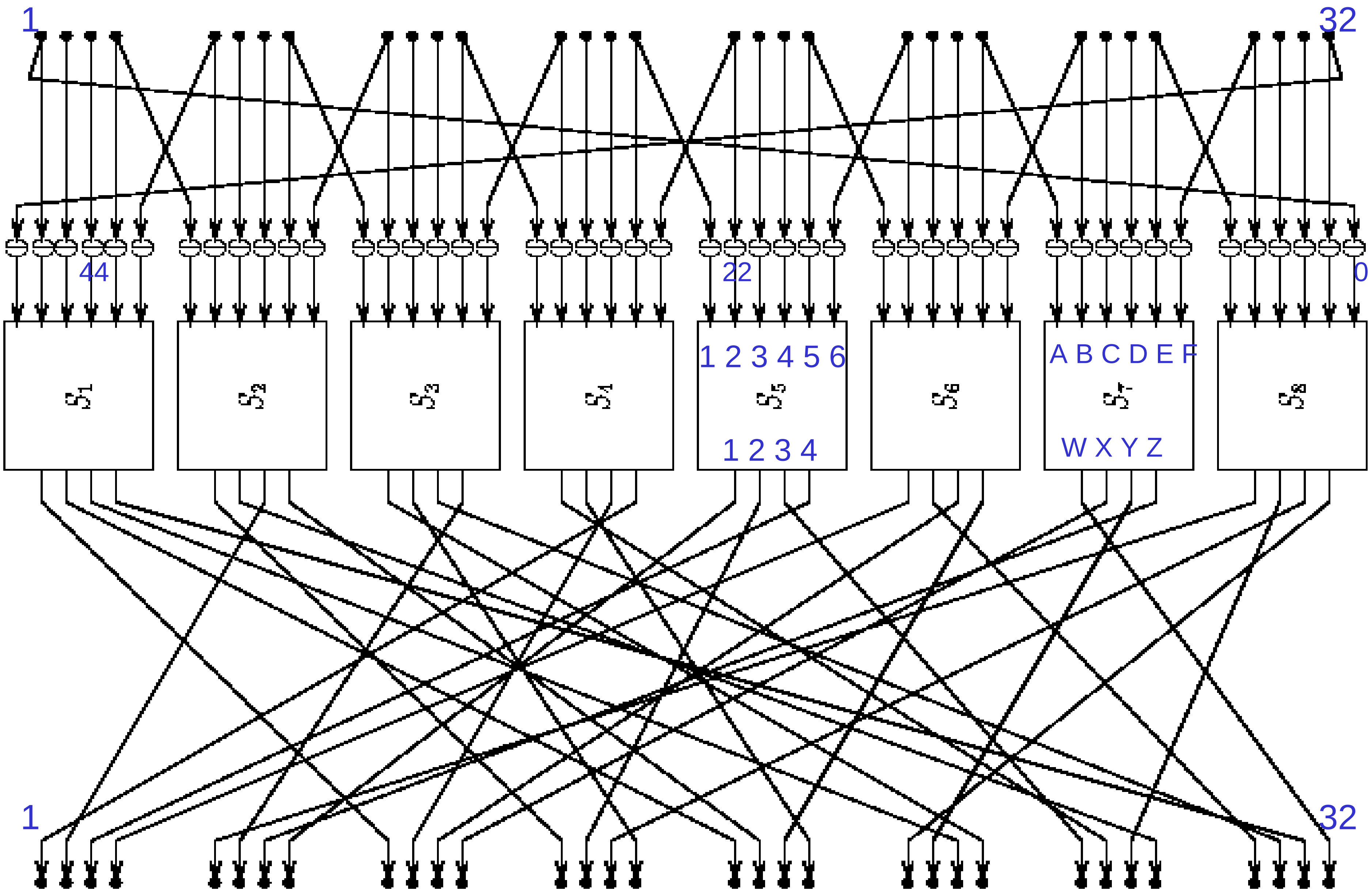
*Claims on GOST

Wikipedia April 2011:

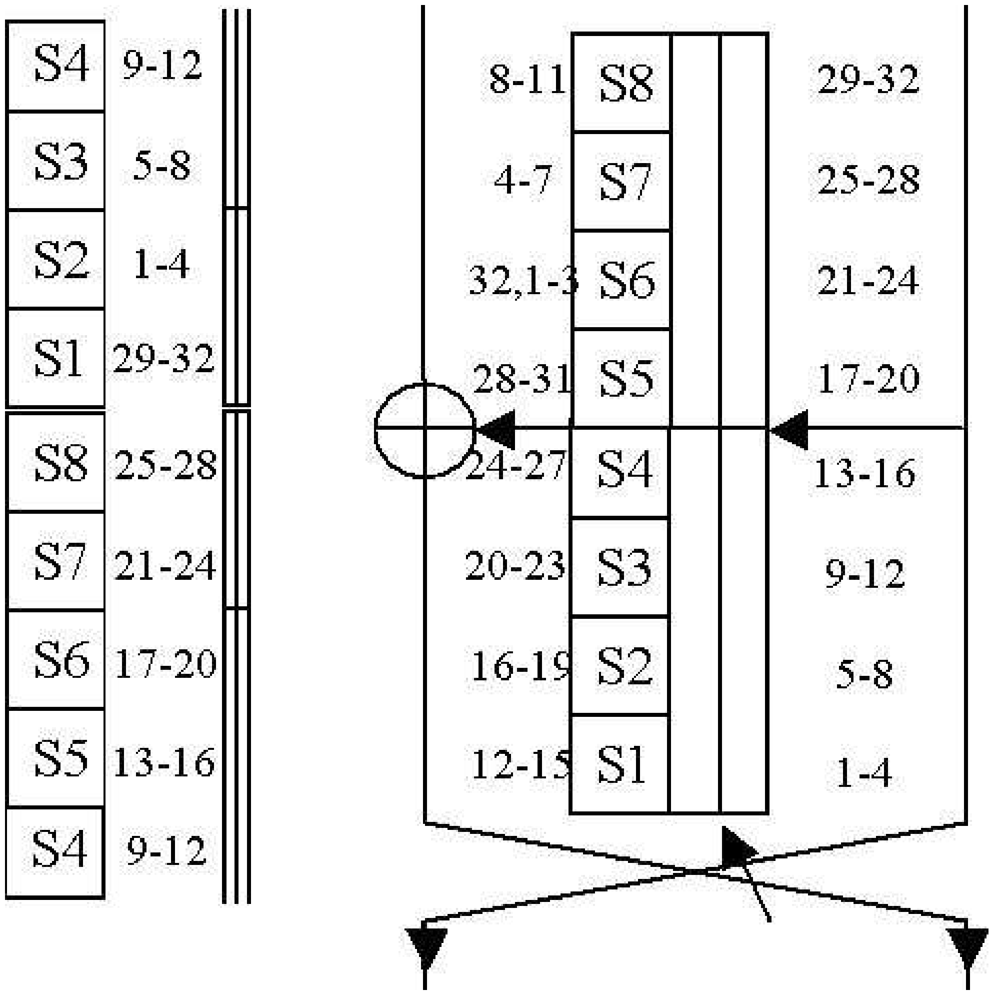
Cryptanalysis of GOST

…Another concern is that the avalanche effect is slower to occur in GOST than in DES.
This is because of GOST's lack of an expansion permutation in the round function,
as well as its use of a rotation instead of a permutation.
Again, this is offset by GOST's increased number of rounds…

DES:



1 Round + Next Round of GOST



Carry Propagation

determine a:

need S3, S4 and c

3 1 1

d,e known

=> $2^{0.6}$ possibilities

↓

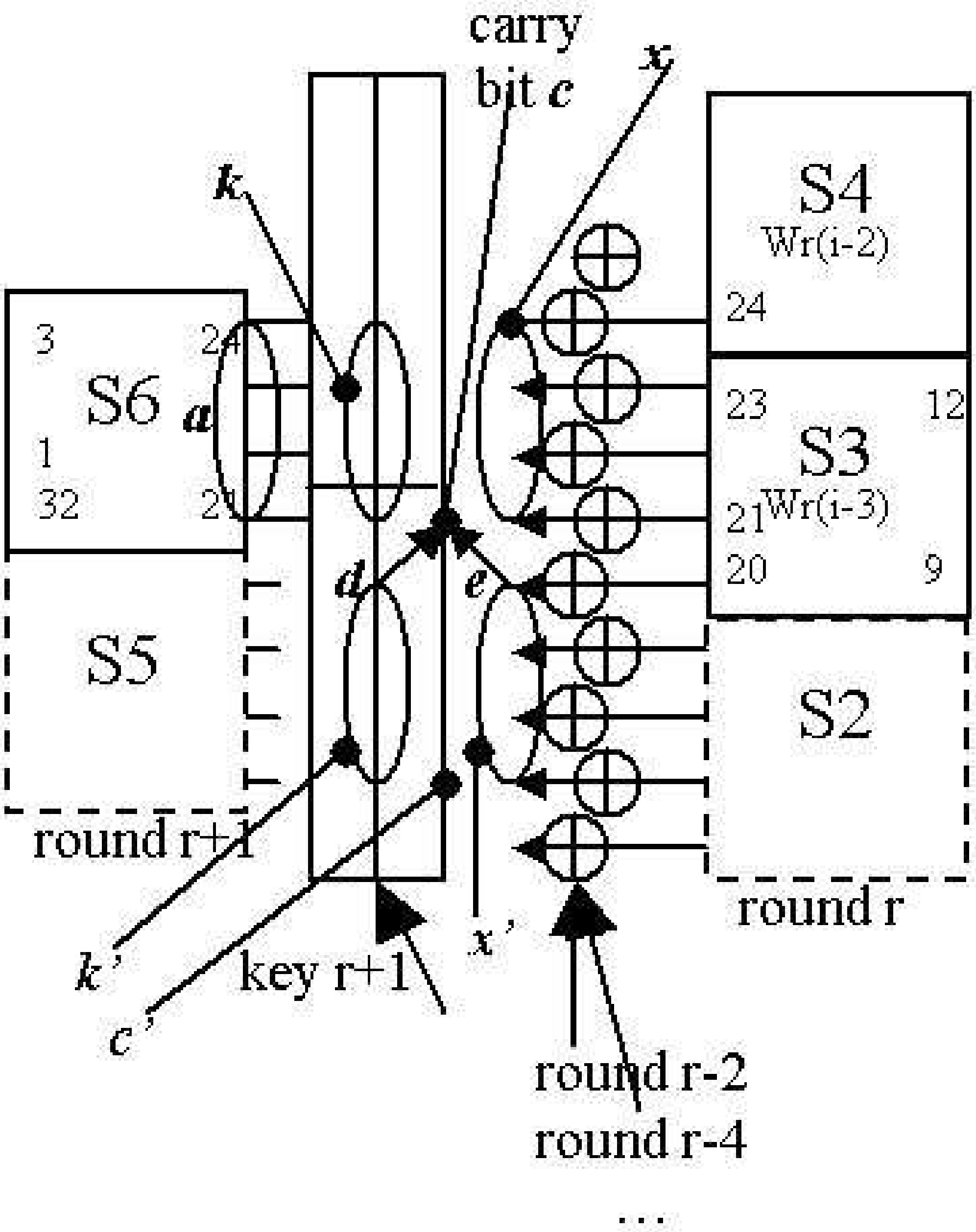
3 more bits known

=> $2^{0.3}$ possibilities

↓

…

$2^{0.0}$

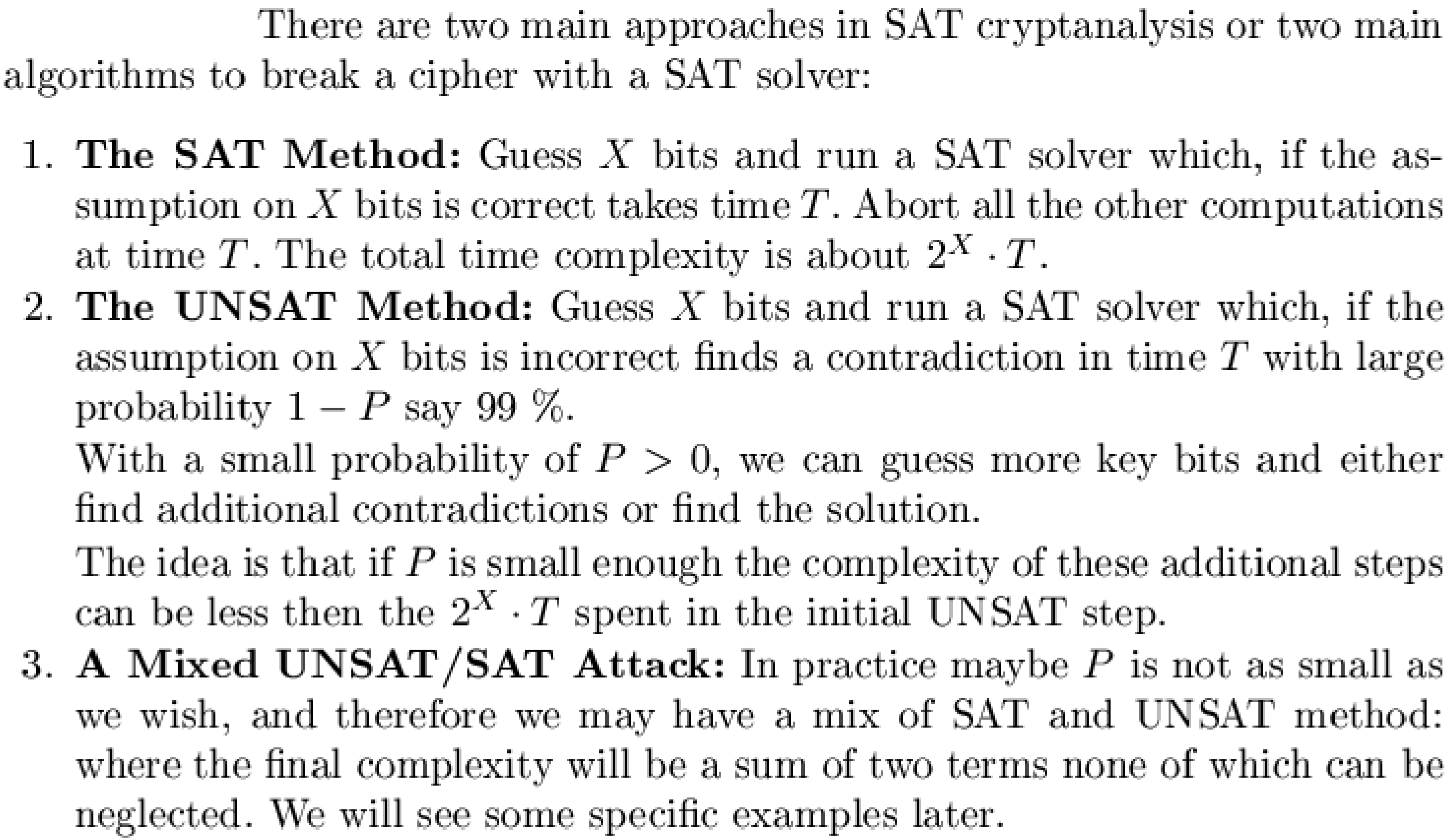


10.2. Guess-Then-Determine: What to Guess?

10.2.1. Contradiction Immunity

Attacks With SAT Solvers

2 strategies:

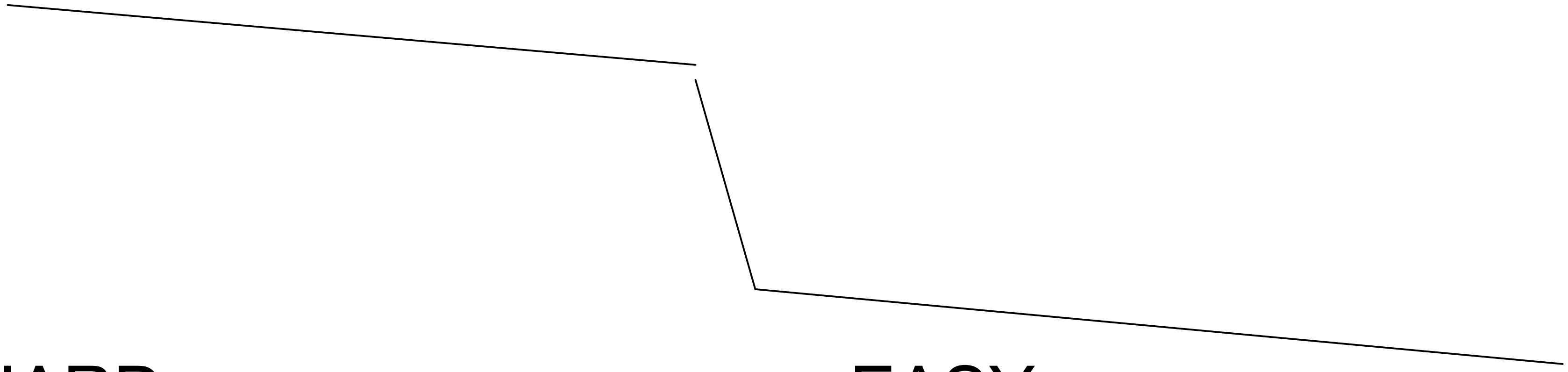


There are two main approaches in SAT cryptanalysis or two main algorithms to break a cipher with a SAT solver:

1. **The SAT Method:** Guess X bits and run a SAT solver which, if the assumption on X bits is correct takes time T. Abort all the other computations at time T. The total time complexity is about $2^X \cdot T$.
2. **The UNSAT Method:** Guess X bits and run a SAT solver which, if the assumption on X bits is incorrect finds a contradiction in time T with large probability $1 - P$ say 99 %.
   With a small probability of $P > 0$, we can guess more key bits and either find additional contradictions or find the solution.
   The idea is that if P is small enough the complexity of these additional steps can be less then the $2^X \cdot T$ spent in the initial UNSAT step.
3. **A Mixed UNSAT/SAT Attack:** In practice maybe P is not as small as we wish, and therefore we may have a mix of SAT and UNSAT method: where the final complexity will be a sum of two terms none of which can be neglected. We will see some specific examples later.

Phase Transitions for Naïve Cryptologists:

1 dimensional



HARD ………………………… EASY

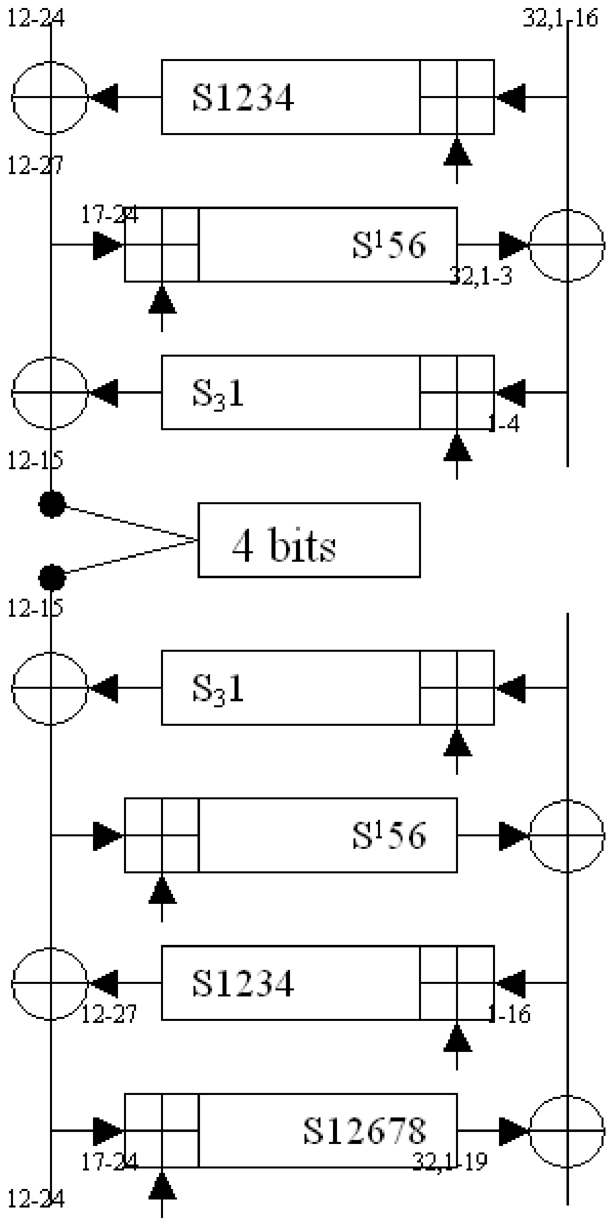
For Serious Cryptologists:

In fact we need to look
at an exponential number of subsets!

UNSAT Immunity

Well chosen set of 68 bits.

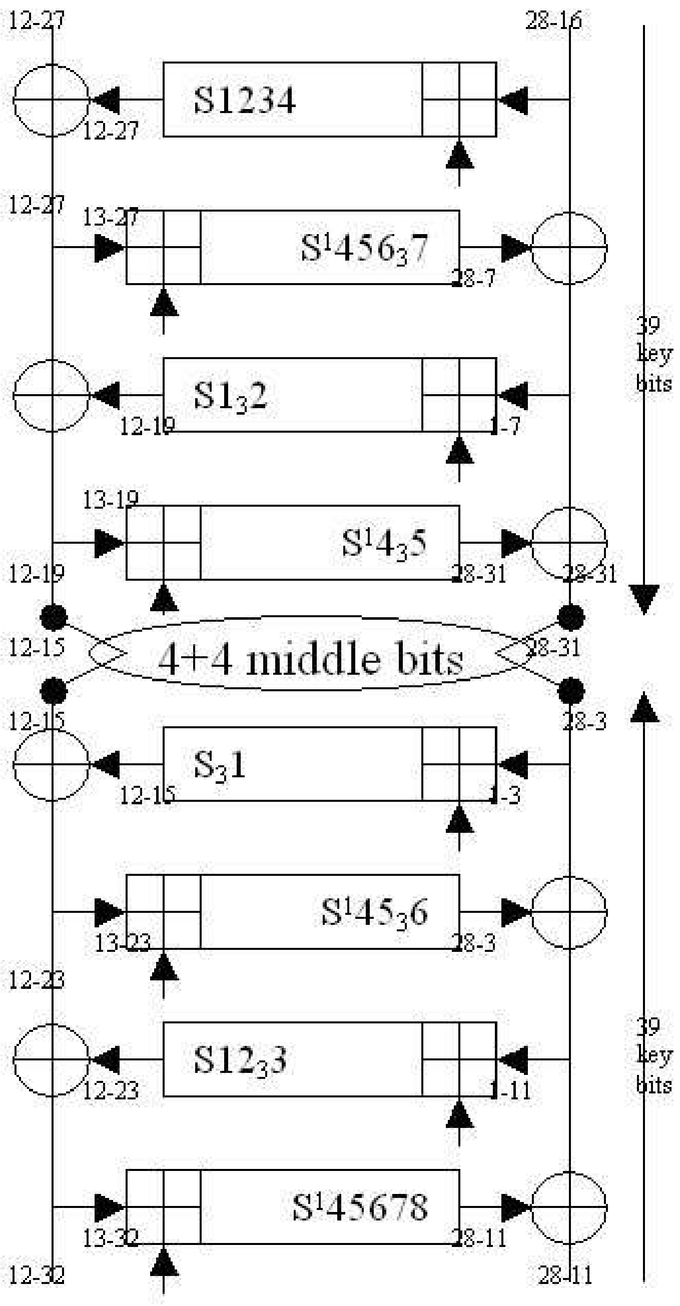
UNSAT proba=39%.



Jumps…

To increase 39% to 50%
we need 10 more bits
= 78 bits.

UNSAT proba=50%.

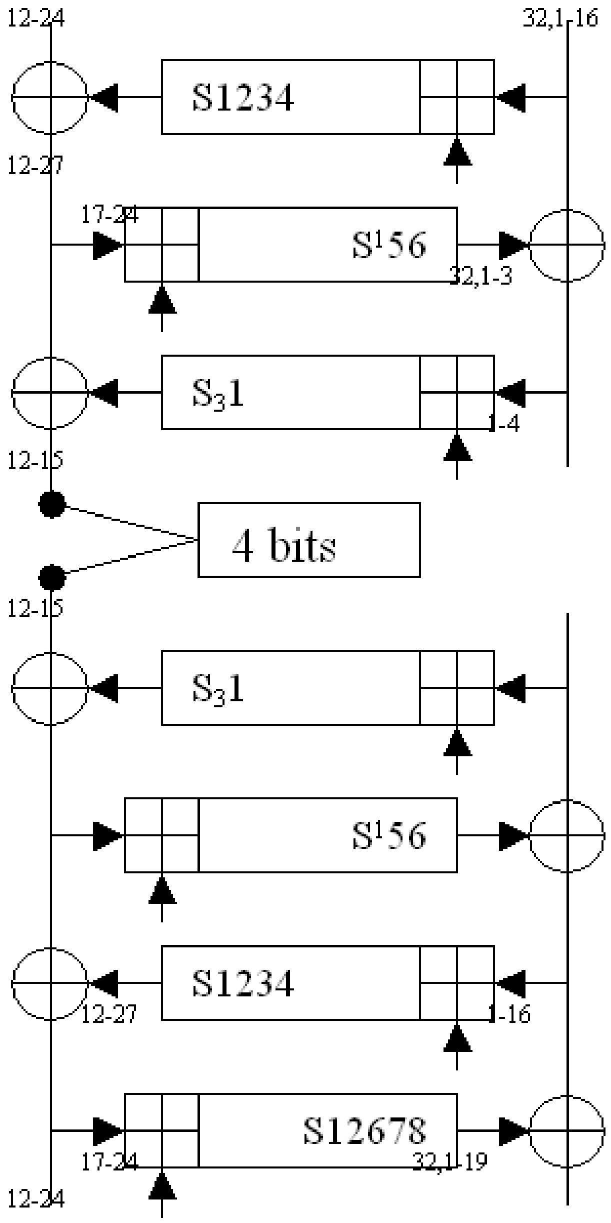


SAT Immunity

Same set of 68 bits as before.

All the other bits
are found in 400 s on
one laptop i7 CPU

=> using CryptoMiniSat x64 2.92.



UNSAT Immunity in DES

Fact 1. The Contradiction Immunity is at most 44 for 8 rounds of DES.

For 8 rounds of GOST:

it is 78 [unpublished set].

More on UNSAT Immunity

See:

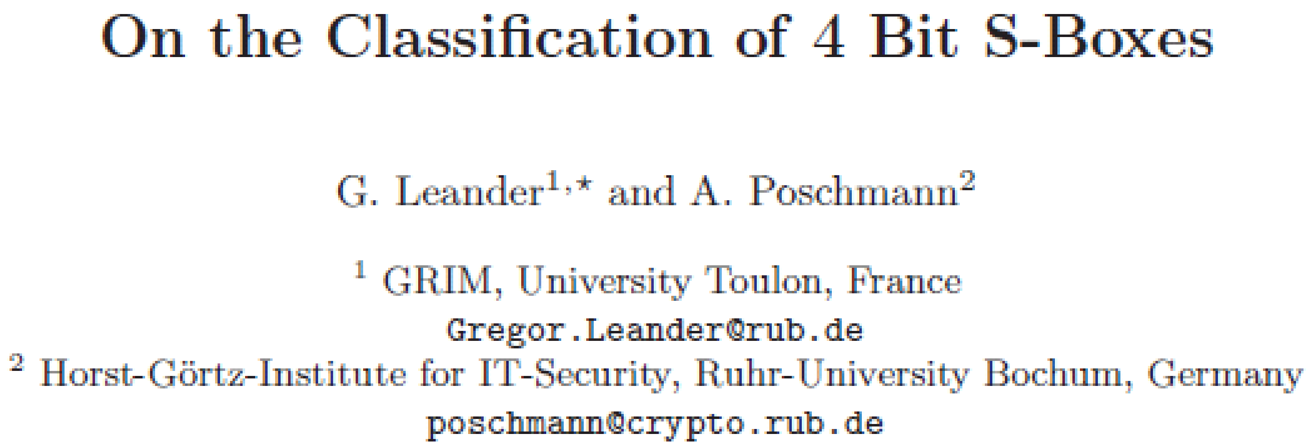
Nicolas Courtois, Jerzy A. Gawinecki, Guangyan Song: Contradiction Immunity and Guess-Then-Determine Attacks On GOST, In Tatra Mountains Mathematic Publications, 53 (2013), pp. 1-15?.

****

Multiplicative Complexity in GOST
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…

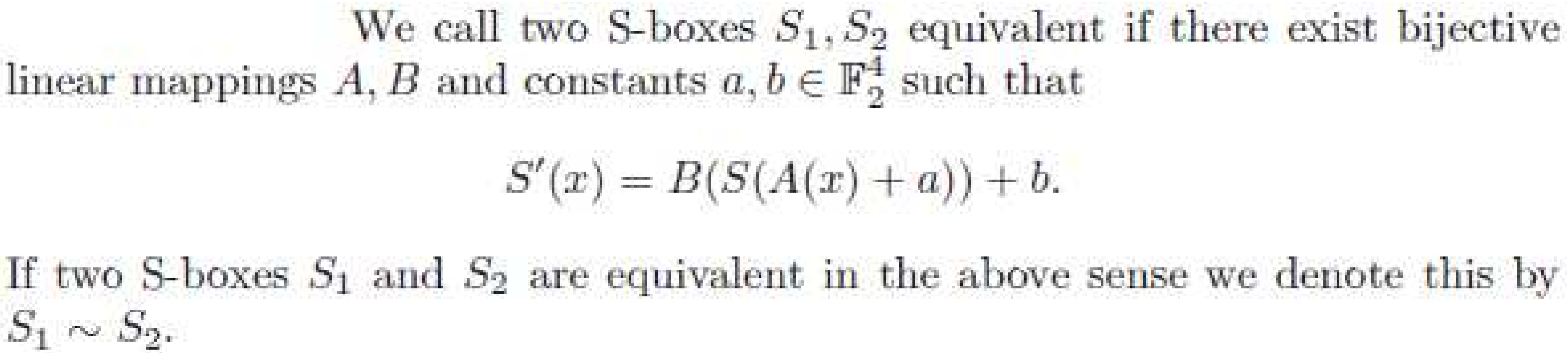


On the Classification of 4 Bit S-Boxes

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

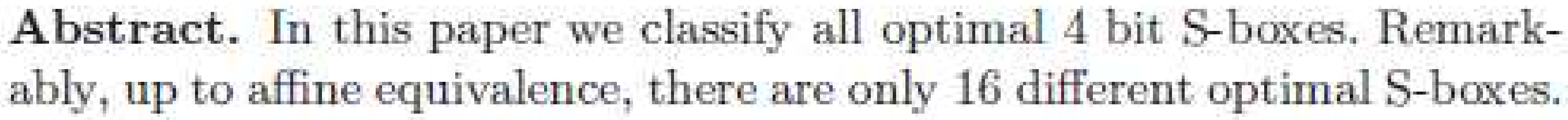
Affine Equivalence



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

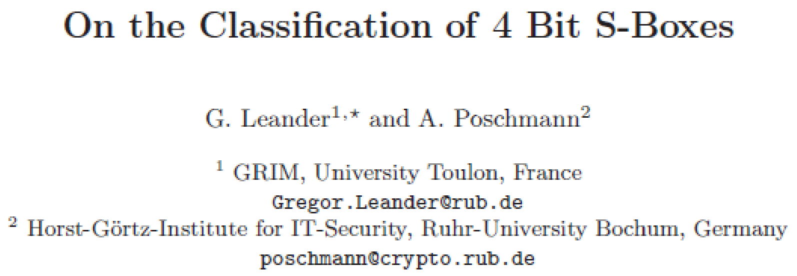
$$S'(x) = B(S(A(x) + a)) + b.$$

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



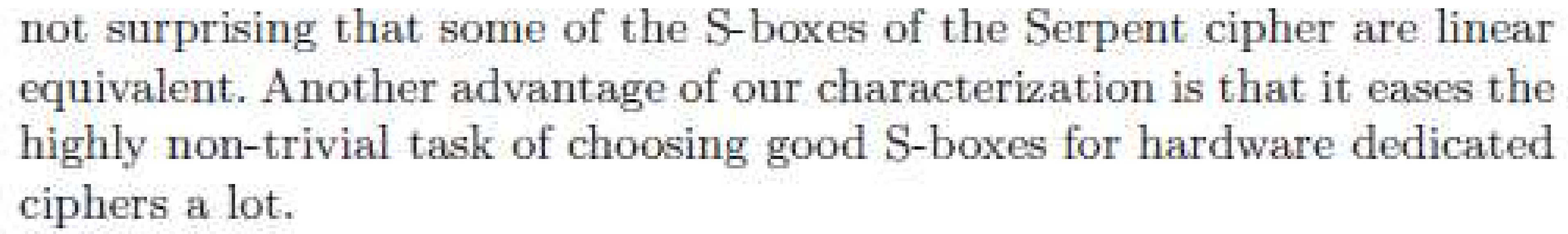
On the Classification of 4 Bit S-Boxes

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

Only 16 S-boxes are "good".

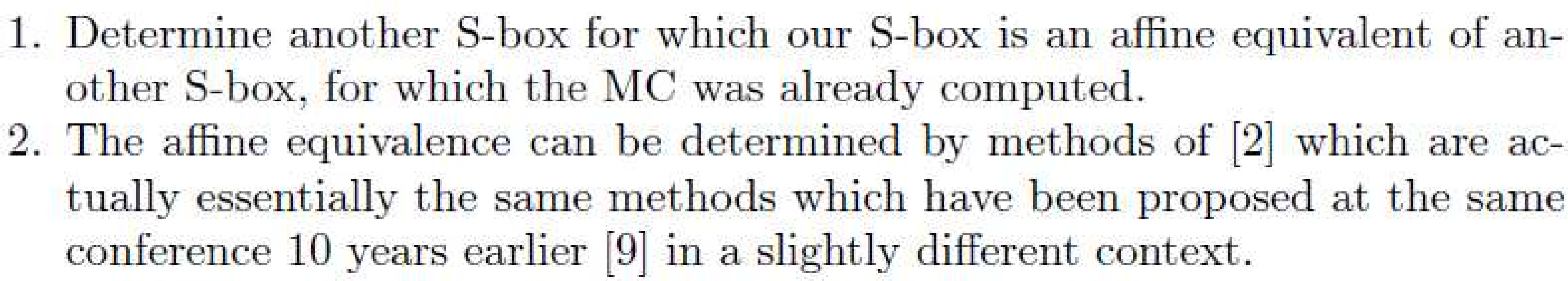
4x4 occur in Serpent, PRESENT, GOST, [AES…]



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

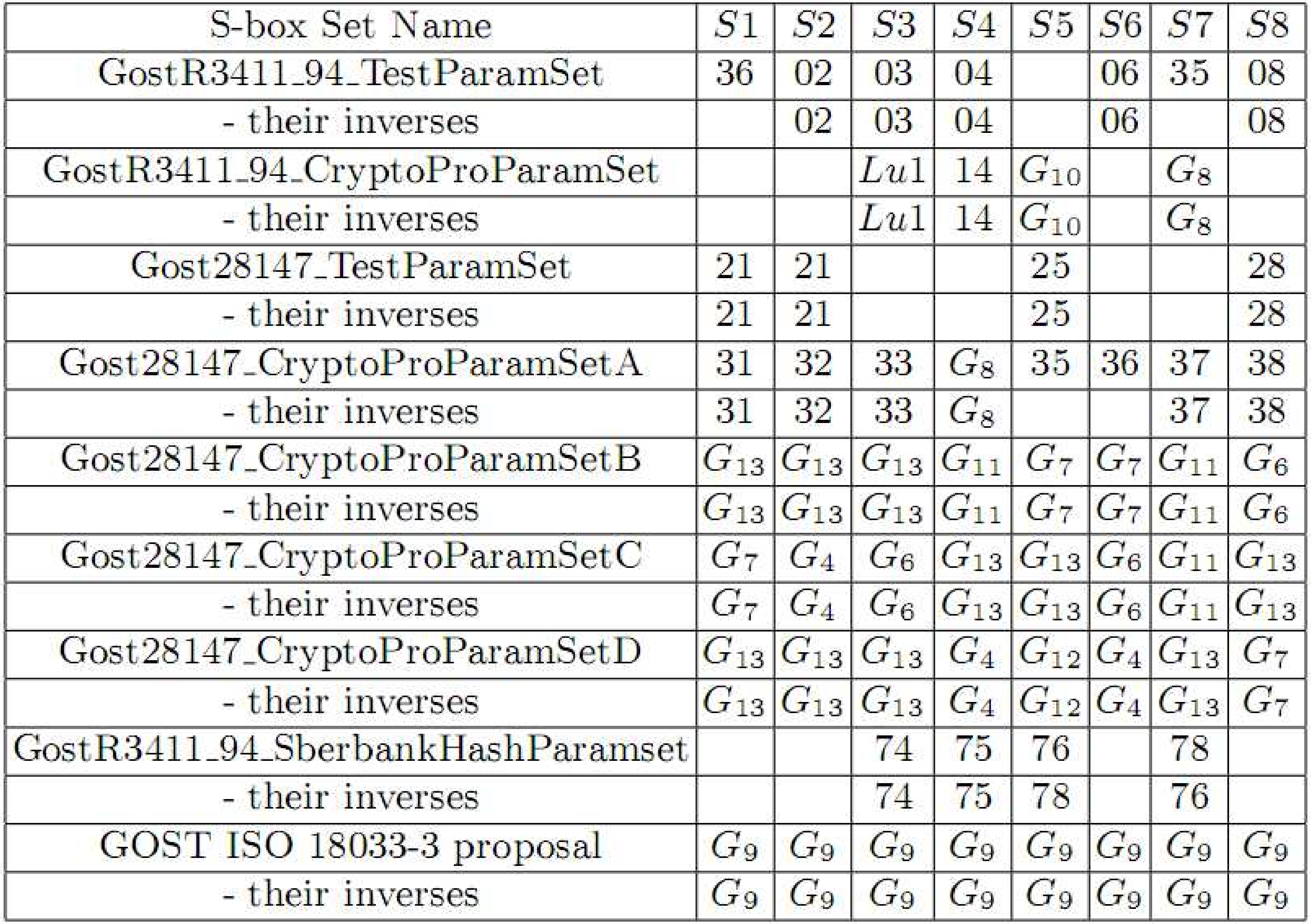
- Courtois Goubin Patarin, Eurocrypt 1998

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.

GOST 28148-89

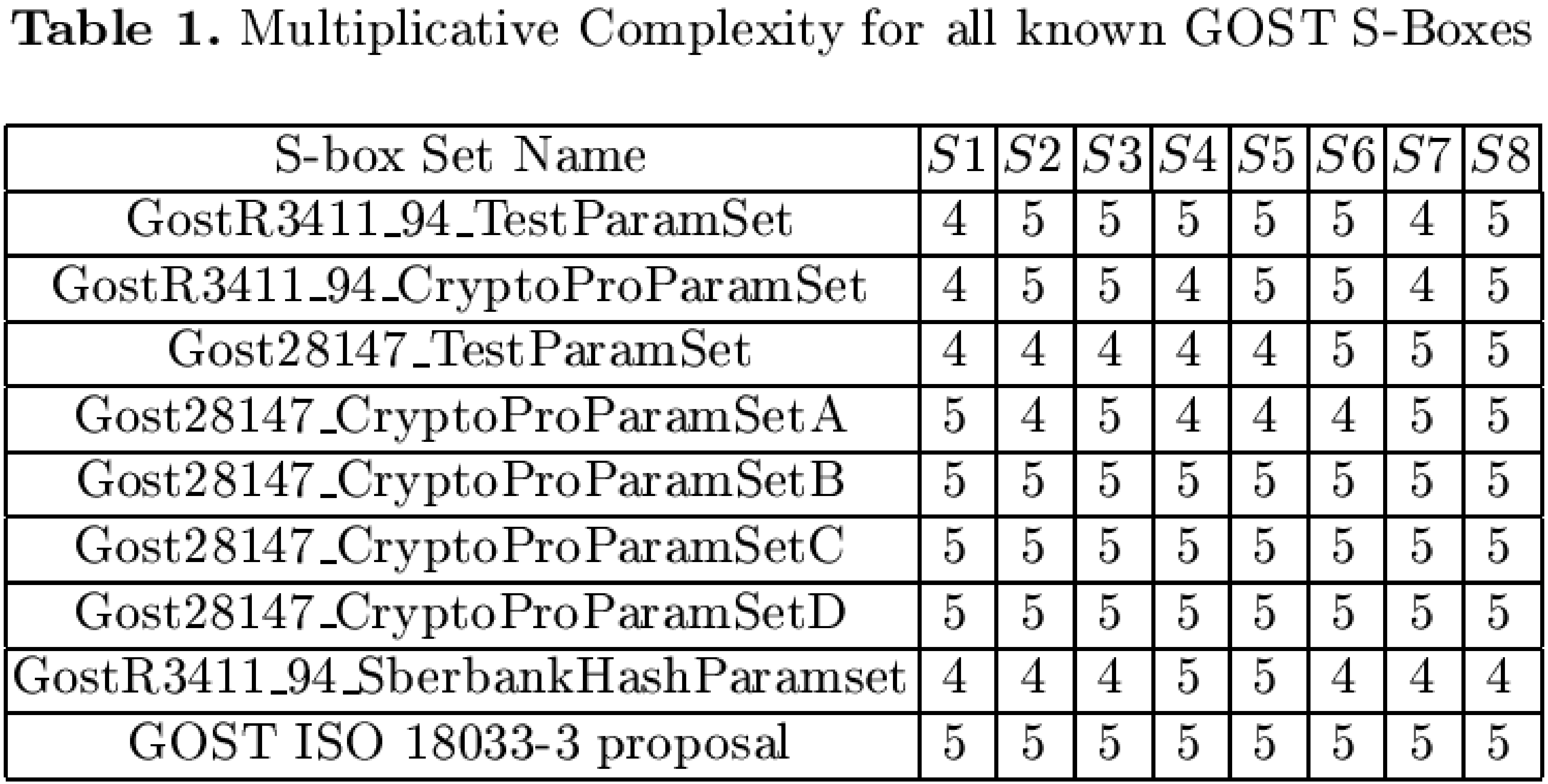


Table 1. Multiplicative Complexity for all known GOST S-Boxes

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.

11.
DC

COMP128v1
DES,
GOST

GOST vs. LC and DC

Bruce Schneier, Applied Cryptography, 1996, Section 14.1. page 334

“Against differential and linear cryptanalysis, GOST is probably stronger than DES”

Gabidulin 2000-2001:

For security = 2^{256}, 5 rounds are sufficient to protect GOST against linear cryptanalysis.

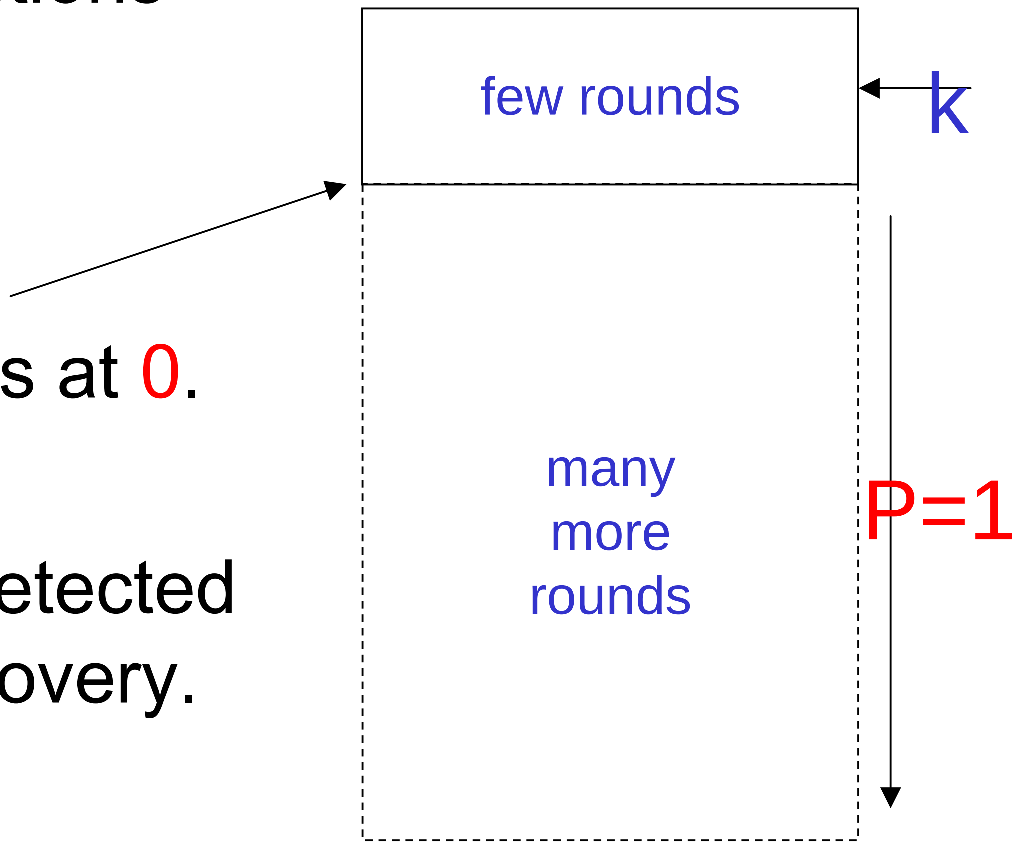
Moreover, even if the S-boxes are replaced by identity, and the only non-linear operation in the cipher is the addition modulo 2^{32}, the cipher is still secure against linear cryptanalysis after 6 rounds out of 32.

11.1.
The “Holy Grail” of DC

How To Reduce The Number Of Rounds

Attack on Keyed One-Way F == or Keyed Hash Functions == MACs.

Produce extinguishing differentials: All Δ bits at 0.



Each collision leads is detected and leads to key recovery.

Huge weakness.

COMP128v1 – Very Weak

Closed-source algorithm designed by the GSM association.

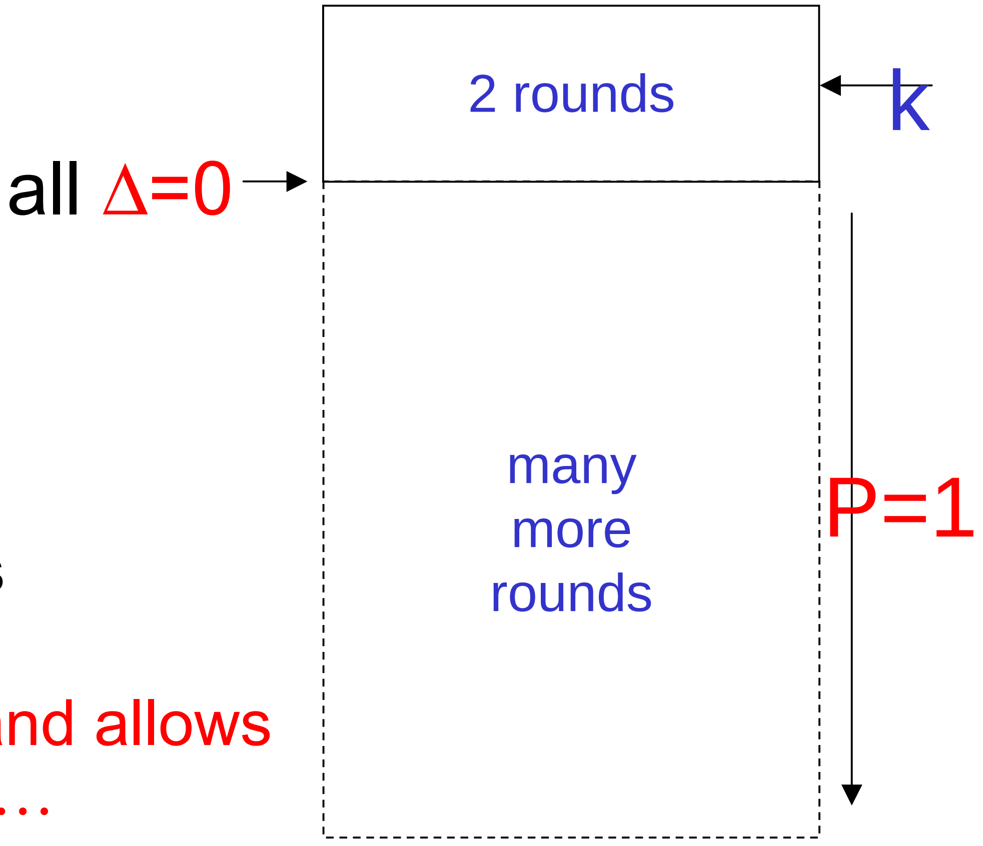
Kept secret until leaked and broken in 1997.

After it was BADLY broken,

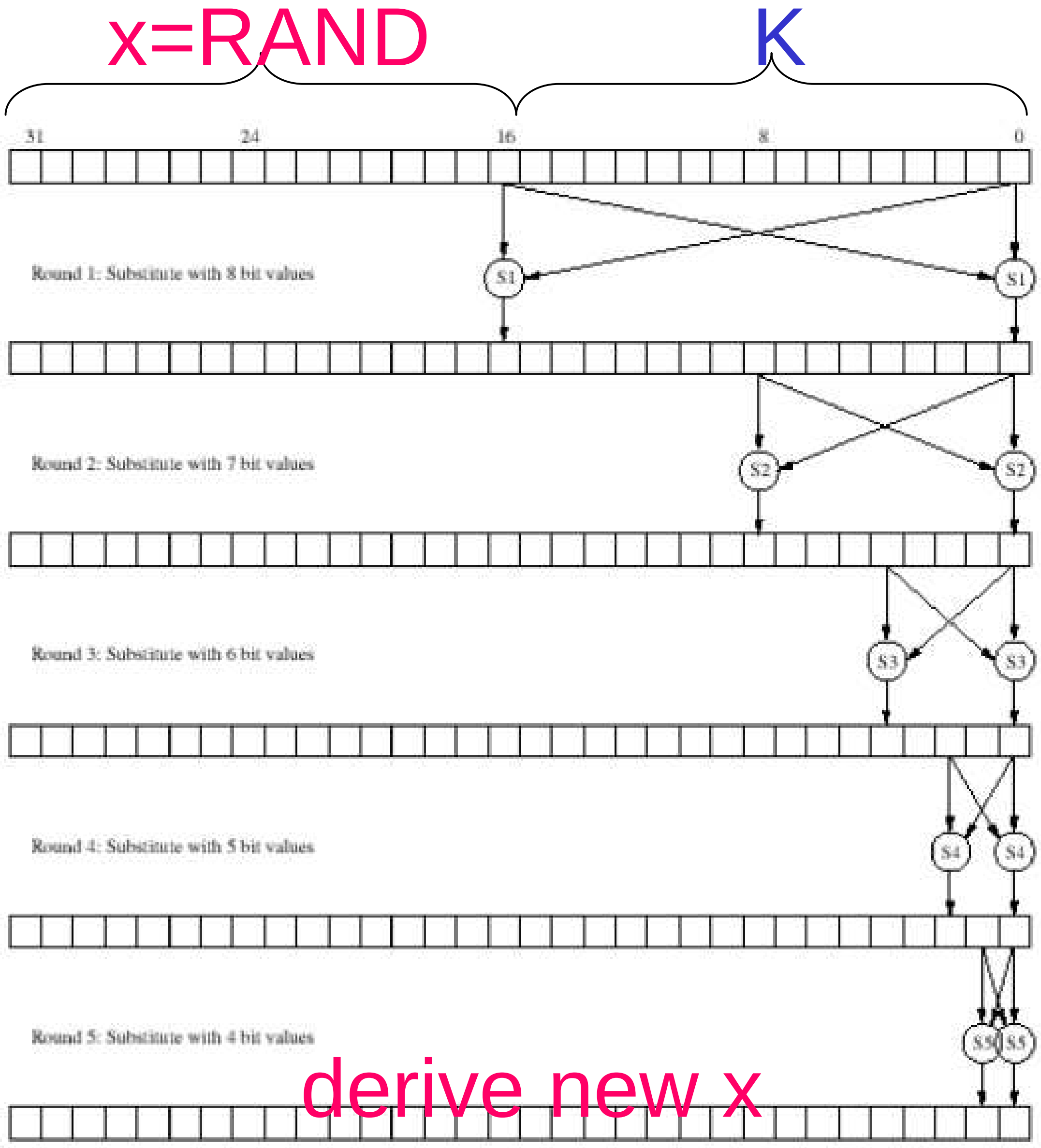
GSM Committee issued a statement saying it was just an example…

To this day the attack works and allows to clone many SIM cards…

We have extracted many keys…



COMP128v1 = Butterfly Algorithm, 8*5 rounds


x=RAND
K
31
24
16
8
0
Round 1: Substitute with 8 bit values
S1
S1
Round 2: Substitute with 7 bit values
S2
S2
Round 3: Substitute with 6 bit values
S3
S3
Round 4: Substitute with 5 bit values
S4
S4
Round 5: Substitute with 4 bit values
S5
S5
derive new x

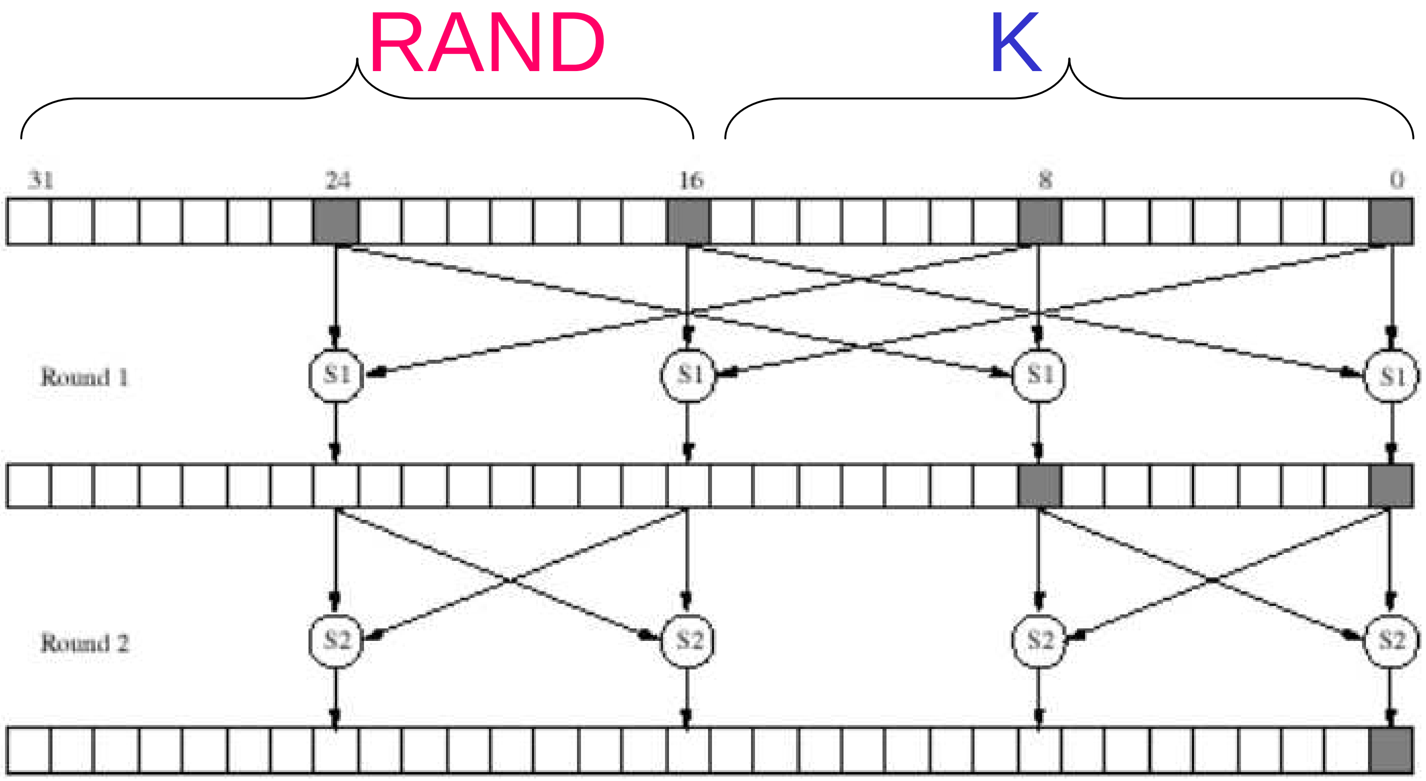
5-round

compression_K

F_K:128 -> 128

Weakness : “All Zero Output Difference”

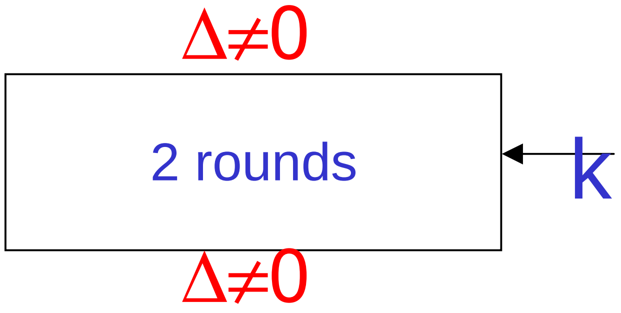
Collision for the first 2 rounds!   a.k.a. “Narrow Pipe”.



"All Zero Output Difference" for DES?

Impossible

for bijective functions.



The best we can hope:

reproduction

of small HW pattern Δ.

“All Zero Output Difference” for Round Functions

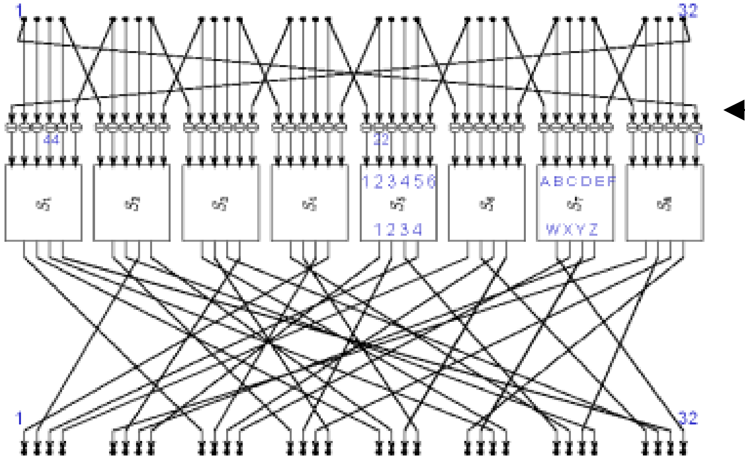
Possible for DES:

not bijective.

Not easy

(3 or more boxes).

Δ≠0

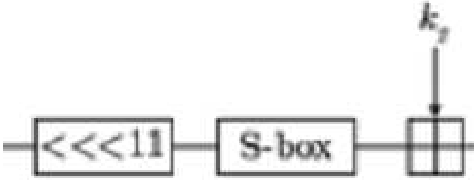


k

Δ=0

Impossible for GOST: bijective.

Δ≠0



Δ≠0

11.2. CPA =

Comparative Power Analysis

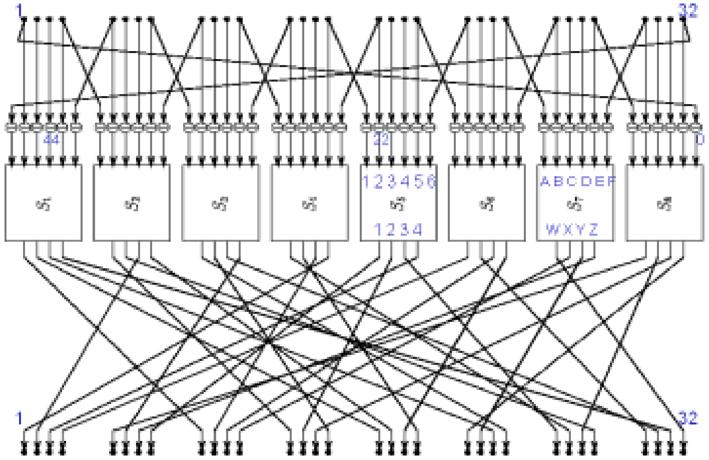
[Shamir et al. 2010]

N. Homma, A. Miyamoto, T. Aoki, A. Satoh, and A. Shamir: Comparative Power Analysis of Modular Exponentiation Algorithms, IEEE Transaction on Computers 59(6), pp. 795-807, 2010

“All Zero Output Difference” on 32 Bits

- ⇒ the Same trace
- ⇒ If deterministic…

Δ=0



k

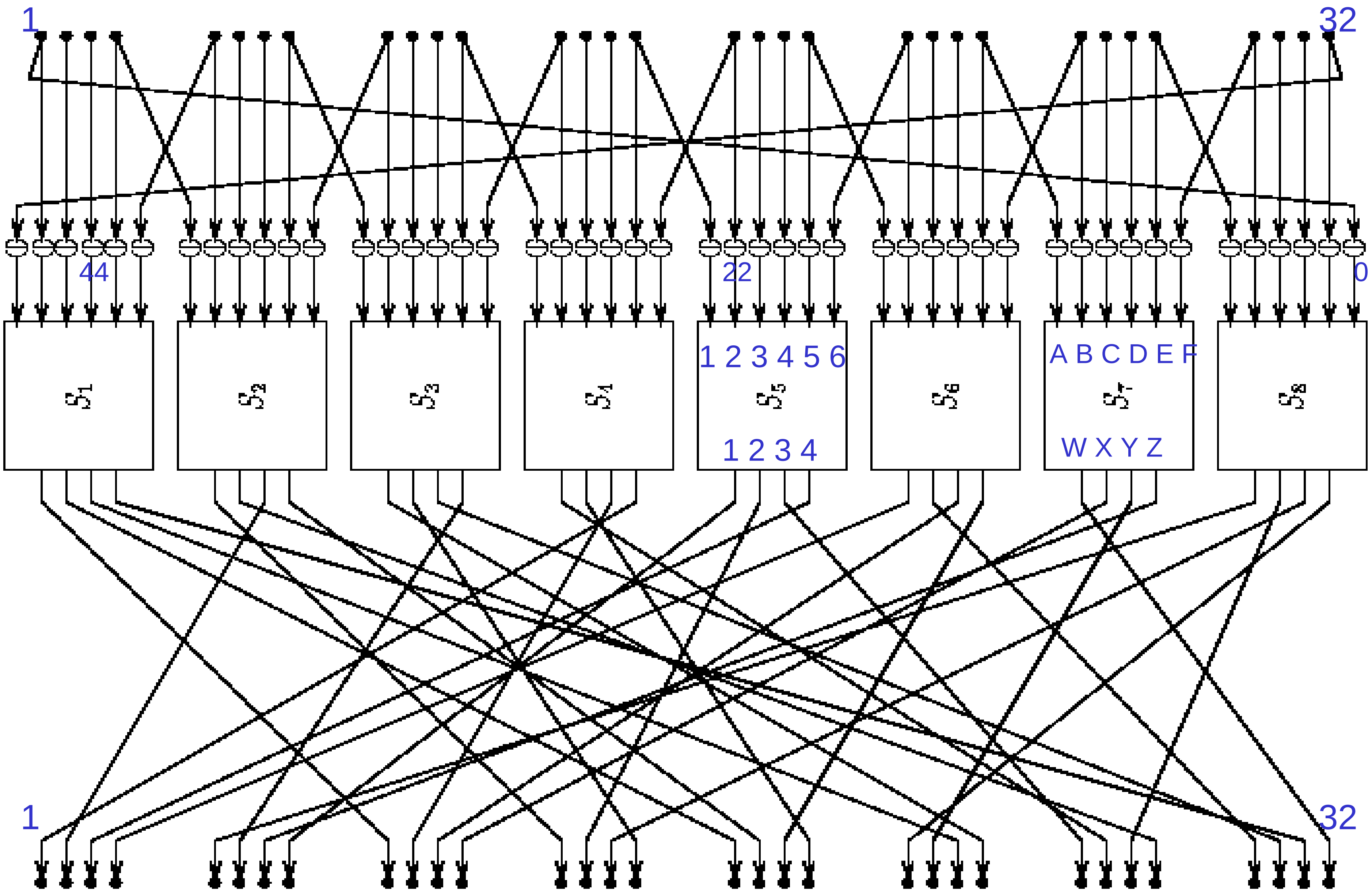
CPA =

Comparative Power Analysis: Extended def:

- Compare longer traces:
  - if identical, we have an “all-zero differential” (all the inputs must be the same).
  - Usually a CPA (better chances of success).

11.3. DC on DES

DES:



DC on DES [Biham-Shamir]

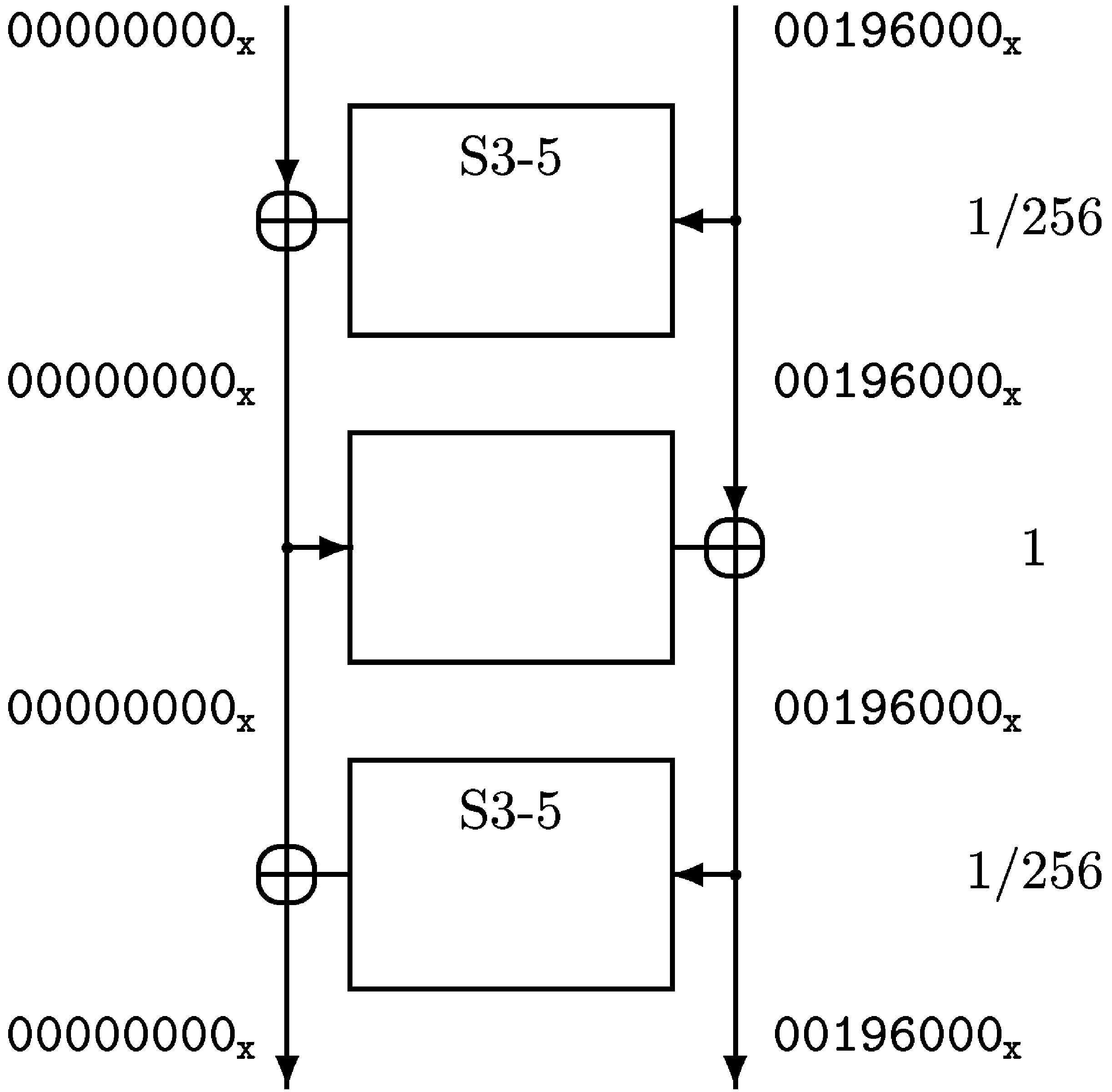


Figure 1: An example of Differential Cryptanalysis

11.4.
Classical DC
or How to Get Misled

DC Complexity

Simple “naïve” attack like Biham-Shamir attack on DES.

Assume “Differential Property of any kind”

Propagation $P = 2^{-X}$

Data Complexity = $1/P = 2^{X}$. Data can be obtained with different keys!!!!

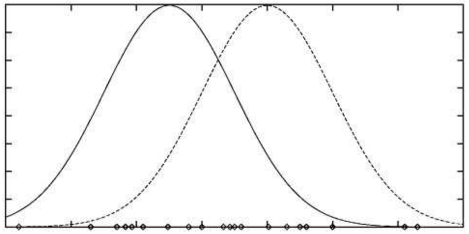
Time Complexity = $1/P = 2^{X}$.

This Assuming there is no “noise”.

Guess some key bits => observe an “exceptional” event

=> right key with high proba.

Advanced differential attacks: “signal” + “noise”.



Natural Event $P = 2^{-Y}$ for a RP.

Propagation: $P = 2^{-Y} + 2^{-X}$ for XXX rounds.

Distinguishing between two Gaussian distributions.

Q: How many standard deviations?

Right key with proba? <= Gauss error function.

StDev = $2^{-Y/2}$ => it is sufficient to obtain $2^{-X} = C * 2^{-Y/2}$!!!!! Complexity ≥ $O(2^{Y/2})$??

Biham-Shamir DC and GOST

If our model was DES…
we have totally misunderstood differential cryptanalysis.

Gabidulin 2000-2001:

Also claimed that 7 rounds are sufficient
to protect GOST against DC.

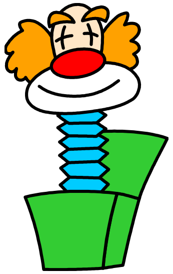
How To Be Led Astray

There are many papers about “provably security of ciphers” against DC and LC. Such works was published also about GOST, even in 2010…

⇒ In fact it is possible to CHEAT someone and to make them believe that GOST is provably secure against DC…

⇒ While in reality GOST in insecure against DC!

How interesting…



2 Rounds Further?

The most recent paper about this topic:

Martin Albrecht and Gregor Leander:

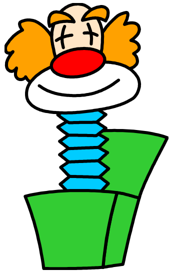
An All-In-One Approach to Differential Cryptanalysis for Small Block Ciphers, Preprint, eprint.iacr.org/2012/401.

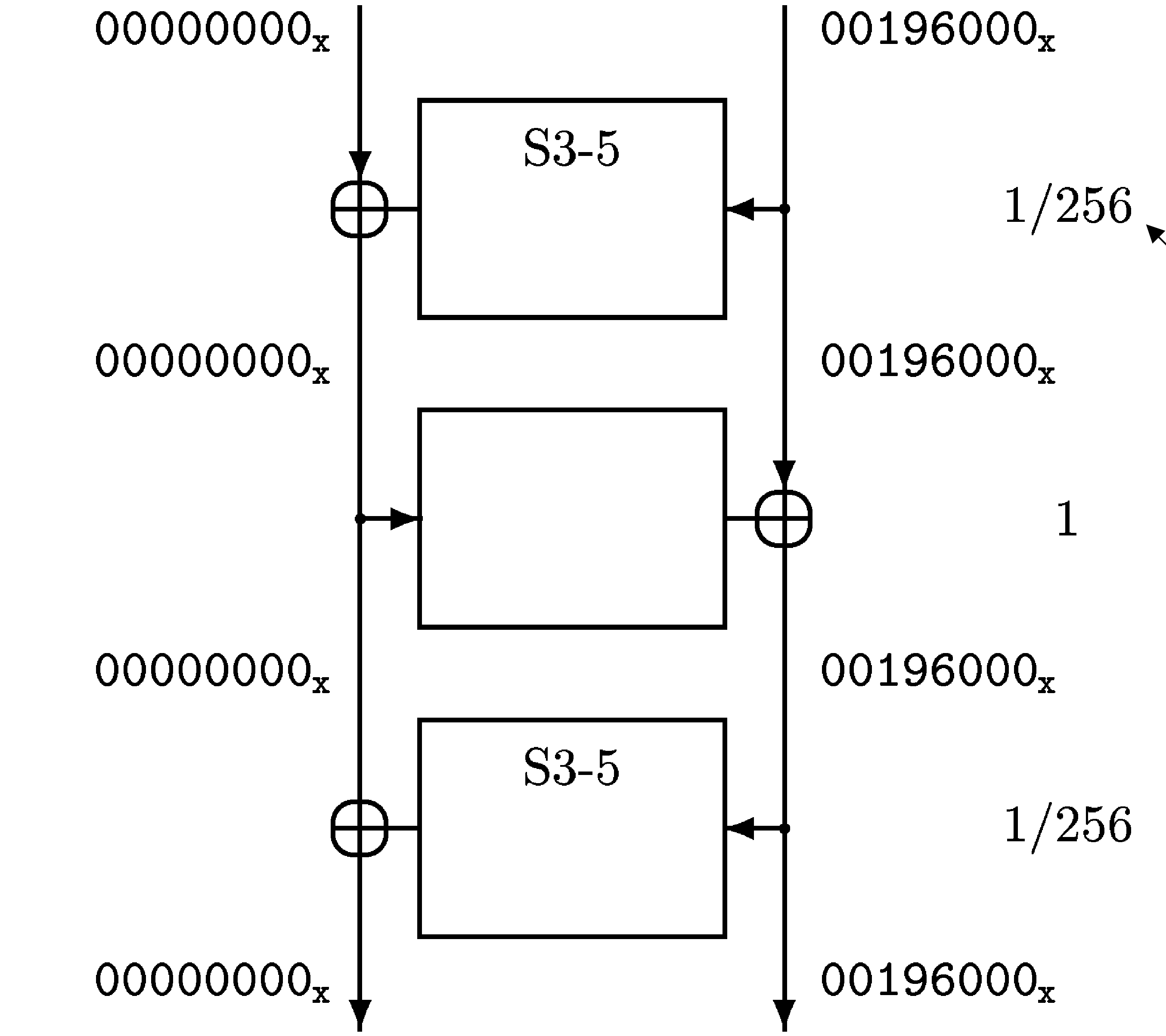
In Section 1.1. page 3:

“ *Truncated differentials, first mentioned in [15] can be seen as a collection of differentials and in some cases allow to push differential attacks one or two rounds further…* “

NOT QUITE …

⇒ For Russian GOST they allowed us to push the attack more than 20 rounds further!





DES:

Quasi constant probability, or 2 cases…

Figure 1: An example of Differential Cryptanalysis

GOST vs. DES

DES: quasi constant probability. Does not become zero typically.

GOST, general case: propagation probability depends on the key.
Can be zero.

The problem:

For some keys it will be 0.
With probabilities as high as ½ or similar.

If for some keys it is 0,

then however strong it can sometimes be…
it is guaranteed to be 0 after a few rounds(!)

(assuming independent round keys…)

Our early estimation: a single differential attack on GOST would propagate with probability not better than 2^{-62} for 32 rounds.

For most keys it would propagate with probability 0.

11.5.
DC With Sets

More Differential Cryptanalysis

[Seki, Kaneko SAC 2000]:

Sets of differentials = most general

Incomplete/truncated Differentials = With free bits…

Between 12 and 17 rounds out of 32 can be broken…

No attack beyond.

Or it is not clear how one would proceed: signal>noise…

Sets Of Differentials [Seki-Kaneko,Courtois-Misztal]

$$A \rightarrow B$$

any non-zero $a \in A$, any non-zero $b \in B$

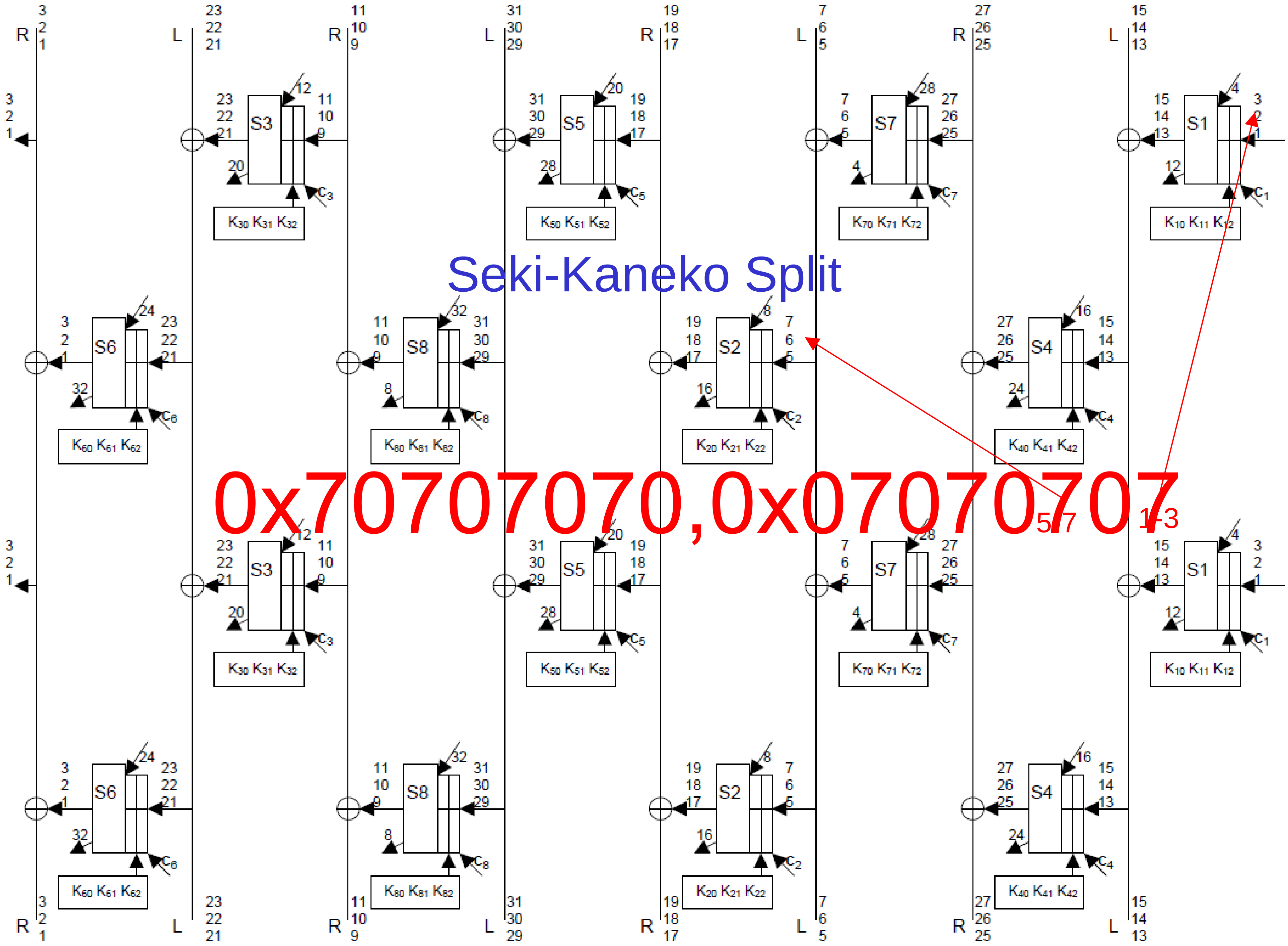
In this 64-bit string:

0x70707070,0x07070707

one half can be 0,
the whole must be non-zero

$2^{24}-1$ differences

24 active bits



Seki-Kaneko Split

0x70707070,0x07070707

Seki-Kaneko Set

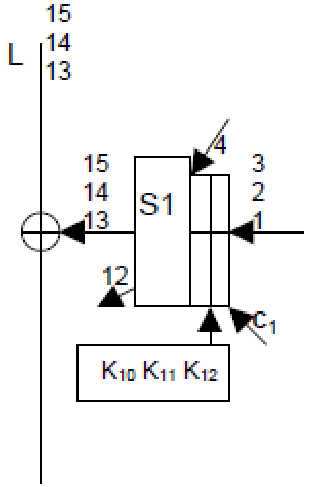
3 bits active per every second box.

S1357 in odd rounds 1,3,…

S2468 in even rounds 2,4,…

Rough estimation: there are only 4 bits coming “out” in each round. These differences must be 0 “by accident”.

Maybe 0x70707070,0x07070707 propagates with probability 2^{-4} per round?

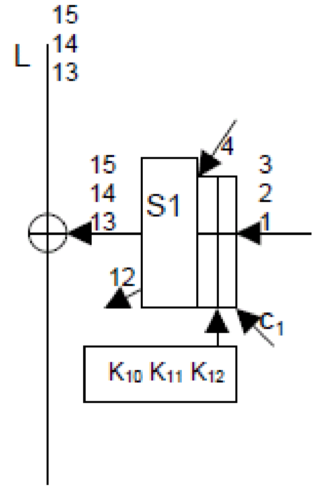


Seki-Kaneko Set (contd.)

4 bits coming “out” in each round.
these differences must be 0 “by accident”.

So 0x70707070,0x07070707 propagates with probability 2^{-4} per round?

Not quite. There are also carries: on picture bits 123 active, 4 always inactive, S2 will be active with proba about $1\text{-}3.5/16 = 2^{-0.36}$.



So we expect $2^{-4-3.5*0.36} = 2^{-5.3}$.

Simulations also give $2^{-5.3}$ average

(odd vs. even rounds, for the S-boxes of Central Bank of Russia)

Seki-Kaneko

Is 0x70707070,0x07070707 dangerous?

Probability $2^{-5.3}$ for 1 round.

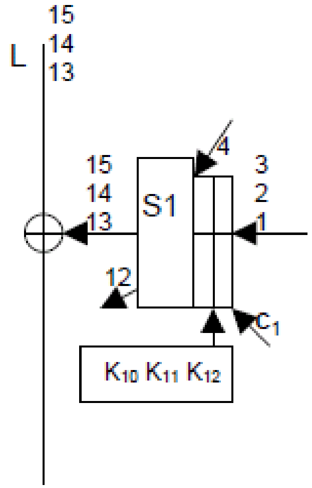
Means 2^{-170} for 32 rounds.

No hope to break GOST so far.

There is only $2^{64+24-1} = 2^{87}$

pairs with input difference

$\in$ 0x70707070,0x07070707.



Very Surprising

Propagation is MUCH better than expected. Already true for this old Japanese set from 2000.

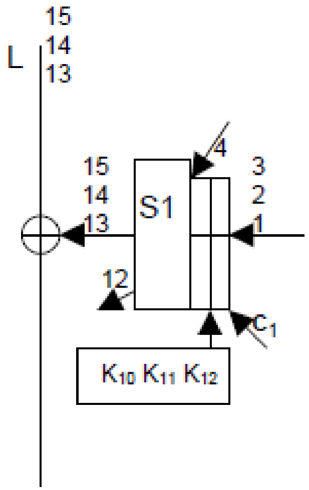
0x70707070,0x07070707.

Strong improvement. Examples:

2 Rounds: predicted $2^{-10.6}$ actual $2^{-8.6}$.

4 Rounds: predicted $2^{-21.2}$ actual $2^{-16.7}$.

8 Rounds: predicted $2^{-42.4}$ actual $2^{-28.4}$.



11.6.
Better Sets [2011]

New Sets [Courtois-Misztal, 2011]

References:

1. Nicolas Courtois, Michał Misztal:
Aggregated Differentials and Cryptanalysis of PP-1 and GOST,
In CECC 2011, 11th Central European Conference on Cryptology,
Budapest 2011, post-proceedings in preparation.
=> invention of new sets

2. Nicolas Courtois, Michał Misztal:
First Differential Attack On Full 32-Round GOST, In ICICS'11, Beijing, China,
pp. 216-227, Springer LNCS 7043, 2011.
=> first simple attack (very slightly) faster than brute force $2^{254.6}$

3. Nicolas Courtois, Michał Misztal:
Differential Cryptanalysis of GOST,
Preprint, 14 June 2011 eprint.iacr.org/2011/312.
=> progressive improved approach, heuristic and not very precise… 2^{226}

4. Nicolas Courtois:
An Improved Differential Attack on Full GOST,
Preprint Archive, 15 March 2012, eprint.iacr.org/2012/138.
=> symmetric + many further refinements + very careful work on individual bits + tight [barely working] distinguishers + justification of earlier results 2^{179}

New vs. Old Sets

- Seki-Kaneko:

  0x70707070,0x07070707

  2^{24}-1 differences

  24 active bits

  naturally occurs: 2^{-40}

- Courtois-Misztal

  0x80700700,0x80700700

  2^{14}-1 differences

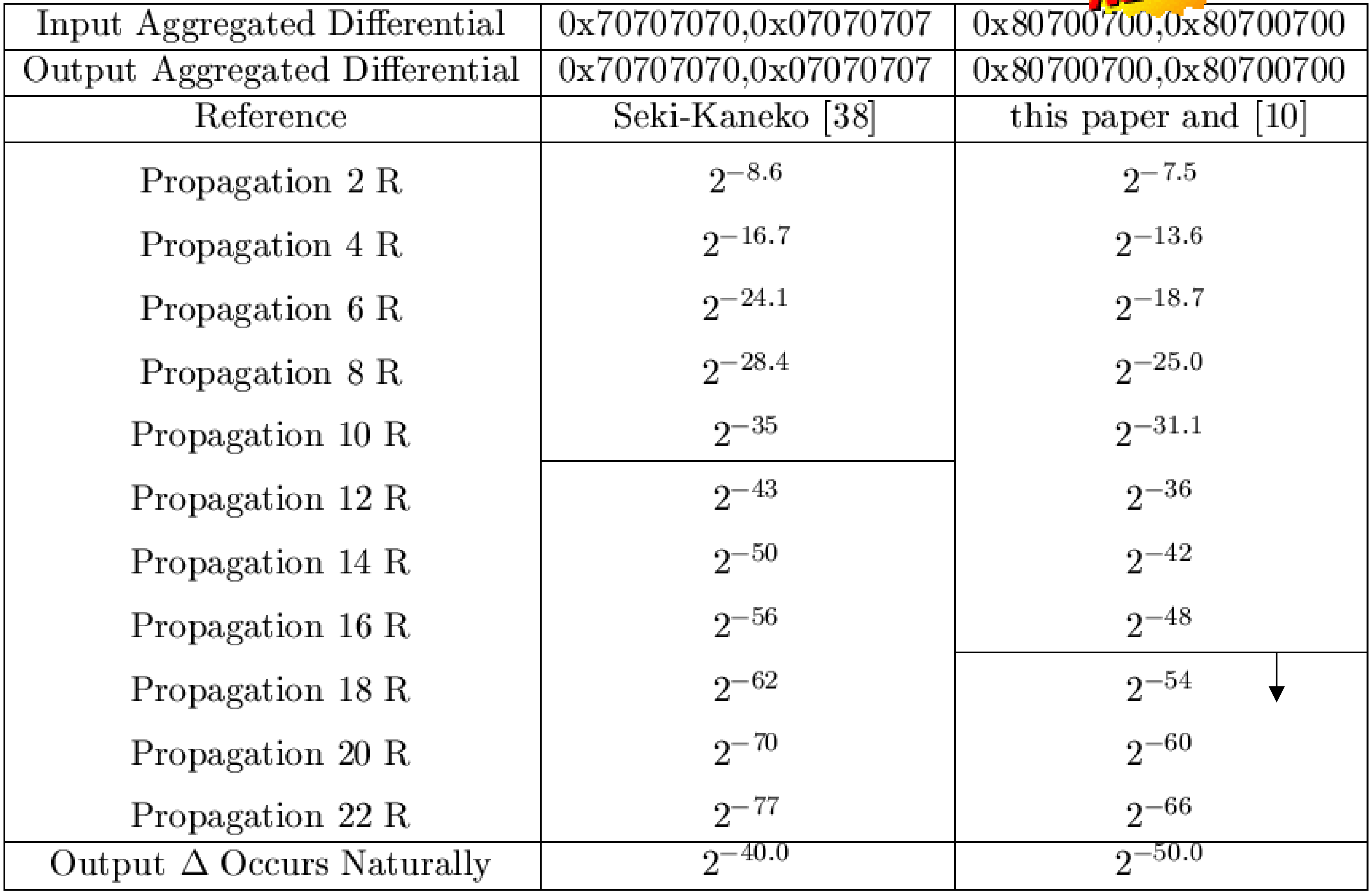
  14 active bits

  naturally occurs: 2^{-50}

simultaneously bigger signal and smaller noise

New Sets [Courtois,Misztal, 2011]

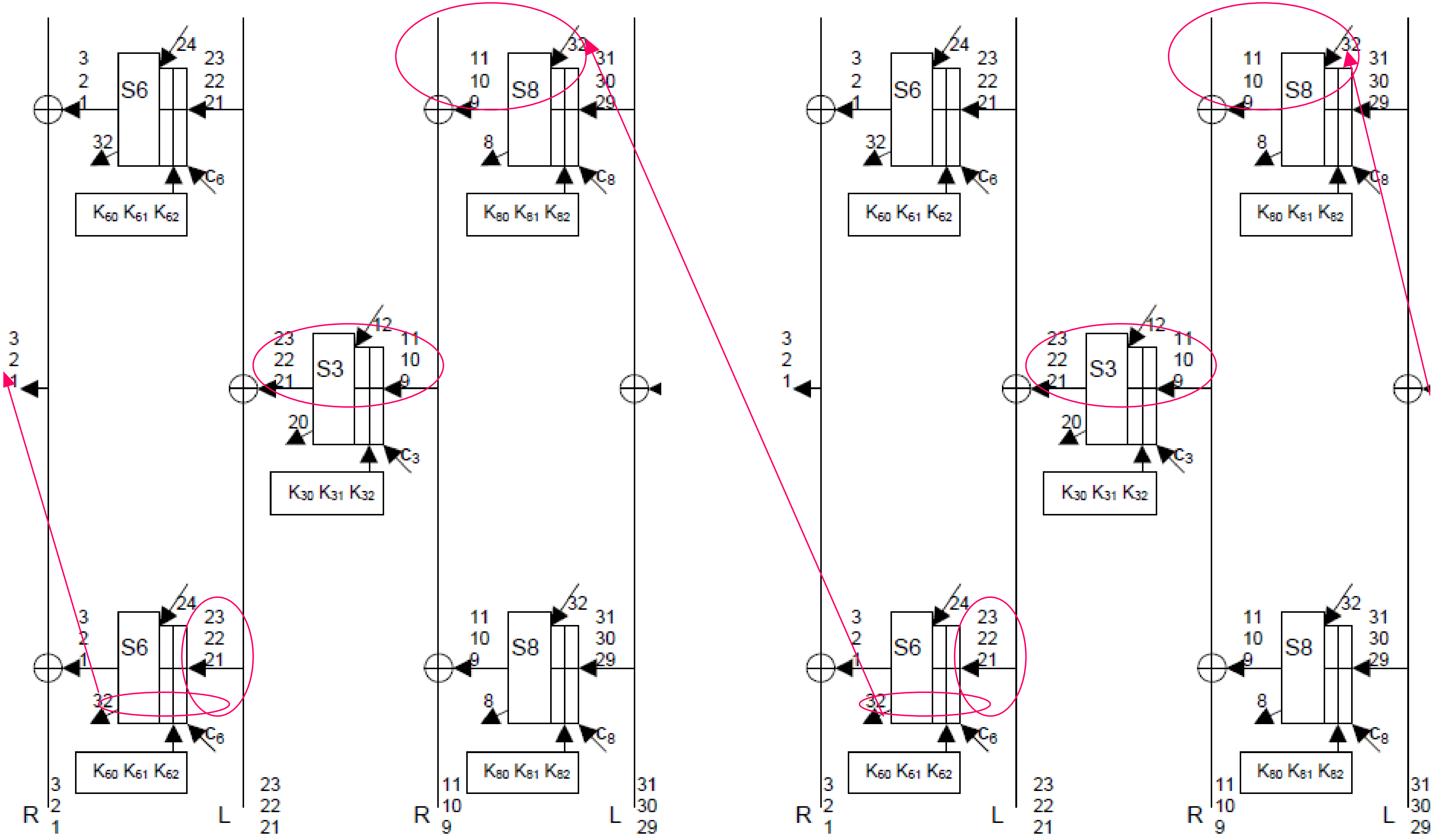
NEW!



Input Aggregated Differential	0x70707070,0x07070707	0x80700700,0x80700700
Output Aggregated Differential	0x70707070,0x07070707	0x80700700,0x80700700
Reference	Seki-Kaneko [38]	this paper and [10]
Propagation 2 R	$2^{-8.6}$	$2^{-7.5}$
Propagation 4 R	$2^{-16.7}$	$2^{-13.6}$
Propagation 6 R	$2^{-24.1}$	$2^{-18.7}$
Propagation 8 R	$2^{-28.4}$	$2^{-25.0}$
Propagation 10 R	2^{-35}	$2^{-31.1}$
Propagation 12 R	2^{-43}	2^{-36}
Propagation 14 R	2^{-50}	2^{-42}
Propagation 16 R	2^{-56}	2^{-48}
Propagation 18 R	2^{-62}	2^{-54}
Propagation 20 R	2^{-70}	2^{-60}
Propagation 22 R	2^{-77}	2^{-66}
Output Δ Occurs Naturally	$2^{-40.0}$	$2^{-50.0}$

0x80700700,0x80700700

Type 3+3: S836 + S836

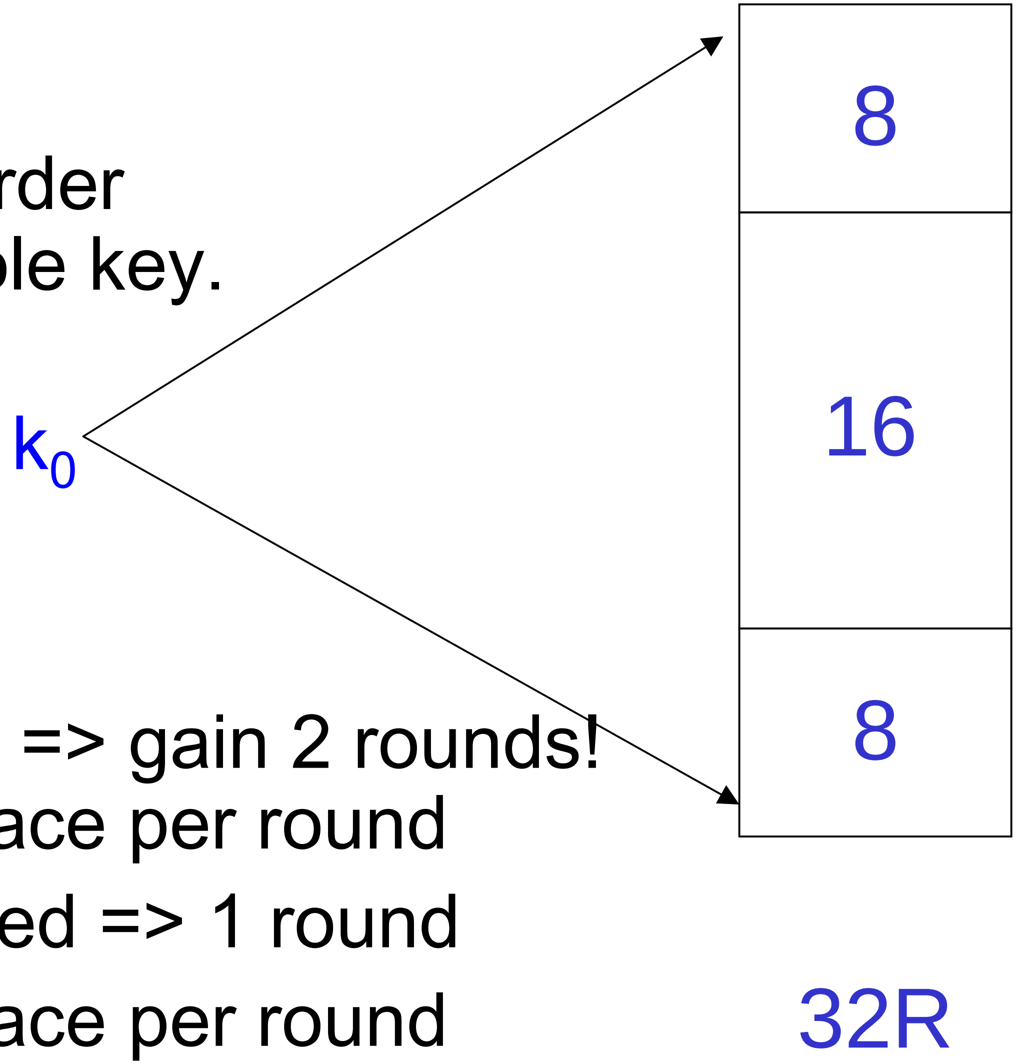

S6
S8
S3
K_{60} K_{61} K_{62}
K_{80} K_{81} K_{82}
K_{30} K_{31} K_{32}
C_6
C_8
C_3
R
L
3
2
1
32
24
23
22
21
11
10
9
8
31
30
29
12
20

11.7. Refined Attacks

Key Scheduling

Essential Weakness:

Same Keys Inversed Order
+ small size << whole key.



GOST: 32 bits guessed => gain 2 rounds!
- 0.06 of the key space per round

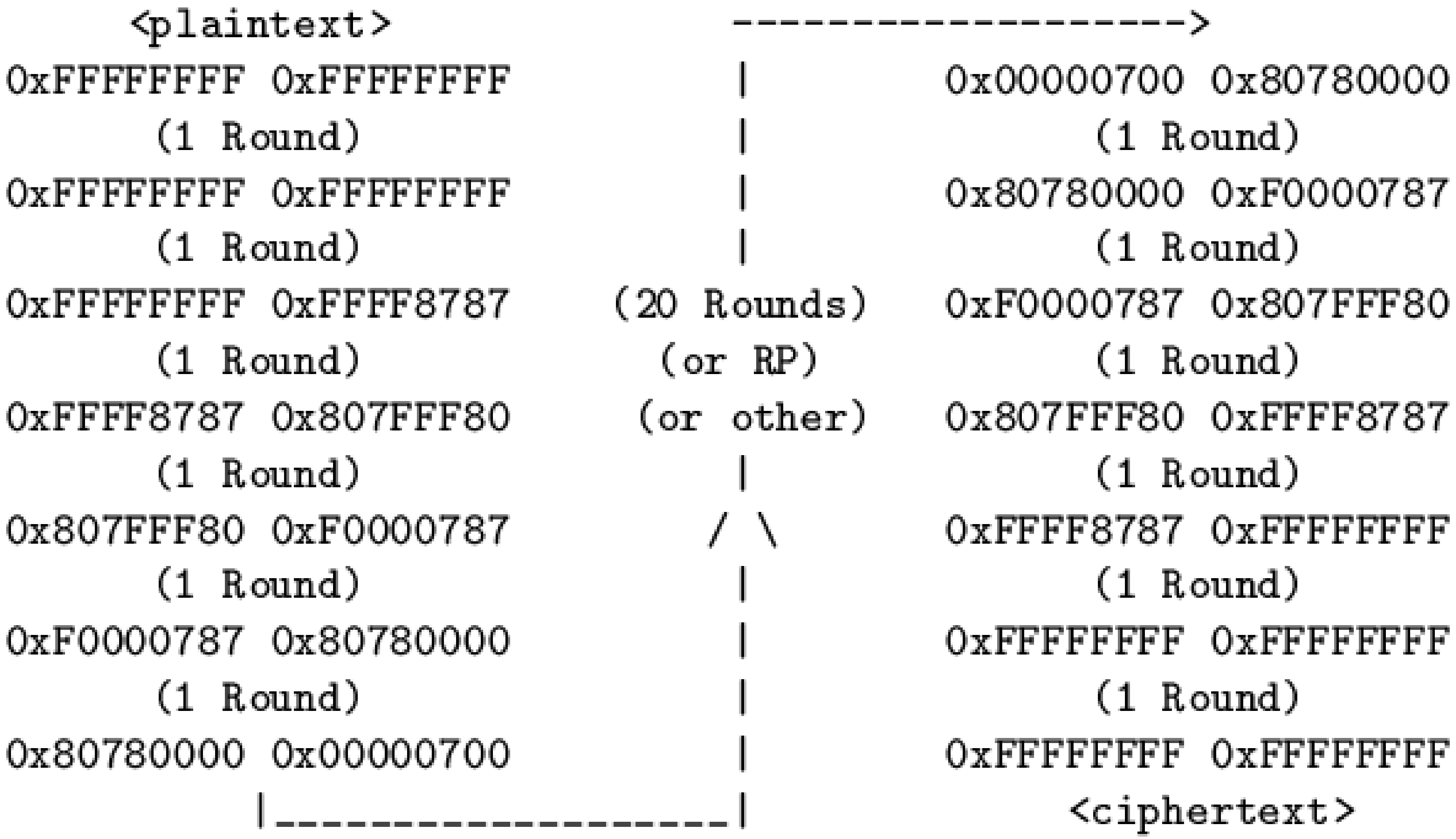
DES: 48 key bits guessed => 1 round
- 0.86 of the key space per round

New Attacks

References:

1. Nicolas Courtois, Michał Misztal:
   Aggregated Differentials and Cryptanalysis of PP-1 and GOST,
   In CECC 2011, 11th Central European Conference on Cryptology,
   Budapest 2011, post-proceedings in preparation.
   => invention of new sets
2. Nicolas Courtois, Michał Misztal:
   First Differential Attack On Full 32-Round GOST, In ICICS'11, Beijing, China,
   pp. 216-227, Springer LNCS 7043, 2011.
   => first simple attack (very slightly) faster than brute force $2^{254.6}$
3. Nicolas Courtois, Michał Misztal:
   Differential Cryptanalysis of GOST,
   Preprint, 14 June 2011 eprint.iacr.org/2011/312.
   => progressive improved approach, heuristic and not very precise… 2^{226}
4. Nicolas Courtois:
   An Improved Differential Attack on Full GOST,
   Preprint Archive, 15 March 2012, eprint.iacr.org/2012/138.
   => symmetric + many further refinements + very careful work on individual bits + tight [barely working] distinguishers + justification of earlier results 2^{179}

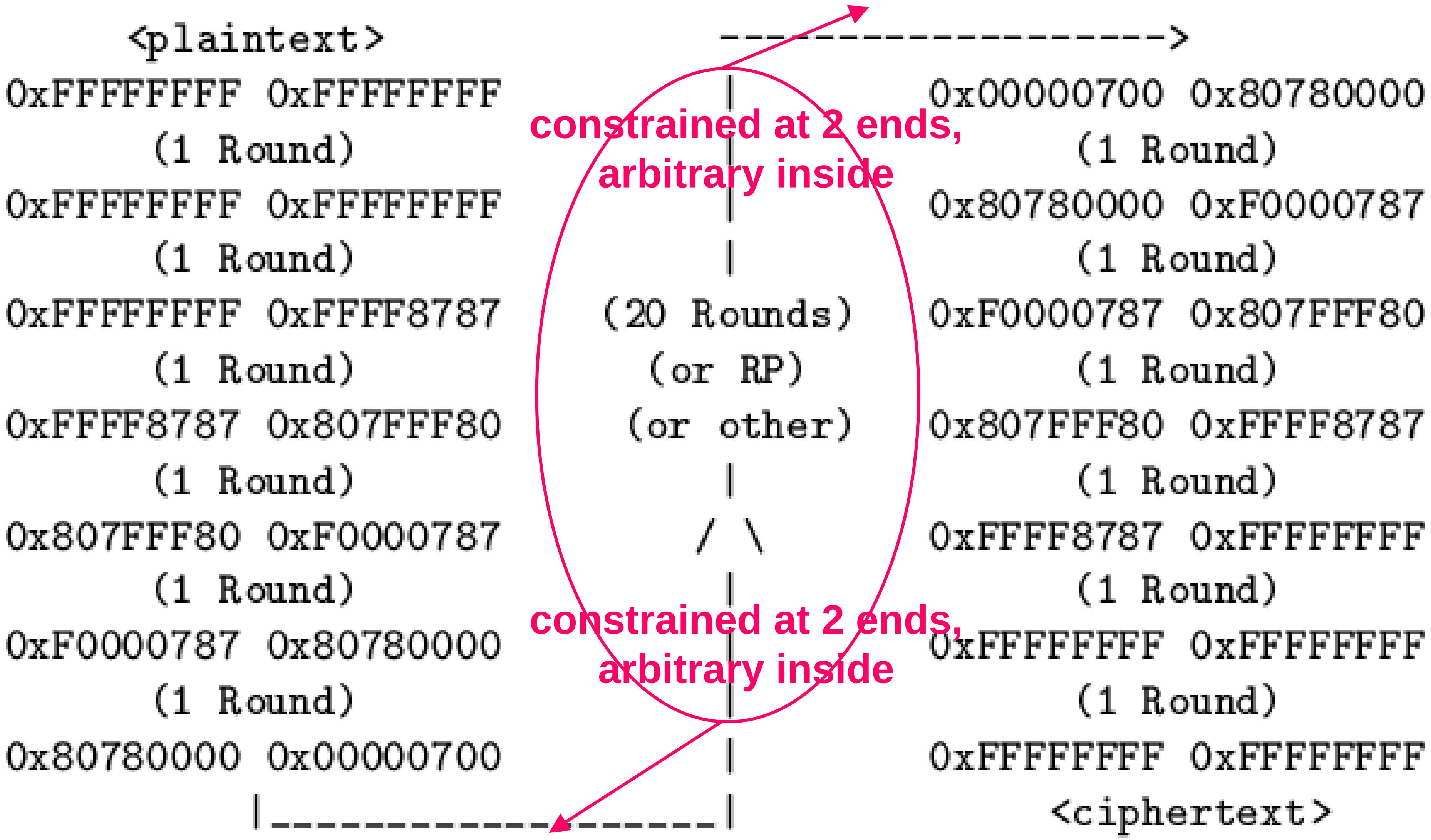
Refined Attacks [March 2012] - Symmetric



```
     <plaintext>                    -------------------->
0xFFFFFFFF 0xFFFFFFFF                |        0x00000700 0x80780000
      (1 Round)                      |              (1 Round)
0xFFFFFFFF 0xFFFFFFFF                |        0x80780000 0xF0000787
      (1 Round)                      |              (1 Round)
0xFFFFFFFF 0xFFFF8787           (20 Rounds)   0xF0000787 0x807FFF80
      (1 Round)                   (or RP)           (1 Round)
0xFFFF8787 0x807FFF80            (or other)   0x807FFF80 0xFFFF8787
      (1 Round)                      |              (1 Round)
0x807FFF80 0xF0000787               / \       0xFFFF8787 0xFFFFFFFF
      (1 Round)                      |              (1 Round)
0xF0000787 0x80780000                |        0xFFFFFFFF 0xFFFFFFFF
      (1 Round)                      |              (1 Round)
0x80780000 0x00000700                |        0xFFFFFFFF 0xFFFFFFFF
           |-------------------------|             <ciphertext>
```

Figure 6: The Alpha Property

Key Principles



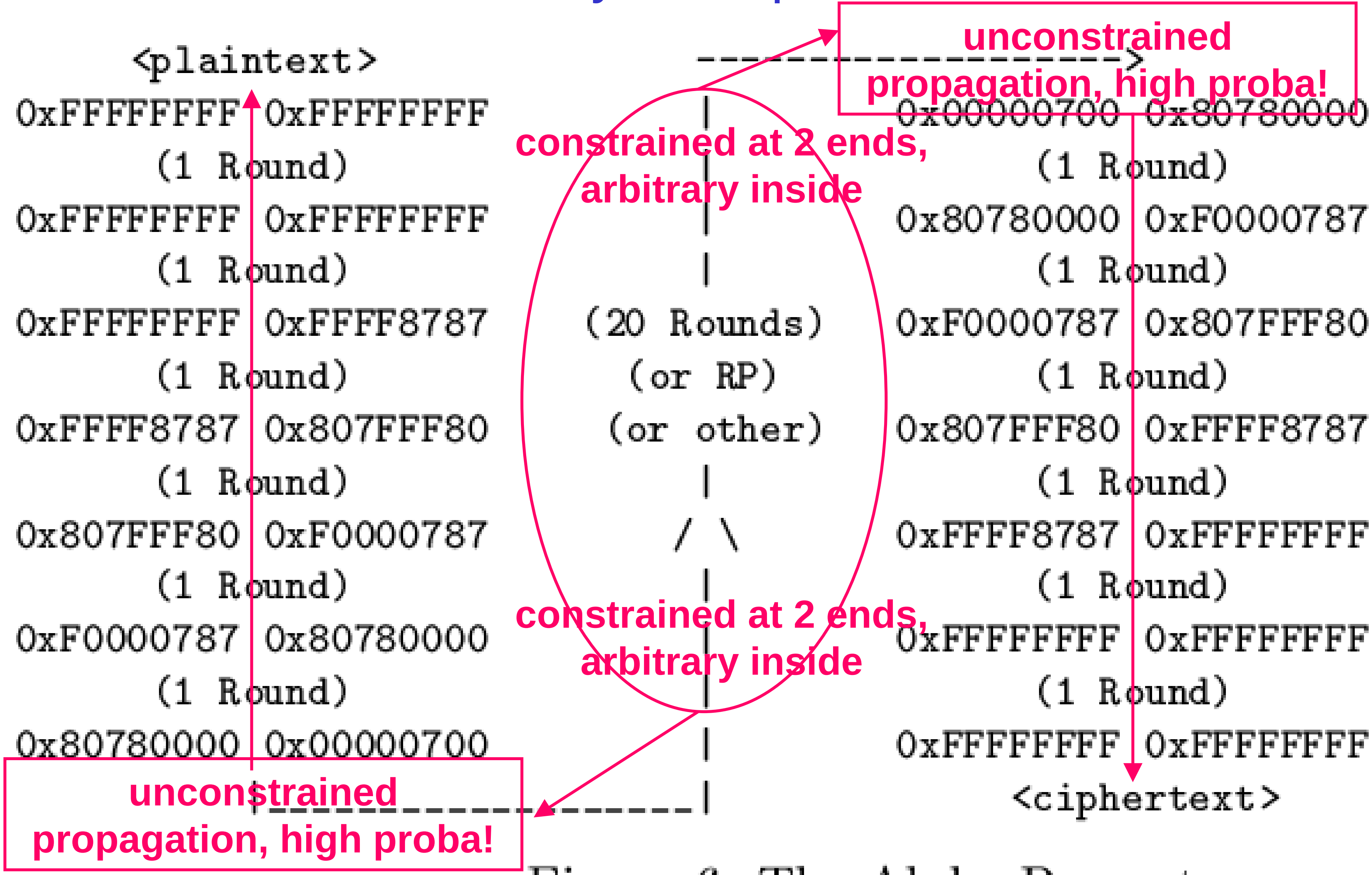
```
     <plaintext>                      --------------------->
0xFFFFFFFF 0xFFFFFFFF          |         0x00000700 0x80780000
     (1 Round)                 |              (1 Round)
0xFFFFFFFF 0xFFFFFFFF          |         0x80780000 0xF0000787
     (1 Round)                 |              (1 Round)
0xFFFFFFFF 0xFFFF8787     (20 Rounds)    0xF0000787 0x807FFF80
     (1 Round)              (or RP)           (1 Round)
0xFFFF8787 0x807FFF80     (or other)     0x807FFF80 0xFFFF8787
     (1 Round)                 |              (1 Round)
0x807FFF80 0xF0000787         / \        0xFFFF8787 0xFFFFFFFF
     (1 Round)                 |              (1 Round)
0xF0000787 0x80780000          |         0xFFFFFFFF 0xFFFFFFFF
     (1 Round)                 |              (1 Round)
0x80780000 0x00000700          |         0xFFFFFFFF 0xFFFFFFFF
          |--------------------|             <ciphertext>
```

constrained at 2 ends,
arbitrary inside

constrained at 2 ends,
arbitrary inside

Figure 6: The Alpha Property

Key Principles



```
          <plaintext>                           ------------------>
0xFFFFFFFF 0xFFFFFFFF             |             0x00000700 0x80780000
       (1 Round)                  |                    (1 Round)
0xFFFFFFFF 0xFFFFFFFF             |             0x80780000 0xF0000787
       (1 Round)                  |                    (1 Round)
0xFFFFFFFF 0xFFFF8787        (20 Rounds)        0xF0000787 0x807FFF80
       (1 Round)               (or RP)                 (1 Round)
0xFFFF8787 0x807FFF80         (or other)        0x807FFF80 0xFFFF8787
       (1 Round)                  |                    (1 Round)
0x807FFF80 0xF0000787            / \            0xFFFF8787 0xFFFFFFFF
       (1 Round)                  |                    (1 Round)
0xF0000787 0x80780000             |             0xFFFFFFFF 0xFFFFFFFF
       (1 Round)                  |                    (1 Round)
0x80780000 0x00000700             |             0xFFFFFFFF 0xFFFFFFFF
          |-----------------------|                  <ciphertext>
```

unconstrained propagation, high proba!
constrained at 2 ends, arbitrary inside
constrained at 2 ends, arbitrary inside
unconstrained propagation, high proba!

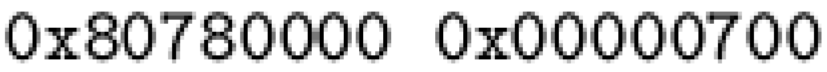
Figure 6: The Alpha Property

11.8.
Best Symmetric Result for 20 R (best known)

Propagation - Middle 20 Rounds

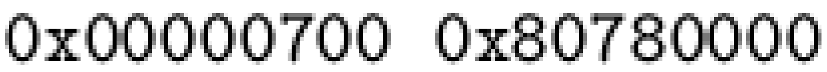
Propagation with probability???

What is Propagation???



0x80780000 0x00000700

20 rounds



0x00000700 0x80780000

What is Propagation? - 20 R

For 6 middle rounds:

We have 14 active bits, 2^{14}-1 differences.

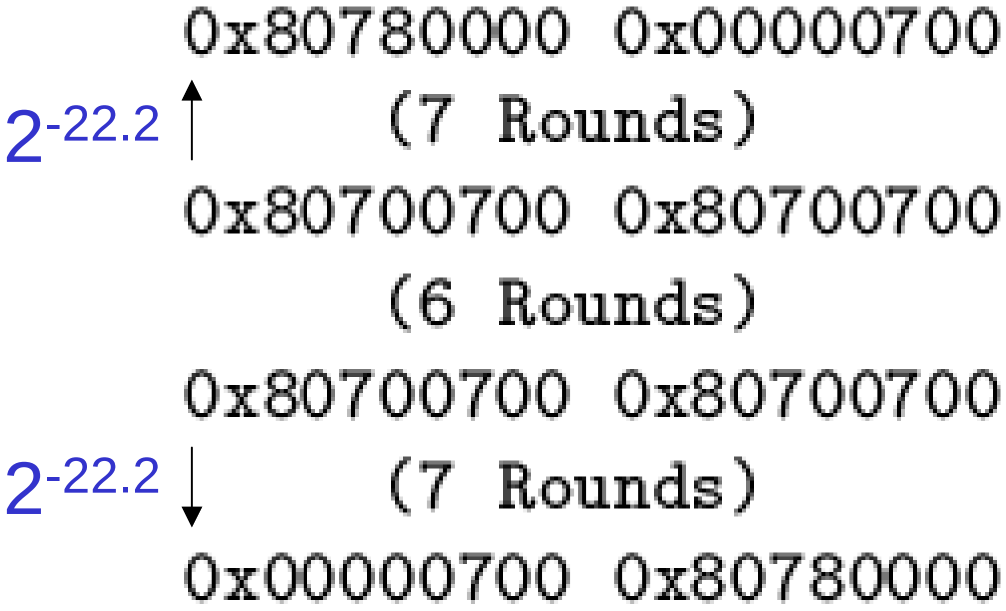
There are $2^{64+14-1} = 2^{77}$ input differences.

Propagation with probability $2^{-18.7}$ (experimental).

There are $2^{77-18.7} = 2^{58.3}$ pairs for the 6 middle rounds.

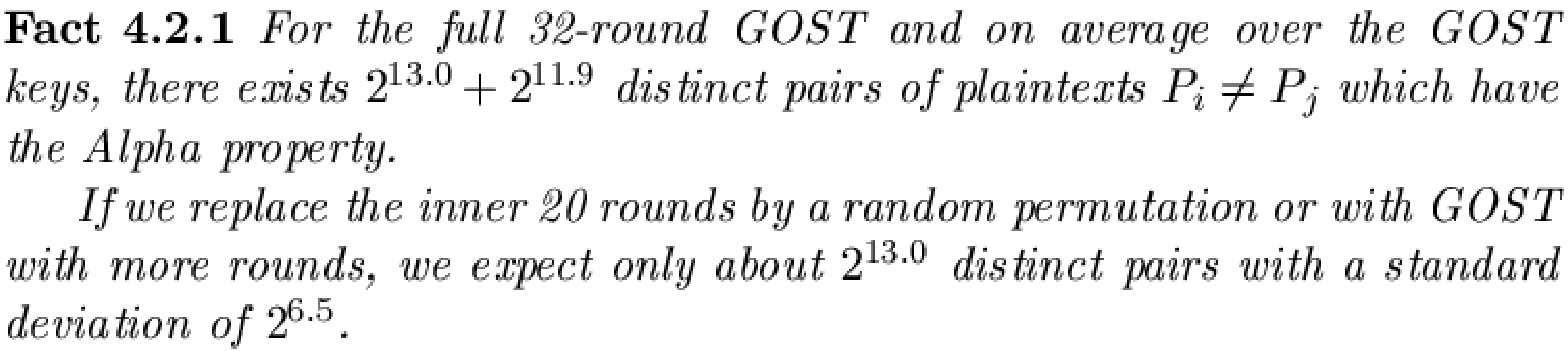
Result: $2^{58.3-22.2-22.2}$
$= 2^{13.9}$ cases.

Natural: 2^{15} .



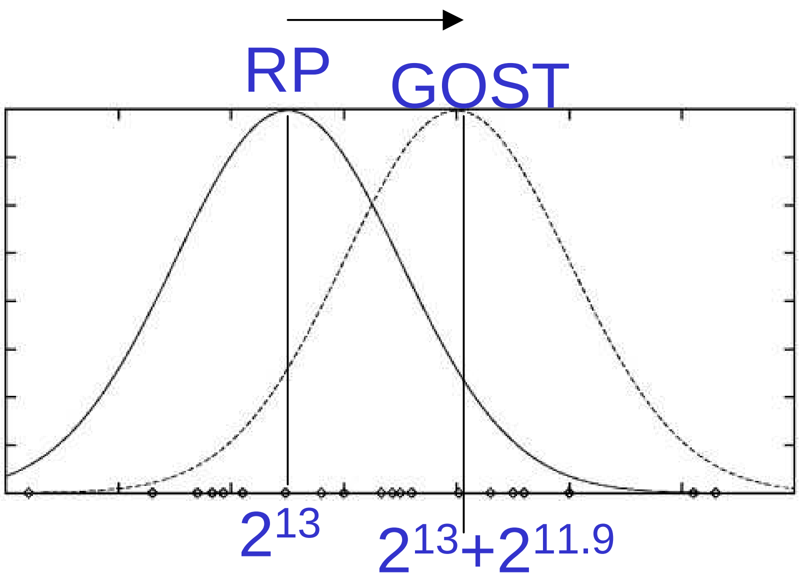
11.9. Distinguishers

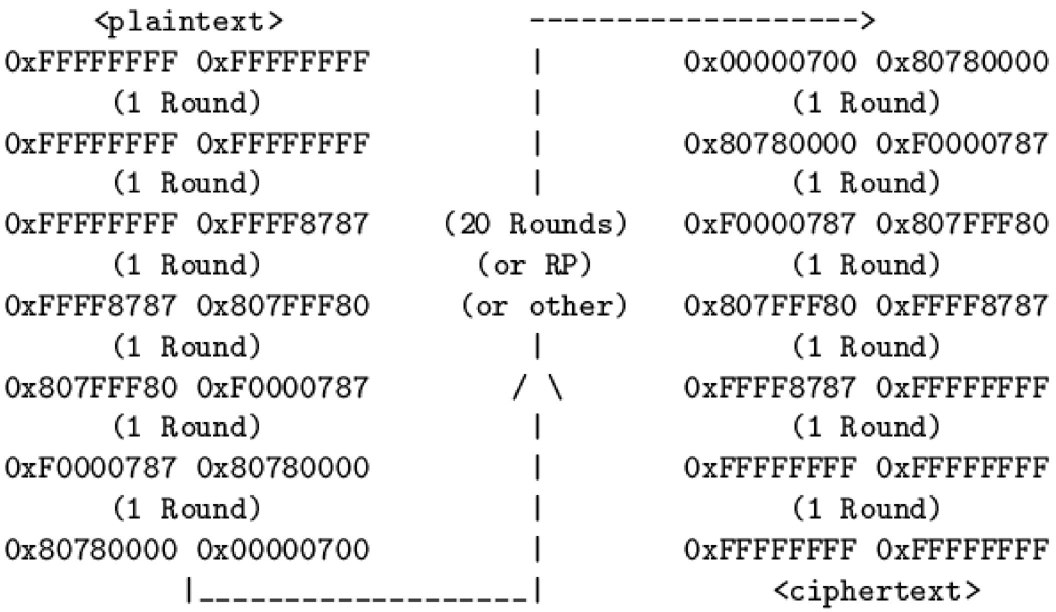
Key Result



Fact 4.2.1 *For the full 32-round GOST and on average over the GOST keys, there exists* $2^{13.0} + 2^{11.9}$ *distinct pairs of plaintexts* $P_i \neq P_j$ *which have the Alpha property.*

If we replace the inner 20 rounds by a random permutation or with GOST with more rounds, we expect only about $2^{13.0}$ *distinct pairs with a standard deviation of* $2^{6.5}$.



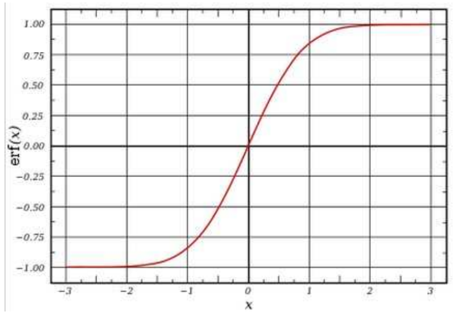


```
      <plaintext>                    ------------------->
0xFFFFFFFF 0xFFFFFFFF           |        0x00000700 0x80780000
      (1 Round)                 |              (1 Round)
0xFFFFFFFF 0xFFFFFFFF           |        0x80780000 0xF0000787
      (1 Round)                 |              (1 Round)
0xFFFFFFFF 0xFFFF8787      (20 Rounds)   0xF0000787 0x807FFF80
      (1 Round)              (or RP)           (1 Round)
0xFFFF8787 0x807FFF80       (or other)   0x807FFF80 0xFFFF8787
      (1 Round)                 |              (1 Round)
0x807FFF80 0xF0000787          / \       0xFFFF8787 0xFFFFFFFF
      (1 Round)                 |              (1 Round)
0xF0000787 0x80780000           |        0xFFFFFFFF 0xFFFFFFFF
      (1 Round)                 |              (1 Round)
0x80780000 0x00000700           |        0xFFFFFFFF 0xFFFFFFFF
           |____________________|              <ciphertext>
```

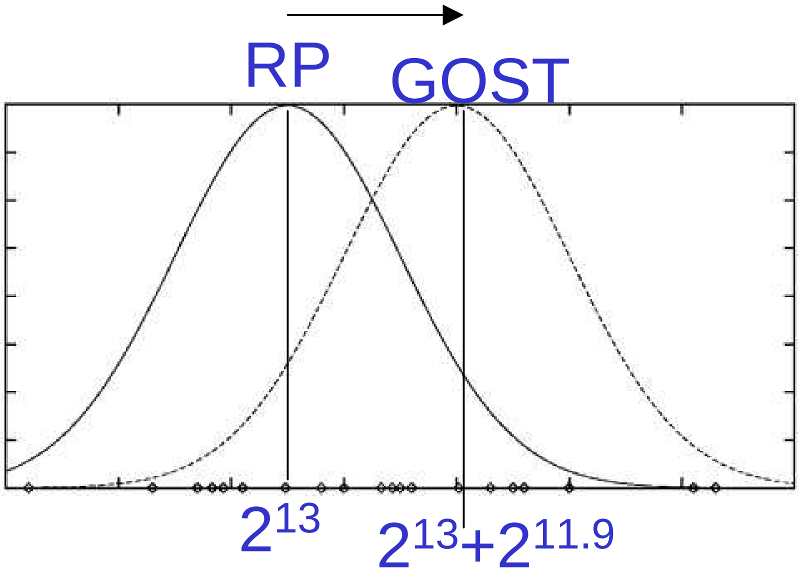
Figure 6: The Alpha Property

Gauss Error

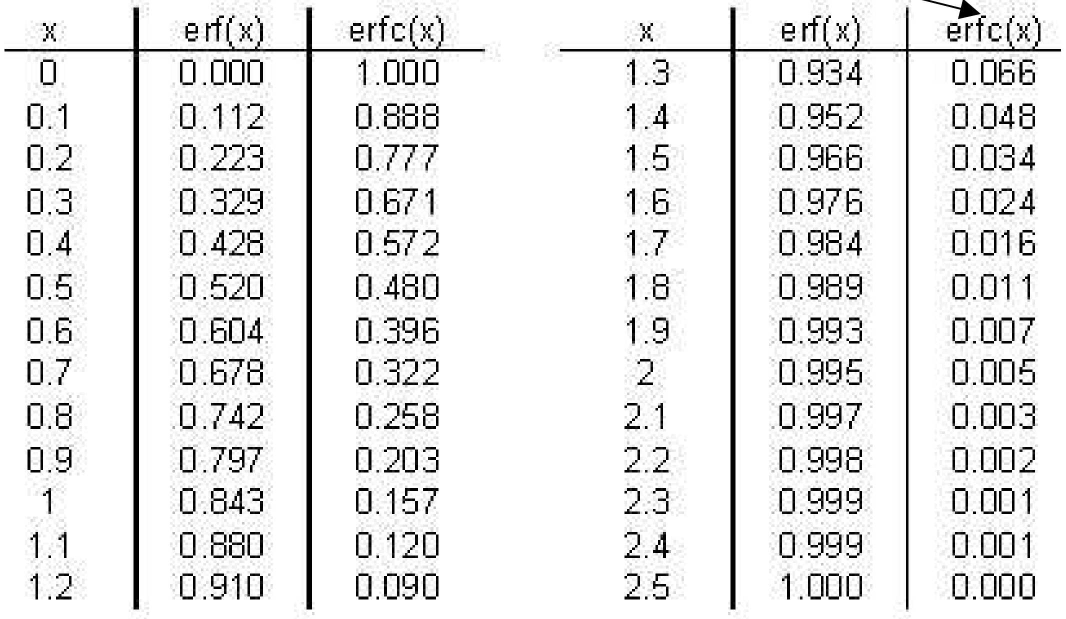
How many
standard deviations?



$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt.$$



Example: right key assumption rejected
= half of this number



x	erf(x)	erfc(x)
0	0.000	1.000
0.1	0.112	0.888
0.2	0.223	0.777
0.3	0.329	0.671
0.4	0.428	0.572
0.5	0.520	0.480
0.6	0.604	0.396
0.7	0.678	0.322
0.8	0.742	0.258
0.9	0.797	0.203
1	0.843	0.157
1.1	0.880	0.120
1.2	0.910	0.090
1.3	0.934	0.066
1.4	0.952	0.048
1.5	0.966	0.034
1.6	0.976	0.024
1.7	0.984	0.016
1.8	0.989	0.011
1.9	0.993	0.007
2	0.995	0.005
2.1	0.997	0.003
2.2	0.998	0.002
2.3	0.999	0.001
2.4	0.999	0.001
2.5	1.000	0.000

Separation

Natural: 2^{13} Attack: $2^{13} + 2^{11.9}$

Crucial Question.

Without this, NONE of differential attacks on GOST work.

We need a solid argument to say that this works.

- a quantitative argument to show that our distinguisher works.
- (and then a precise computation of number of right keys being rejected…)
- Etc…

Separation: Problem

Natural: 2^{13} Attack: $2^{13} + 2^{11.9}$

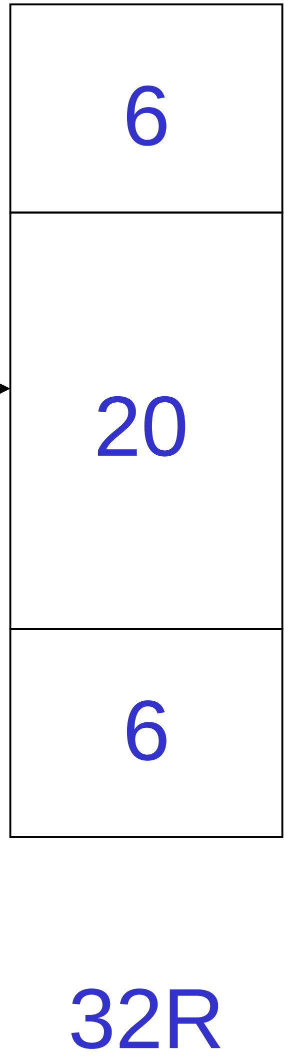
Problem: it does NOT always work.

- For few rounds we get $\mathrm{Max}(2^{13}, 2^{11.9})$.
- For more rounds we get $2^{13} + 2^{11.9}$.

Step By Step

Our plan:

- We will first work on a different case. Not $2^{13} + 2^{11.9}$ but $2^{15} + 2^{13.9}$.
  - For 20 middle rounds.
- Then we will filter out 2^{-2} of cases.
  - Also propagates for the 6+6 outer rounds.



Separation For 20 Middle Rounds

Natural: 2^{15} Attack: $2^{15}+2^{13.9}$.

Problem: it does NOT always work.

- For few rounds we get Max(2^{15},$2^{13.9}$).
- For more rounds we get 2^{15} + $2^{13.9}$.

not holding

def: with middle differences holding

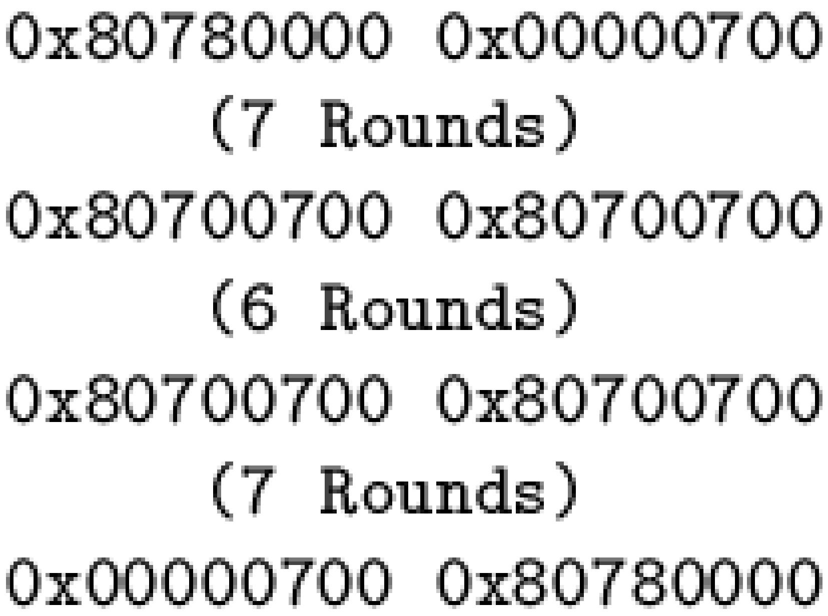
We make an

“artificial distinction” assumption

which separates

the two sets!

20=7+6+7



```
0x80780000 0x00000700
      (7 Rounds)
0x80700700 0x80700700
      (6 Rounds)
0x80700700 0x80700700
      (7 Rounds)
0x00000700 0x80780000
```

Natural Event – Accidental Output Differences

For any 64-bit permutation:

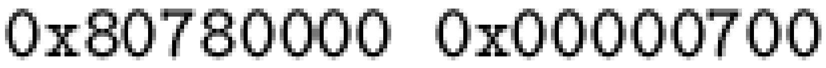
(does NOT have to be a RP!!!)

We have 8 active bits on each side, $2^{8}-1$ differences.

There are $2^{64+8-1} = 2^{71}$ input differences.

Each works with probability $2^{8-64} = 2^{-56}$.

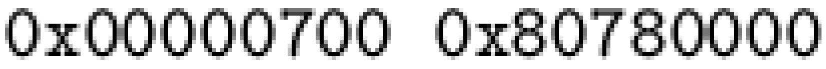
$2^{71-56} = 2^{15}$ survive.



0x80780000 0x00000700

Natural: 2^{15}

XX rounds



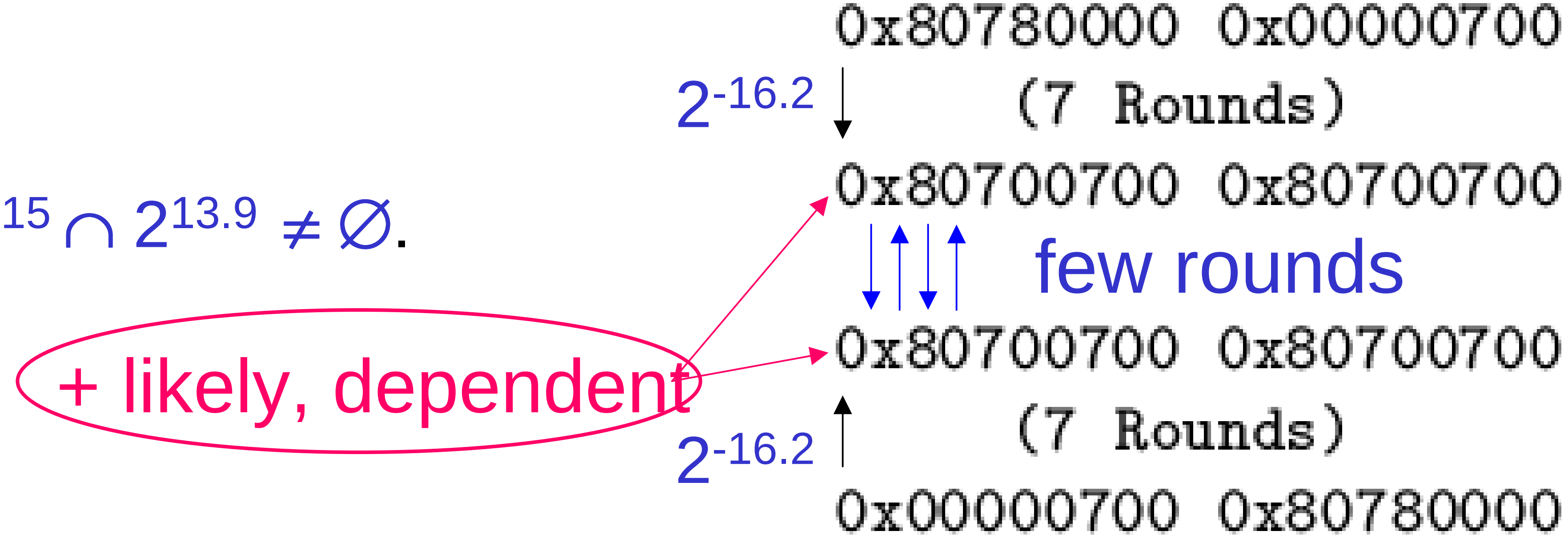
0x00000700 0x80780000

Separation Failure: Less Rounds

For any permutation: we expect 2^{15}.

Propagation in the first 7 rounds:
$2^{-22.2}$ (obtained by simulation).

$2^{15} \cap 2^{13.9} \neq \varnothing$.



0x80780000 0x00000700

$2^{-16.2}$ ↓ (7 Rounds)

0x80700700 0x80700700

few rounds

0x80700700 0x80700700

$2^{-16.2}$ ↑ (7 Rounds)

0x00000700 0x80780000

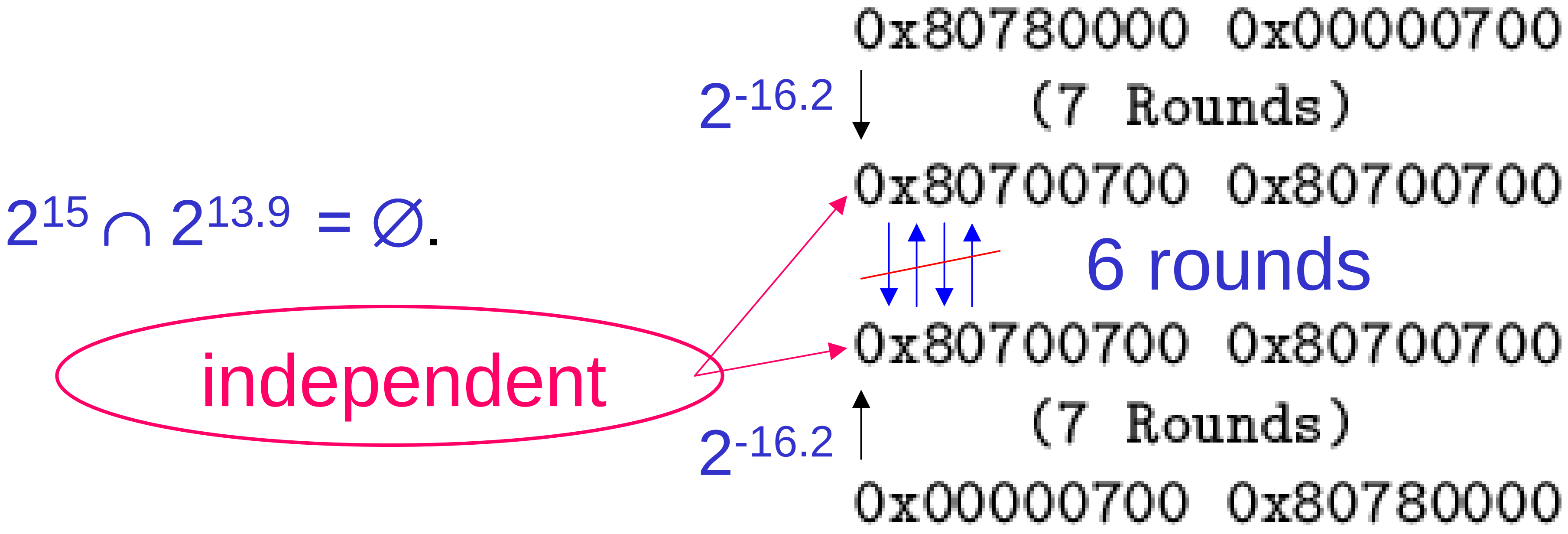
+ likely, dependent

With few rounds in the middle the propagations from both directions will reinforce each other!

Separation Success: More Rounds

For any permutation: we expect 2^{15}.

But only about $2^{15-16.2-16.2} = 2^{-17}$ will have the middle differences required. Zero in practice.



```
0x80780000 0x00000700
        (7 Rounds)
0x80700700 0x80700700
0x80700700 0x80700700
        (7 Rounds)
0x00000700 0x80780000
```

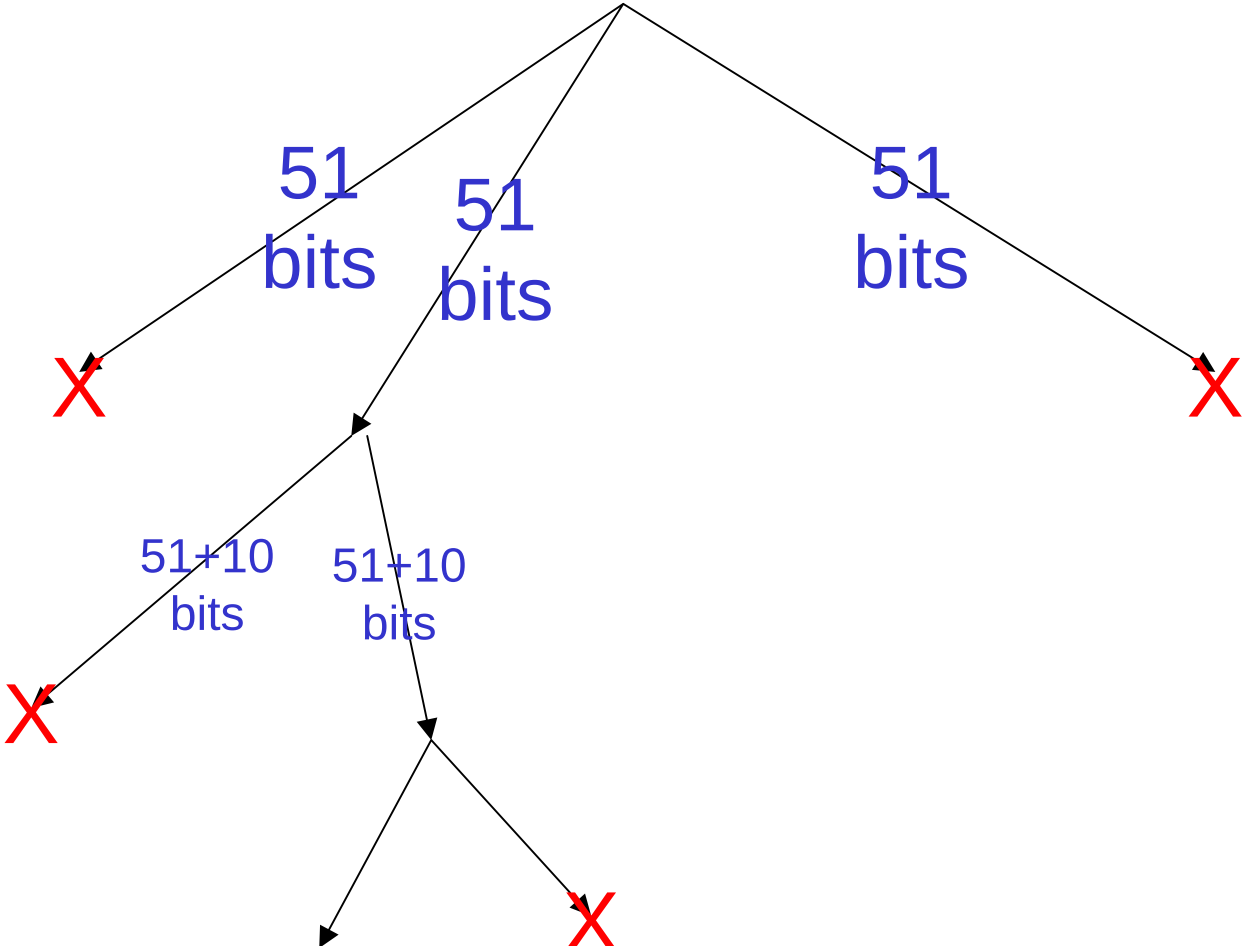
$2^{15} \cap 2^{13.9} = \varnothing$.

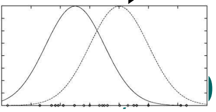
With more rounds no reinforcement.

11.10.
Improved Attacks

Guess Then Eliminate

Depth-First Tree Search.





More Complicated

We need to guess up to 192 key bits in the first 6 rounds. Too costly?

How to avoid it?

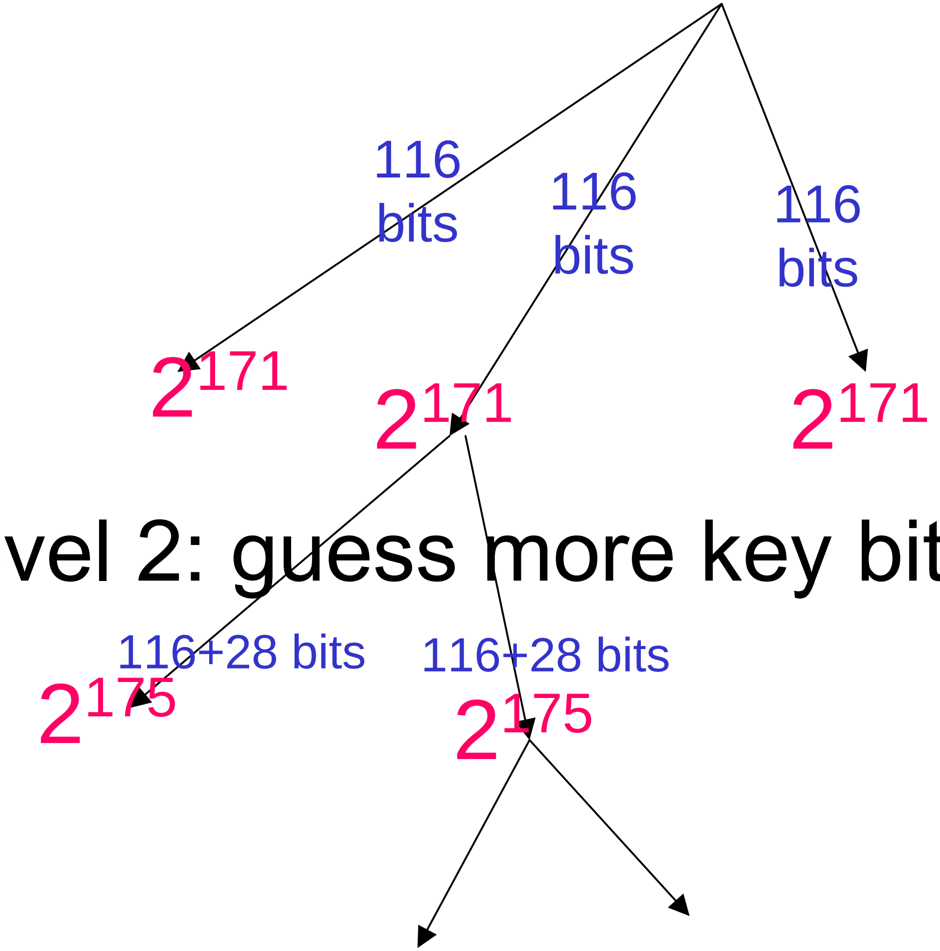
Method 1: Guess 192 key bits => determine $2^{13}+2^{11.9}$ pairs. Too costly.

Method 2: Progressive filtering.

Guess less key bits, determine more pairs, then more key bits but less pairs etc…

More Complicated…

Level 1: Generate Pairs by birthday approach.



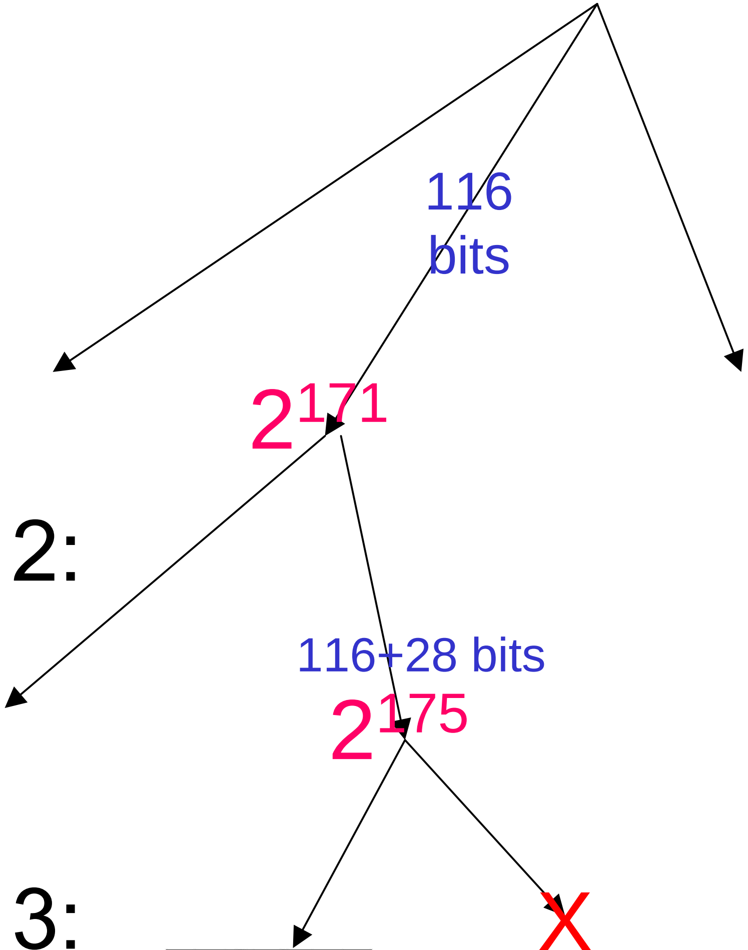
2^{55} cases per key

Level 2: guess more key bits, eliminate cases.

2^{31} cases per key

Much Later:

Level 1:



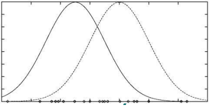
Level 2:

Level 3:

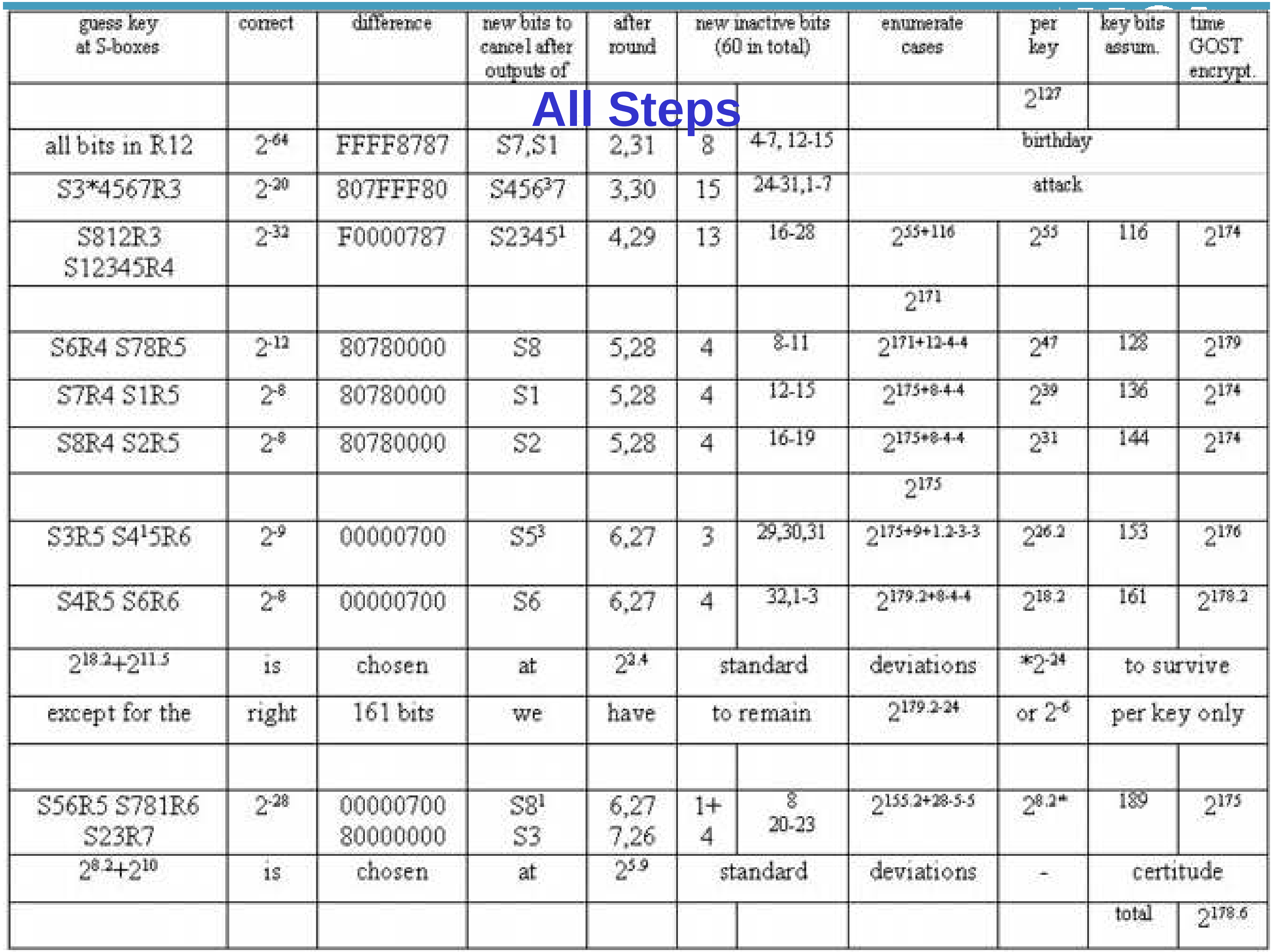
2^{55} cases per key

↓

2^{31} cases per key



All Steps



guess key at S-boxes	correct	difference	new bits to cancel after outputs of	after round	new inactive bits (60 in total)		enumerate cases	per key	key bits assum.	time GOST encrypt.
								2^{127}		
all bits in R12	2^{-64}	FFFF8787	S7,S1	2,31	8	4-7, 12-15	birthday			
S3*4567R3	2^{-20}	807FFF80	$S456^{3}7$	3,30	15	24-31,1-7	attack			
S812R3 S12345R4	2^{-32}	F0000787	$S2345^{1}$	4,29	13	16-28	2^{55+116}	2^{55}	116	2^{174}
							2^{171}			
S6R4 S78R5	2^{-12}	80780000	S8	5,28	4	8-11	$2^{171+12-4-4}$	2^{47}	128	2^{179}
S7R4 S1R5	2^{-8}	80780000	S1	5,28	4	12-15	$2^{175+8-4-4}$	2^{39}	136	2^{174}
S8R4 S2R5	2^{-8}	80780000	S2	5,28	4	16-19	$2^{175+8-4-4}$	2^{31}	144	2^{174}
							2^{175}			
$S3R5\ S4^{1}5R6$	2^{-9}	00000700	$S5^{3}$	6,27	3	29,30,31	$2^{175+9+1.2-3-3}$	$2^{26.2}$	153	2^{176}
S4R5 S6R6	2^{-8}	00000700	S6	6,27	4	32,1-3	$2^{179.2+8-4-4}$	$2^{18.2}$	161	$2^{178.2}$
$2^{18.2}+2^{11.5}$	is	chosen	at	$2^{2.4}$	standard		deviations	$*2^{-24}$	to survive	
except for the	right	161 bits	we	have	to remain		$2^{179.2-24}$	or 2^{-6}	per key only	
S56R5 S781R6 S23R7	2^{-28}	00000700 80000000	$S8^{1}$ S3	6,27 7,26	1+ 4	8 20-23	$2^{155.2+28-5-5}$	$2^{8.2*}$	189	2^{175}
$2^{8.2}+2^{10}$	is	chosen	at	$2^{5.9}$	standard		deviations	-	certitude	
									total	$2^{178.6}$

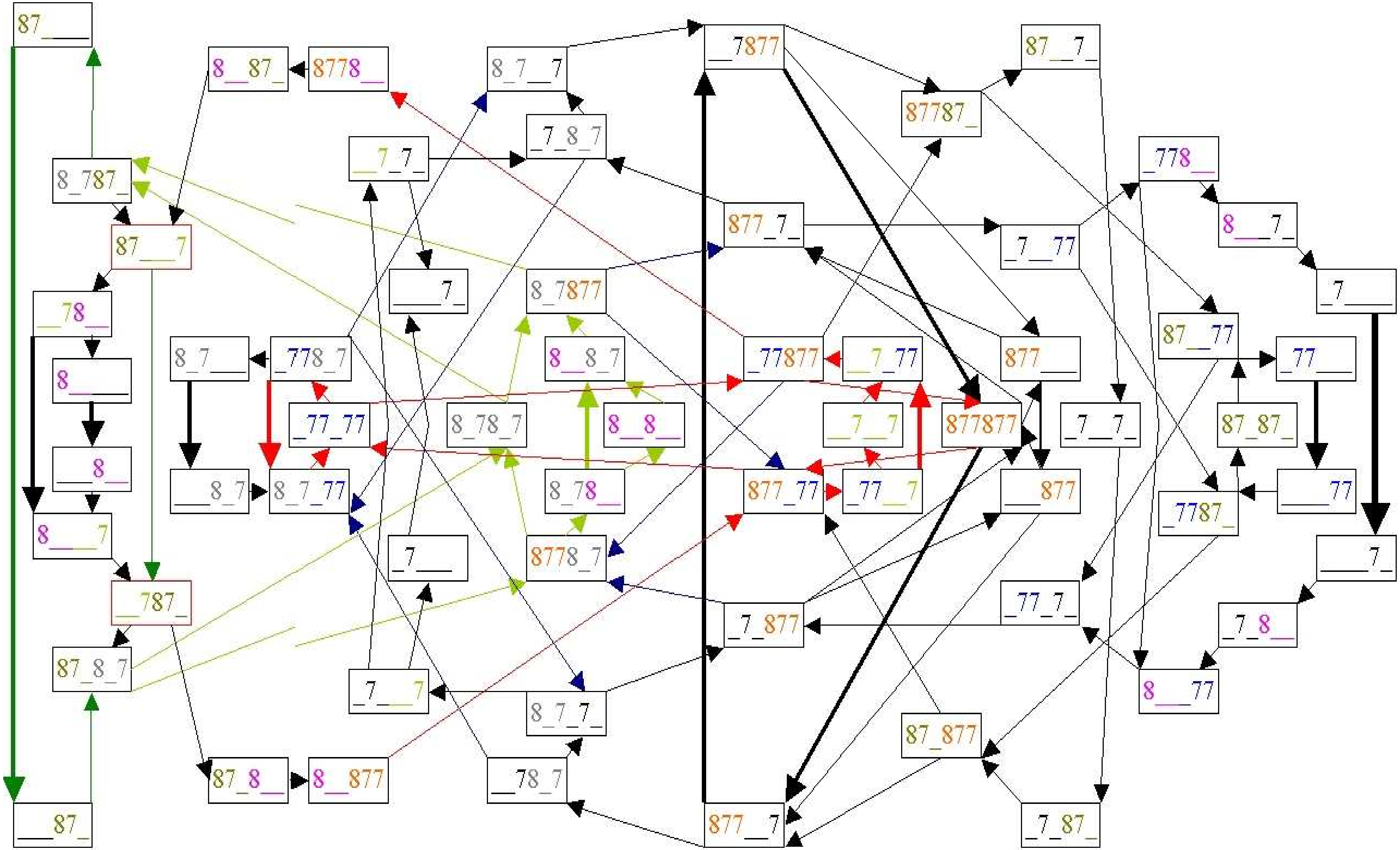
11.11.
Improved Attacks

How To Find Such An Attack

Best differential property
we ever found was found BY HAND.

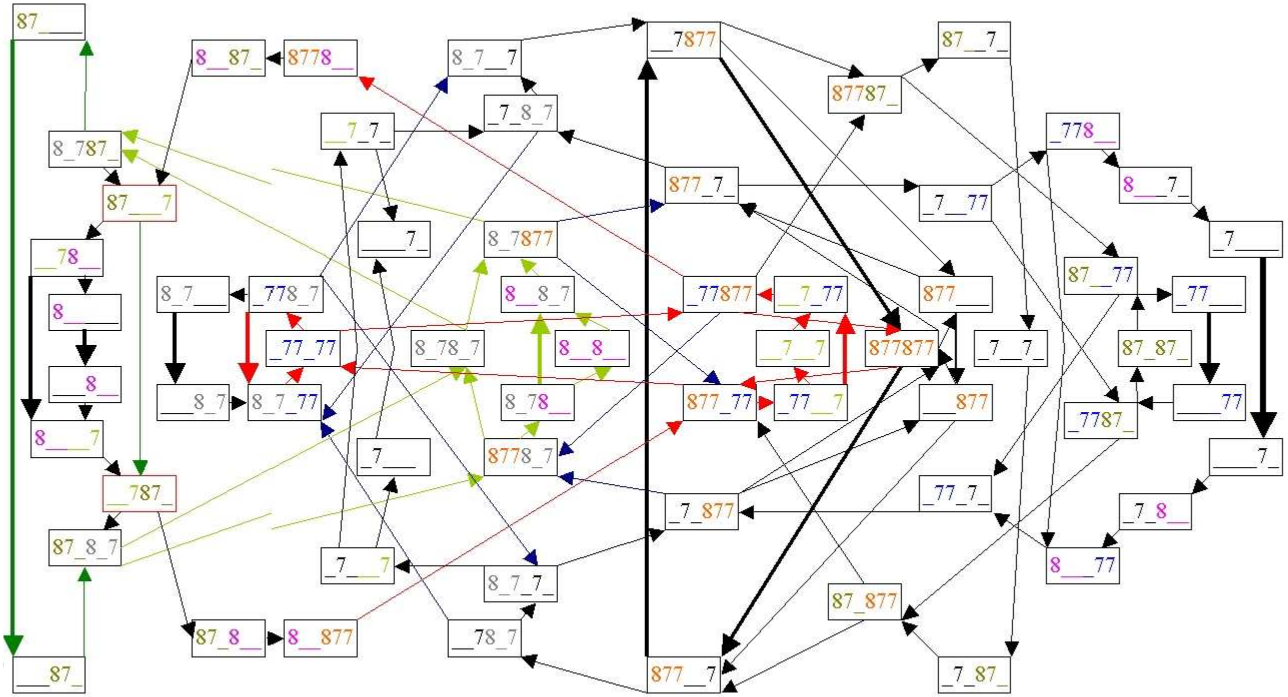
Is systematic approach possible?

Our Attack = Graph Walks With Costs



Remark:

- the structure of this graph does NOT depend on the S-boxes
- only costs (probabilities) depend on the S-boxes



13. New Attacks…

Strange Ideas…

13.1. Amplification Paradox

Involution => Amplification

1 pair 16 R =>

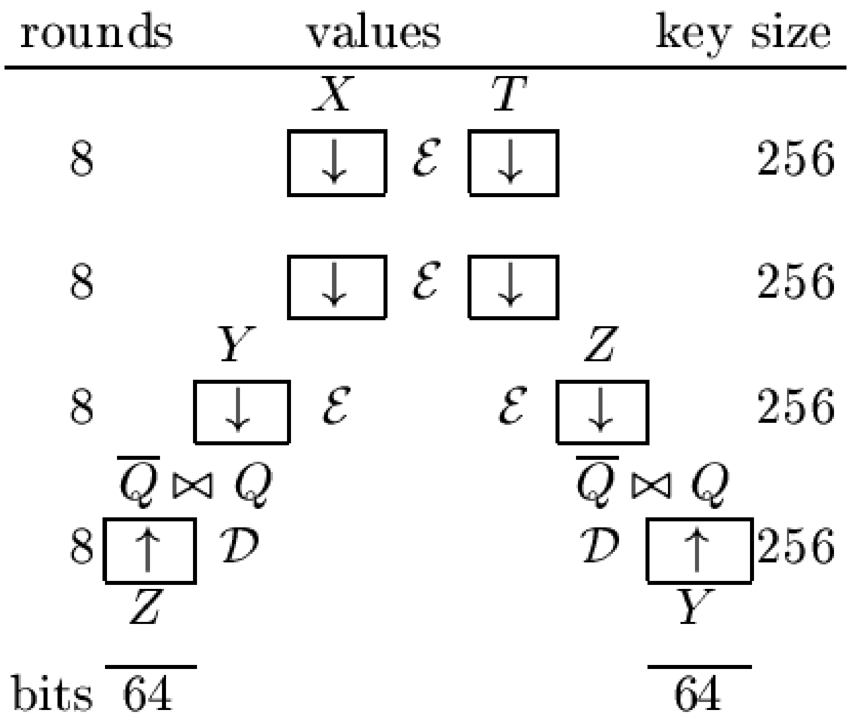
another pair for free

$$Y = \mathcal{E}^2(X)$$

$$\vdash$$

$$Enc_k(X) = \mathcal{E}^2(Dec_k(Y))$$

can we continue?



Bad News

continue?

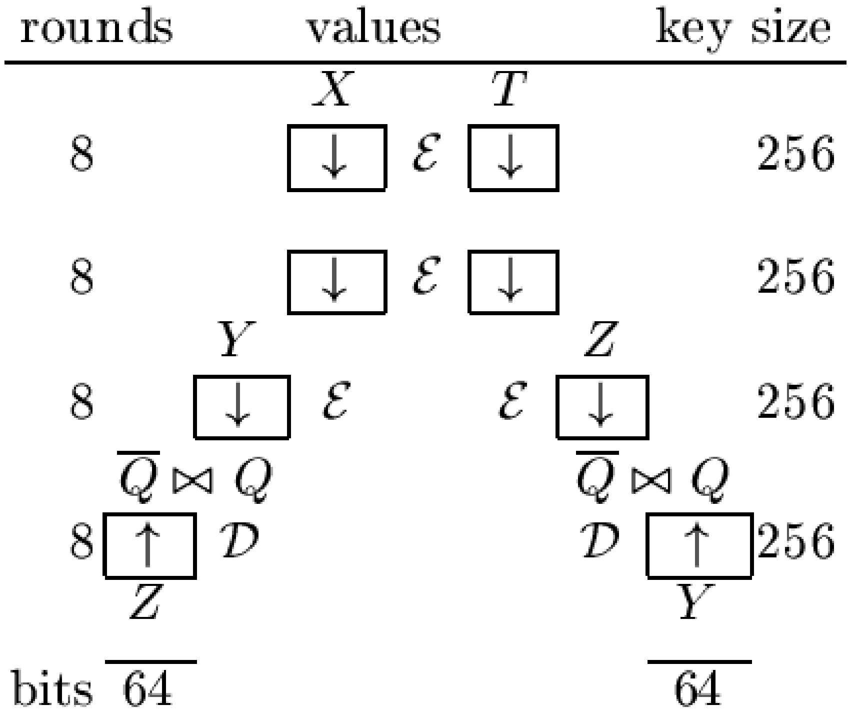
$$Y = \mathcal{E}^2(X)$$

$$\vdash$$

$$Enc_k(X) = \mathcal{E}^2(Dec_k(Y))$$

$$\vdash$$

$$Enc_k(Dec_k(Y)) = \mathcal{E}^2(Dec_k(Enc_k(X)))$$


rounds
values
key size
X
T
8
↓
$\mathcal{E}$
↓
256
8
↓
$\mathcal{E}$
↓
256
Y
Z
8
↓
$\mathcal{E}$
$\mathcal{E}$
↓
256
$\overline{Q} \bowtie Q$
$\overline{Q} \bowtie Q$
8
↑
$\mathcal{D}$
$\mathcal{D}$
↑
256
Z
Y
bits
64
64