Cryptanalysis of GOST

Nicolas T. Courtois
University College London, UK
Outline

1. Cold War cryptography
2. Cryptanalysis
3. Solvers AC and Low-Data Attacks
4. Self-Similarity in cryptanalysis, 4.4. Sliding attacks
5. GOST submission to ISO [2010]
6. Algebraic Complexity Reduction
7. Multiple-key attacks and Bicliques
8. AC Reduction+DC
9. Multiple-point attacks
10. Multiplicative Complexity, Optimisation of S-boxes
11. Diffusion in GOST, Low-Level Attacks, UNSAT Immunity, MITM
12. Inference, Differential Induction, Saturation Analysis
15. Combinatorial Explorations
16. Strange Attacks
17. GOST Hash
What’s Wrong? >50 distinct attacks… Best = $2^{101}$

- Weak Key Schedule
- Poor Diffusion
- Self-similarity
- Guess Then …

"Algebraic Complexity Reduction"

- Reflection
- Slide
- Fixed P.
- Involution

Combination attacks
best = $2^{101}$

- AC / Software / SAT Solvers
- multiple random keys
- MITM
- multiple points, HO

- Combinatorial Optimisation
- Truncated Differentials (DC)

© Nicolas T. Courtois, 2006-2013
1.1. History
History: Cold War
Russia vs. USA
- My Favourite Groups

Your Groups (51)  Reorder »

- Code Breakers  Members (311)

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

- Data ENCRYPTION

- IACR Cryptographers
code breakers ==

взломщики кодов
Cryptanalysis

from Greek

• kryptós, "hidden"
• analýein, "to untie"

Term coined in 1920

by William F. Friedman.

• Born in Moldavia, emigrated to US in 1892.
• Chief cryptologist at National Security Agency in the 50s.
History: 1918

- Tzarist secret services
  => continued their work with the armies of white generals.
- The Red Army General Staff created a new service.
  - Not very strong crypto.
- In 1918 - 1920 almost all encrypted correspondence of the General Staff and the Soviet Government was easily broken by
  - the white (counterrevolutionary) armed forces
  - the British,
  - the Swedish
  - the Polish
  - won the War against Russia in 1920-1921
  - the Warsaw “miracle” victory was due to decryption of key messages…
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”…
Code Makers: M-100 Family

1930s

141 kg, in a van.

used until 1954 or so.
Fialka = Фиалка = Violet = M-125

Around 1965.
MUCH stronger than Enigma…
Used until 1987 in East Germany…
Strength of Fialka

- About $2^{151}$ initial settings
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

© Nicolas T. Courtois, 2006-2013
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
1.2. Modern Ciphers
1.3. DC – Old History
DC

Differential Cryptanalysis (DC) is based on tracking of changes in the differences between two messages as they pass through the consecutive rounds of encryption.

It is one of the oldest classical attacks on modern block ciphers, if not the oldest.
Motivation

In cryptographic literature it was first described and analyzed by Biham and Shamir and applied to DES algorithm.
late 1980s/early 1990s
Motivation

However Coppersmith, a member of the IBM team which have designed DES have reported that this attack was already known to IBM designers around 1974.

I was present in person during his invited talk at Crypto,

=>it is was not IBM but the NSA have chosen the DES S-boxes
Motivation

It was known [IBM 1970s] under the name of T-attack or Tickle attack and DES have been already and specifically designed to resist to this type of attack.

In fact pathologically weak with some random S-boxes.
Motivation

IBM have agreed with the NSA that the design criteria of DES should not be made public.

This precisely because it would “weaken the competitive advantage the United States enjoyed over other countries in the field of cryptography”.

cf. wikipedia, old crypto newsgroups
2. 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 CRYPTOANALYSIS
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
What Can be Done?

Algebraic Cryptanalysis:
- Very special ciphers: 1 M rounds [Courtois’AES4].
- General ciphers:
  SMALL number of rounds, 4,5,6 rounds.
  - If key size > block size – more rounds.
  - CTC2(96,256,10) can be broken.
Def: “I / O Degree” = “Graph AI”

Consider function $f : \mathbb{GF}(2)^n \rightarrow \mathbb{GF}(2)^m$, $f(x) = y$, 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 $y = f(x)$. 
• [2002] XSL paper:

2 “crazy” conjectures:

- **I/O Degree Hypothesis (IOH):** all ciphers with low I/O degree and lots of I/O relations may be broken?
- **The Very Sparse Hypothesis (VSH):** ciphers with very low gate count broken?
Algebraic Attacks on Block Ciphers

1. Write +
2. Solve [key recovery].
GOST, Self-Similarity and Cryptanalysis of Block Ciphers

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.
2.2. 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...

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-CNФ - The Outsider

Principle 1:

each monomial = one dummy variable.

\[ a = wxyz \]

\[ \iff \]

\[ a \iff (w \land x \land y \land z) \]

\[ \iff \]

\[ (w \lor \bar{a})(x \lor \bar{a})(y \lor \bar{a})(z \lor \bar{a})(a \lor w \lor x \lor y \lor z) \]

\[ d+1 \] clauses for each degree \( d \) monomial
Also

Principle 2:
Handling XORs – Not obvious. Long XORs known to be hard problems for SAT solvers.

\[ a \oplus b \oplus c \oplus d = 0 \]

\[ (\bar{a} \lor b \lor c \lor d) (a \lor \bar{b} \lor c \lor d) (a \lor b \lor \bar{c} \lor d) (a \lor b \lor c \lor \bar{d}) \]

\[ (\bar{a} \lor \bar{b} \lor \bar{c} \lor d) (\bar{a} \lor \bar{b} \lor c \lor \bar{d}) (\bar{a} \lor b \lor \bar{c} \lor \bar{d}) (a \lor \bar{b} \lor \bar{c} \lor \bar{d}) \]

- Split longer XORs in several shorter with more dummy variables.
- About 4h 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”.

© Nicolas T. Courtois, 2006-2013
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

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 $\infty$. 
What Are the Limitations of Algebraic Attacks?

• When the number of rounds grows: complexity jumps from 0 to $\infty$.

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

$\Rightarrow$ jumps from $\infty$ 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.
GOST, Self-Similarity and Cryptanalysis of Block Ciphers

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 \( y = f(x)\).
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…


<table>
<thead>
<tr>
<th>Rounds</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key size</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>Data</td>
<td>2 KP</td>
<td>2 KP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>See</th>
<th>Fact 3</th>
<th>[32]</th>
<th>Fact 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf.</td>
<td>page 11</td>
<td>Fact 112</td>
<td>page 152</td>
</tr>
<tr>
<td>cf. also</td>
<td>[69]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory bytes</th>
<th>small</th>
<th>$2^{43}$</th>
<th>$2^{46}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$2^{24}$</td>
<td>$2^{128}$</td>
<td>$2^{127}$</td>
</tr>
</tbody>
</table>

cf. 2011/626

(more in Section 11 of these slides)

© Nicolas T. Courtois, 2006-2013
Up to 8 Rounds

Our methods…

… allow to break up to 8 rounds of GOST…

Can we ever hope to break 32 rounds?
Can We Solve 8R?
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. Identical Boolean functions are implemented differently.

Camouflage?

Due to a combination with another terrible weakness half of the silicon is wasted…
Crypto-1 Algo + Auth. Protocol
Background – Crypto 1 Cipher

- 48 bit LFSR
- The feedback function $L$ is as follows:

  \[
  \text{Definition 2.1. The feedback function } L : \mathbb{F}_2^{48} \rightarrow \mathbb{F}_2
  \text{ is defined by } L(x_0, x_1, \ldots, x_{47}) := x_0 \oplus x_5 \oplus x_9 \oplus x_{10} \oplus
  x_{12} \oplus x_{14} \oplus x_{15} \oplus x_{17} \oplus x_{19} \oplus x_{24} \oplus x_{25} \oplus x_{27} \oplus x_{29} \oplus
  x_{35} \oplus x_{39} \oplus x_{41} \oplus x_{42} \oplus x_{43}.
  \]

- The non-linear filter function $f$ is defined as:

  \[
  f(x_0, x_1, \ldots, x_{47}) := f_c(f_a(x_9, x_{11}, x_{13}, x_{15}),
  f_b(x_{17}, x_{19}, x_{21}, x_{23}), f_b(x_{25}, x_{27}, x_{29}, x_{31}),
  f_a(x_{33}, x_{35}, x_{37}, x_{39}), f_b(x_{41}, x_{43}, x_{45}, x_{47})).
  \]
Cryptol Cipher

Key

Tag/Reader IV

0 5 9 10 11 12 13 14 15 17 18 19 21 23 24 25 27 29 31 33 35 37 39 41 42 43 45 47

$\begin{align*}
f_a^4 &= 0x9E98 = (a+b)(c+1)(a+d)+(b+1)c+a \\
f_b^4 &= 0xB48E = (a+c)(a+b+d)+(a+b)cd+b
\end{align*}$

Tag IV $\oplus$ Serial is loaded first, then Reader IV $\oplus$ NFSR
Waste of Silicon

Internal bits are computed 2-3 times. One could save half of the gates!
Crypto1 Cipher

Key

\[ f_4^a = 0x9E98 = (a+b)(c+1)(a+d)+(b+1)c+a \]
\[ f_4^b = 0xB48E = (a+c)(a+b+d)+(a+b)cd+b \]
4.2. Hitag 2 Cipher
Hitag2 Cipher

Serial

Key

$32 \times \text{init}$

inverse(first 32 keystream bits) = authenticator

$f_a^4 = \text{0x2C79} = abc + ac + ad + bc + a + b + d + 1$

$f_b^4 = \text{0x6671} = abd + acd + bcd + ab + ac + bc + a + b + d + 1$
Strong or Weak?

High Algebraic Immunity.

- Does NOT help.
- Many “direct” algebraic attacks exist. We can break “any cipher”, if not too complex…

First efficient attack on this cipher was an Algebraic Attack [Courtois, O’Neil, Nohl, see eprint/2008/166]. Soon became obsolete.
Exhaustive Key Search

- 48 bits, about 4 years on 1 CPU.
  - Hours with FPGA.

Our First Attack [04/2008]
- 12 seconds on the same CPU.

Better Attack [09/2008]
- 0.05 seconds.
  [de Koning Gans et al, Esorics 2008]
Beyond Crypto-1

...AC can break “any cipher”, if not too complex...
But other attacks are faster...

• However,
  – our attack does NOT require human intervention
  – more generally applicable:
    we can also break Hitag2 in 1 day
    (instead of say 4 years).
    • has fully irregular taps. See: Inversion attacks:
      [Ross Anderson: Searching for the Optimum Correlation Attack, FSE'94]
“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 is more than 10 times better 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: $E_k = g_k \circ f^8_k$.

$528 = 16 + 8 \times 64$ 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

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

\[
\begin{array}{cccc}
P_i & 64 \text{ rounds} & P_j & 64 \text{ rounds} \\
& & & 64 \text{ rounds} \\
& & & 64 \text{ rounds} \\
C_i & & & C_j \\
\end{array}
\]

Gives an unlimited number of other sliding pairs !!!

very large "Amplification"
KeeLoq and Sliding


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

```
<table>
<thead>
<tr>
<th>64 rounds</th>
<th>64 rounds</th>
<th>64 rounds</th>
<th>64 rounds</th>
<th>16 r</th>
</tr>
</thead>
</table>
```

Classical sliding fails – because of the “odd” 16 rounds:
Classical Sliding – Not Easy

HARD - Problem:

What’s the values here?
Answer [Courtois, Bard, Wagner FSE2008]:

Algebraic Sliding

---

© Nicolas T. Courtois, 2006-2013
Algebraic Attack [FSE 2008]

We are able to use $C_i, C_j$ directly!

Write and merge 2 systems of equations:

ignore all these!

(common 64-bit key)

(like 2 different ciphers)
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.
4.5. AES Cipher
Iterated Permutation Attack on Multiple AES

\[ E_k = g_k \circ f^7_k. \]

- Take AES-256 \( f_k \).
- Iterate 7 times \( f^7_k \).
- Re-encrypt with AES-128 \( g_k \).

This is INSECURE,

\[ \Rightarrow \text{Attack in } 2^{128}. \quad \text{Memory } = 2^{128}. \]

Same as our attack on KeeLoq [Tatracrypt’07].
Compare to KeeLoq [Courtois, Bard, TatraCrypt07]

\[ E_k = g_k \circ f^8_k. \]

- Guess 16 key bits.
- Confirm if correct. (!)
- Recover missing key bits by
  - an algebraic attack.
  - correlation attack
  - other..
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 $f : \mathbb{GF}(2)^n \rightarrow \mathbb{GF}(2)^m$, $f(x) = y$, 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

\[ + \text{ modulo } 2^{32} \]

in several ciphers: GOST, SNOW 2.0.

\[(x, y) \mapsto z = x \oplus y \mod 2^n\]

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

Proof:

we have:

$$\begin{align*}
xy + (x + y)c &= (x + c)(y + c) - c^2 \\
1 \times \text{each}
\end{align*}$$
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 $GF(2)^m$. 

© Nicolas T. Courtois, 2006-2013
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 $x+y=111111\ldots111$.

$\Rightarrow$ the component CAN be linearized.
This paper:

For $+ \mod 2^n$:

- The Conditional Describing Degree is 1 (!)
- So what?
  - View it as follows: fix $n$ linear equations, get $2n$!
- $2x$ 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.
Main Result:

**Theorem 1:** Let \( z \) be a fixed output for \( P \).

- If \( z \) has \( r \) consecutive 1s from the bit \( i \geq 0 \) to the bit \( i + r - 1 \leq n - 1 \) in its binary representation, then there is a \( p \)-probable set \( E \) of \( r + 2 \) (only \( r + 1 \) if \( i + r - 1 = n - 2 \), and \( r \) if \( i + r - 1 = n - 1 \)) linear equations for \( P \), with \( p = \frac{1}{2} + \frac{1}{2^{i+1}} \).

- If \( z \) has \( r \) consecutive 0s from the bit 0 to the bit \( r - 1 \leq n - 1 \) in its binary representation, then there is a \( p \)-probable set \( E \) of \( r + 2 \) (only \( r + 1 \) if \( r - 1 = n - 2 \), and \( r \) if \( r - 1 = n - 1 \)) linear equations for \( P \), with \( p = \frac{1}{2} \).

- If \( z \) has \( r \) consecutive 0s from the bit \( i \geq 0 \) to the bit \( i + r - 1 \leq n - 1 \) in its binary representation, then there is a \( p \)-probable set \( E \) of \( r + 2 \) (only \( r + 1 \) if \( i + r - 1 = n - 2 \), and \( r \) if \( i + r - 1 = n - 1 \)) linear equations for \( P \), with \( p = \frac{1}{2} - \frac{1}{2^{i+1}} \).
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% of inputs of this S-box/operation. Very strong and helps AC a lot.
We analyse the keystream generator only, as a cipher with 576 key bits.

Any attack faster than $2^{576}$ is interesting…
Conditional algebraic attacks:

Amplification:

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

• $2x$ for $+ \mod 2^n$.

• $4x$ 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.
Future work

What is the optimal guess-then-algebraic attack on Snow 2.0. at degree 3?

– Maybe less guessing makes it solvable…
  • Future research…
4.8. Variants of Snow 2.0. Sosemanuk, ZUC, etc.
**4G Telephony / LTE: Chinese Variant of Snow**
**ZUC Cipher in 4G**

\[ \pi \approx \frac{355}{113} \]

Zu Chongzhi

[429-500]
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.
Why Declassified

• 1994:
  – Shortly after the dissolution of the USSR, in a very troubled period where locations of nuclear weapons were sold for 5 $, it was indeed declassified and released to the public.
  – By mistake???
  – No country ever declassified their national algorithm.
  • In the UK no journalist would ever write anything about UK or NATO cryptography, due to so called DA-Rules
    – (BTW. Russia, China, Japan is not in NATO)
    – Secret algorithms, never made public, not even 50 years later…
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, from the English preface to a translation of the Russian standard, by Aleksandr Malchik and Whitfield Diffie, Link: http://193.166.3.2/pub/crypt/cryptography/papers/gost/russian-des-preface.ps.gz

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

- 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…
Research on GOST

Before 2010 there were many papers on
- weak keys in GOST,
- attacks for some well-chosen number of rounds [Kara, some sliding attacks],
- attacks with modular additions removed [Biryukov-Wagner]
- related-key attacks [Kelsey, LucksFleischmann, Russian rebuttal]
- reverse engineering attacks on S-boxes [Saarinen, Furya]
- and collision and pre-image attacks on the hash function based on this cipher [Mendel, Szmidt et al.].

In all these attacks the attacker had much more freedom than we allow ourselves.
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”
“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.”

-----disagree, it seems that bijective S-boxes are always quite secure, even identity functions!

“A huge number of rounds (32) and a well studied Feistel construction combined with Shannon's substitution-permutation sequence provide a solid basis for GOST's security.”

“However, as in DES everything depends on the exact choice of the S-boxes and the key-schedule.”

NO YES!
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


- 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:
    - TDES, MISTY1, CAST-128, HIGHT
  - Only three 128-bit block ciphers:
    - AES, Camellia, SEED
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].
Finally..

GOST was rejected at ISO

• by a majority vote
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 […]

• Why?
  – Half-broken in very serious sense
  – Really broken in academic sense
What's Wrong? >50 distinct attacks… Best = $2^{101}$

Weak Key Schedule

Poor Diffusion

Self-similarity

"Algebraic Complexity Reduction"

Reflection Slide Fixed P. Involution

AC / Software / SAT Solvers

multiple random keys

MITM

combination attacks

best = $2^{101}$

Combinatorial Optimisation

Truncated Differentials (DC)

multiple points, HO

Best = $2^{179}$
6. Algebraic Complexity Reduction
Conditional AC

Definition [informal on purpose] Methods to substantially reduce the size of and the complexity of equations that appear throughout the computations…

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

How to lower the degree?

• By adding new equations
• Which split the system into pieces and decrease the number of rounds
[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 
\textit{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 never studied 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].
Revision: Feistel Schemes
2x Same
6.2. Structure of GOST

\[ Enc_k = D \circ S \circ E \circ E \circ E \]
Self-Similar Key Schedule
Periodic Repetition + Inversed Order

<table>
<thead>
<tr>
<th>rounds</th>
<th>1</th>
<th>8</th>
<th>9</th>
<th>16</th>
<th>17</th>
<th>24</th>
<th>25</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>keys</td>
<td>$k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$</td>
<td>$k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$</td>
<td>$k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$</td>
<td>$k_7 k_6 k_5 k_4 k_3 k_2 k_1 k_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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]:

$$\text{Enc}_k = D \circ S \circ E \circ E \circ E$$

(1)

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

© Nicolas T. Courtois, 2006-2013
*Compare: DES

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
Fixed Points: DES Key Schedule

• 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 $G = + 13 \text{ mod } 28$ (and not 14)
• Does NOT have many fixed points.
"Theorem Which Won World War 2",

\[ E_n C_k = D \circ S \circ E \circ E \circ E \]

P and
\[ Q^{-1} \circ P \circ Q \]

have the same cycle structure.
Last 16 Rounds of GOST

\[ E_n c_k = D \circ S \circ E \circ E \circ E \]

“‘Theorem Which Won World War 2’”,

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

⇒ 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 amplification => 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

\[ Enc_k = D \circ S \circ E \circ E \circ D \circ \bar{D} \circ E \circ E \]

\[ E \]

\[ A \]

\[ B \]

\[ C \]

\[ D \]

\[ \bar{D} \]

\[ 256 \]
Two Encryptions with A Slide

not similar
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 $D = \mathcal{E}^3(A)$. 
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
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.

<table>
<thead>
<tr>
<th>rounds</th>
<th>values</th>
<th>key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$\mathcal{E}$</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>$A$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$\downarrow$</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$\downarrow$</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>$C$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$\downarrow$</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>$D$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$D$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$\uparrow$</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>$C$</td>
<td></td>
</tr>
</tbody>
</table>

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

New Attack on GOST

$2^{64}$ KP

guess A, B

correct $P=2^{-128}$

$P=2^{-128} => 4$ pairs for 8 rounds
6.5. Can We Solve 8R?
Final Key Recovery 8R

4 Pairs, 8 rounds.
The key is found within $2^{94}$ GOST computations.

(more about this in Section 11. of these slides)
Overall Attack

$2^{128+94}$ GOST computations.

$2^{33}$ 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
6.6. More Single Key Attacks…
Many more single-key attacks on full 32-round GOST...

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

<table>
<thead>
<tr>
<th>Reduction Type</th>
<th>Red. 1 §9.1</th>
<th>Red. 2 §10</th>
<th>Red. 3 §11</th>
<th>Red. 4 §11.1</th>
<th>Red. 5 §12</th>
</tr>
</thead>
<tbody>
<tr>
<td>From (data 32 R)</td>
<td>1x Internal Reflection</td>
<td>2(^{32}) KP</td>
<td>2 KP</td>
<td>3 KP</td>
<td>3 KP</td>
</tr>
<tr>
<td>Obtained (for 8R)</td>
<td>2x Reflection</td>
<td>2(^{64}) KP</td>
<td>2(^{-96})</td>
<td>2(^{-128})</td>
<td>2(^{-96})</td>
</tr>
<tr>
<td>Valid w. prob.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Last step</th>
<th>MITM</th>
<th>Guess+ Det. Hybrid MITM-Software/Algebraic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases ∈ Inside Then Fact cf.</td>
<td>2(^{128})</td>
<td>2(^{128})</td>
</tr>
<tr>
<td>Time to break 8R</td>
<td>Fact 9 2(^{128})</td>
<td>Fact 4 2(^{127}/2^{128})</td>
</tr>
<tr>
<td>Storage bytes</td>
<td>2(^{132})</td>
<td>2(^{39}/2^{46})</td>
</tr>
<tr>
<td># false positives</td>
<td>2(^{224})</td>
<td>2(^{192})</td>
</tr>
<tr>
<td>Attack time 32 R</td>
<td>2(^{224})</td>
<td>2(^{228}/2^{224})</td>
</tr>
</tbody>
</table>
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 about, 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 \textbf{(best attack)} / \textbf{(implementation cost)} ratio
\begin{itemize}
  \item Key schedule could be easily fixed to avoid academic shortcut attacks…
  \item GOST-P is even better (better S-box <= PRESENT: new ISO standard).
\end{itemize}

cf. Poschmann et al CHES 2010
6.7. 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=$2^{238}$
6.8. Double Reflection Attack
2x Reflection, Happens About Once:

- 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:

\[ D = 2^{64} \quad T = 2^{179} \]


However ciphers are NEVER used with single keys in the real life… On the contrary.
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 $2^{-64}$

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

No guessing =>

Very high amplification.

All data obtained nearly “for free”.

© Nicolas T. Courtois, 2006-2013
8. Combined Attacks: DC + Algebraic Complexity Reduction

two totally unrelated families of attacks…
…until December 2012
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.
8.1. GOST and DC

...DC is yet another form of self-similarity (!)
GOST vs. LC and DC

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

“Against DC and LC, GOST is probably stronger than DES”

Gabidulin 2000-2001:

7 rounds are sufficient to protect GOST against DC.

8.1.2. DC With Sets
Advances Differential Cryptanalysis of GOST

[Seki, Kaneko SAC 2000]:
Some 13 rounds out of 32 broken…

Sets of differentials = most general formulation
Incomplete/truncated Differentials = With free bits…
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
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!
8.1.3.
Better Sets [2011]
Recent Differential Attacks on GOST

References:

   => invention of new sets

   => first simple attack (very slightly) faster than brute force $2^{254.6}$

   => progressive improved approach, heuristic and not very precise… $2^{226}$

   => symmetric + many further refinements + very careful work on individual bits + tight [barely working] distinguishers + justification of earlier results $2^{179}$


   => discovery that size matters in advanced differential attacks, work on various sets of S-boxes, search for optimal properties
New vs. Old Sets

- Seki-Kaneko [2000]:
  \[0x70707070,0x07070707\]
  \[2^{24}-1\] differences
  24 active bits
  naturally occurs: \(2^{-40}\)

- Courtois-Misztal [2011]
  \(0x80700700,0x80700700\)
  \(2^{14}-1\) differences
  14 active bits
  naturally occurs: \(2^{-50}\)
### New Sets [Courtois, Misztal, 2011]

<table>
<thead>
<tr>
<th>Input Aggregated Differential</th>
<th>0x70707070,0x07070707</th>
<th>0x80700700,0x80700700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Aggregated Differential</td>
<td>0x70707070,0x07070707</td>
<td>0x80700700,0x80700700</td>
</tr>
<tr>
<td>Reference</td>
<td>Seki-Kaneko [38]</td>
<td>this paper and [10]</td>
</tr>
<tr>
<td>Propagation 2 R</td>
<td>$2^{-8.6}$</td>
<td>$2^{-7.5}$</td>
</tr>
<tr>
<td>Propagation 4 R</td>
<td>$2^{-16.7}$</td>
<td>$2^{-13.6}$</td>
</tr>
<tr>
<td>Propagation 6 R</td>
<td>$2^{-24.1}$</td>
<td>$2^{-18.7}$</td>
</tr>
<tr>
<td>Propagation 8 R</td>
<td>$2^{-28.4}$</td>
<td>$2^{-25.0}$</td>
</tr>
<tr>
<td>Propagation 10 R</td>
<td>$2^{-35}$</td>
<td>$2^{-31.1}$</td>
</tr>
<tr>
<td>Propagation 12 R</td>
<td>$2^{-43}$</td>
<td>$2^{-36}$</td>
</tr>
<tr>
<td>Propagation 14 R</td>
<td>$2^{-50}$</td>
<td>$2^{-42}$</td>
</tr>
<tr>
<td>Propagation 16 R</td>
<td>$2^{-56}$</td>
<td>$2^{-48}$</td>
</tr>
<tr>
<td>Propagation 18 R</td>
<td>$2^{-62}$</td>
<td>$2^{-54}$</td>
</tr>
<tr>
<td>Propagation 20 R</td>
<td>$2^{-70}$</td>
<td>$2^{-60}$</td>
</tr>
<tr>
<td>Propagation 22 R</td>
<td>$2^{-77}$</td>
<td>$2^{-66}$</td>
</tr>
<tr>
<td>Output $\Delta$ Occurs Naturally</td>
<td>$2^{-40.0}$</td>
<td>$2^{-50.0}$</td>
</tr>
</tbody>
</table>
8.4.
AC Reduction+DC Attacks
Combined DC+Algebraic Complexity Reduction

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

<table>
<thead>
<tr>
<th>rounds</th>
<th>values/differences</th>
<th>key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$A \leftarrow 80700700 \ 80700700 \rightarrow B$</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>$A \leftarrow 80700700 \ 80700700 \rightarrow B$</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>$A \leftarrow 80700700 \ 80700700 \rightarrow B$</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>$A \leftarrow 80700700 \ 80700700 \rightarrow B$</td>
<td>256</td>
</tr>
<tr>
<td>bits 64</td>
<td></td>
<td>64</td>
</tr>
</tbody>
</table>
9. Multiple-Point Events and Bicliques
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.
Fig. 18. An approximate fixed point biclique with $k = 4$
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…
9.2. Summary: All Single+Multiple Key Attacks
The Multiple Key Scenario (1)

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

<table>
<thead>
<tr>
<th>Attack Ref.</th>
<th>§10.3/[32]</th>
<th>§13.1/[32]</th>
<th>Red. 3 §12</th>
<th>[27]</th>
<th>F.0 [54]</th>
<th>Fam. 2</th>
<th>Fam. 2</th>
<th>Fam. 3</th>
<th>Fam. 4.X.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keys density $d$</td>
<td>0.63</td>
<td>0.63</td>
<td>1</td>
<td></td>
<td>$2^{-32}$</td>
<td>$2^{-64}$</td>
<td>$2^{-64}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data/key 32R</td>
<td>$2^{32}$ KP</td>
<td>$2^{54}$ KP</td>
<td>$2^{64}$ KP</td>
<td>$2^{64}$ KP</td>
<td>$2^{32}$ CP</td>
<td>$2^{32}$ CC</td>
<td>$2^{32}$ ACC</td>
<td>$2^{64}$ KP</td>
<td>$2^{32}$ CP/2^{64}</td>
</tr>
<tr>
<td>Obtained for 8R</td>
<td>2 KP</td>
<td>3 KP</td>
<td>-</td>
<td>1 KP</td>
<td>3 KP</td>
<td>4 KP</td>
<td>2 KP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid w. prob.</td>
<td>$2^{-96}$</td>
<td>$2^{-64}$</td>
<td>$2^{-64}$</td>
<td>-</td>
<td>$2^{-1}$</td>
<td>$2^{-64}$</td>
<td>$2^{-64}$</td>
<td>$2^{-1}$</td>
<td>$2^{-0}$</td>
</tr>
<tr>
<td>Storage bytes</td>
<td>$2^{46}/2^{39}$</td>
<td>$2^{46}/2^{39}$</td>
<td>$2^{57}$</td>
<td>$2^{0}$</td>
<td>small</td>
<td></td>
<td></td>
<td>$2^{57}$</td>
<td>for data</td>
</tr>
<tr>
<td># False positives</td>
<td>$2^{128}$</td>
<td>$2^{128}$</td>
<td>$2^{192}$</td>
<td>$2^{64}$</td>
<td>$2^{-0}$</td>
<td>$2^{64}$</td>
<td>$2^{128}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time for 8 R</td>
<td>$2^{127}/2^{128}$</td>
<td>$2^{127}/2^{128}$</td>
<td>$2^{110}$</td>
<td>-</td>
<td>$2^{192}$</td>
<td>$2^{110}$</td>
<td>$2^{94}$</td>
<td>$2^{94}$</td>
<td>$2^{128}$</td>
</tr>
<tr>
<td>Attack time 32 R</td>
<td>$2^{223}/2^{224}$</td>
<td>$2^{191}/2^{192}$</td>
<td>$2^{206}$</td>
<td>$2^{179}$</td>
<td>$2^{192}$</td>
<td>$2^{174}$</td>
<td>$2^{158}$</td>
<td>$2^{95}$</td>
<td>$2^{128}$</td>
</tr>
<tr>
<td>Cost of 1 key, if key diversity ≥ single key attacks or for &gt; 50% of keys</td>
<td>$2^{224}/2^{225}$</td>
<td>$2^{192}/2^{193}$</td>
<td>$2^{207}$</td>
<td>$2^{179}$</td>
<td>$2^{193}$</td>
<td>$2^{206}$</td>
<td>$2^{190}$</td>
<td>$2^{159}$</td>
<td>$2^{129}$</td>
</tr>
</tbody>
</table>

© Nicolas T. Courtois, 2006-2013
### 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.
9.3.
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


A. Dmukh, V. Rudskoy

Easy: try CryptoMiniSat

See Cryptologia Jan 2013 and eprint/2011/626

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...
10. 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\textsuperscript{1,*} and A. Poschmann\textsuperscript{2}

\textsuperscript{1} GRIM, University Toulon, France
Gregor.Leander@rub.de

\textsuperscript{2} Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum, Germany
poschmann@crypto.rub.de
Affine Equivalence

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

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

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

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

Only 16 S-boxes are “good”.

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

<table>
<thead>
<tr>
<th>S-box Set Name</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
</tr>
</thead>
<tbody>
<tr>
<td>GostR3411_94_TestParamSet</td>
<td>36</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>06</td>
<td>35</td>
<td>08</td>
<td></td>
</tr>
<tr>
<td>- their inverses</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>06</td>
<td>08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GostR3411_94_CryptoProParamSet</td>
<td>Lu1</td>
<td>14</td>
<td>G10</td>
<td>G8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- their inverses</td>
<td>Lu1</td>
<td>14</td>
<td>G10</td>
<td>G8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gost28147_TestParamSet</td>
<td>21</td>
<td>21</td>
<td>25</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- their inverses</td>
<td>21</td>
<td>21</td>
<td>25</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetA</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>G8</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>- their inverses</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>G8</td>
<td>37</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetB</td>
<td>G13</td>
<td>G13</td>
<td>G13</td>
<td>G11</td>
<td>G7</td>
<td>G7</td>
<td>G11</td>
<td>G6</td>
</tr>
<tr>
<td>- their inverses</td>
<td>G13</td>
<td>G13</td>
<td>G13</td>
<td>G11</td>
<td>G7</td>
<td>G7</td>
<td>G11</td>
<td>G6</td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetD</td>
<td>G13</td>
<td>G13</td>
<td>G13</td>
<td>G12</td>
<td>G12</td>
<td>G13</td>
<td>G7</td>
<td></td>
</tr>
<tr>
<td>- their inverses</td>
<td>G13</td>
<td>G13</td>
<td>G13</td>
<td>G12</td>
<td>G12</td>
<td>G13</td>
<td>G7</td>
<td></td>
</tr>
<tr>
<td>GostR3411_94_SberbankHashParamset</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- their inverses</td>
<td>74</td>
<td>75</td>
<td>78</td>
<td>76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOST ISO 18033-3 proposal</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
</tr>
<tr>
<td>- their inverses</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
<td>G9</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>S-box Set Name</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
</tr>
</thead>
<tbody>
<tr>
<td>GostR3411_94_TestParamSet</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>GostR3411_94_CryptoProParamSet</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Gost28147_TestParamSet</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetA</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetB</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetC</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Gost28147_CryptoProParamSetD</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>GostR3411_94_SberbankHashParamset</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>GOST ISO 18033-3 proposal</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
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. Diffusion in GOST

Guess-Then-Determine

UNSAT Immunity
*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…
1 Round + Next Round of GOST
Carry Propagation

determine \( a \):

need \( S3, S4 \) and \( c \)

\[
\begin{array}{ccc}
3 & 1 & 1 \\
\end{array}
\]

d, e known

\[ \Rightarrow 2^{0.6} \] possibilities

3 more bits known

\[ \Rightarrow 2^{0.3} \] possibilities

\[ \ldots \]

\[ 2^{0.0} \]
11.2. Guess-Then-Determine: What to Guess?
11.2.1. 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!
Fact 1. The Contradiction Immunity is at most 44 for 8 rounds of DES.

GOST?
UNSAT Immunity

Well chosen set of 68 bits. [cf. Tatracrypt 2012]

UNSAT proba=39%.
Jumps…

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

UNSAT proba=50%.
Conclusion:

For 8 rounds of GOST:
The UNSAT Immunity is at most 78

[Tatracrypt 2012]
More on UNSAT Immunity

See:

Nicolas Courtois, Jerzy A. Gawinecki, Guangyan Song: *Contradiction Immunity and Guess-Then-Determine Attacks On GOST*,
SAT Immunity – 4 pairs

Same set of 68 bits as before.

=> all the other bits?
SAT Immunity – 4 pairs

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.

Corollary: Given 4KP for 8R we determine all the key bits in time $2^{94}$.

[Courtois Cryptologia vol 37, 2013]
### Principal Attacks on 8 Rounds

<table>
<thead>
<tr>
<th>Rounds</th>
<th>8</th>
<th>8</th>
<th>8</th>
<th>8</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key size</td>
<td>256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>2 KP</td>
<td>3 KP</td>
<td>4 KP</td>
<td>6 KP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cf.</td>
<td>Fact 112 page 152</td>
<td>page 27</td>
<td>page 27</td>
<td>page 163</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cf. also</td>
<td>Fact 113</td>
<td>Fact 115</td>
<td>page 163</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory bytes</th>
<th>$2^{43}$</th>
<th>$2^{40}$</th>
<th>$2^{98}$</th>
<th>small</th>
<th>$2^{99}$</th>
<th>small</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$2^{128}$</td>
<td>$2^{127}$</td>
<td>$2^{117}$</td>
<td>$2^{110}$</td>
<td>$2^{94}$</td>
<td>$2^{94}$</td>
</tr>
</tbody>
</table>

**Table 1.** Principal attacks on 8 rounds of GOST with 2, 3, 4 and 6 KP

Shamir et al. FSE 2012

Cryptologia vol 37, 2013

2011/626
*12. Another take on it: Inference, Induction, Saturation Analysis

or how “order” emerges in various attacks
Overcoming Chaos

inputs

outputs

data??? key???
Add Information => Amplify => Solve

more constraints => saturation?
Growth Leads To Saturation

Add Info => Emerging Order => Inference Saturation
determine all

2x Amplification

phase transition hard=>easy
12.1. SAT Immunity

\[ \geq 1KP \]
SAT Immunity – 4 pairs

Same set of 68 bits as before.

=> all the other bits?
SAT Immunity – 4 pairs

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.

Corollary: Given 4KP for 8R we determine all the key bits in time $2^{94}$.

[Courtois Cryptologia vol 37, 2013]
12.2. Inference With 1KP

yes, due to the key schedule
First 16 Rounds of GOST

1 point, first 16 rounds of GOST

\[ A \]

becomes

\[ P \approx 50\% \]

\[ A \]

1 bit

A is an arbitrary unknown value

Fig. 29. Fixed points in the first 16 rounds of GOST seen as an Induction property: the value in the middle is obtained nearly for free instead of \(2^{-64}\)
Compare to Last 16 Rounds:

1 point, LAST 16 rounds of GOST

A

(16 Rounds)

becomes

100 %

(8 Rounds)

A

0 bit

A is an arbitrary unknown value, many solutions

A

(8 Rounds)

32 bits
12.3. Differential Induction: 2, 3, 4 KP
Differential Induction = Two-Sided Propagation
Any 16 Rounds of GOST

2 points

A, B sharing 50 bits
0x80700700 0x80700700

(16 Rounds) becomes 50 %

0x80700700 0x80700700
C, D sharing 50 bits

50+50 bits
2^28 events

A, B sharing 50 bits
0x80700700 0x80700700

(8 Rounds)

0x80700700 0x80700700
C, D sharing 50 bits

150 bits
2^27 events

Fig. 31. Differential Induction: 50 additional differences nearly for free instead of 2^-50
20 Rounds

2 points

A, B sharing 63 bits
0x80000000 0x00000000

(20 Rounds) becomes 50 %

0x00000000 0x80000000
C, D sharing 63 bits

128 bits
2^-1 events

A, B sharing 63 bits
0x80000000 0x00000000
(10 Rounds)

0x80700700 0x80700700
(10 Rounds)

0x00000000 0x80000000
C, D sharing 63 bits

178 bits
2^-2 events

more rounds requires stronger I/O constraints
3 points make it quite strong
13. DC Primer

COMP128v1
DES,
GOST
GOST vs. DC

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

“Against DC and LC, GOST is probably stronger than DES”

Gabidulin 2000-2001:
7 rounds are sufficient to protect GOST against DC.

13.1. 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 $\Delta$ 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

\[ x = \text{RAND} \]

5-round compression

\[ F_K : 128 \rightarrow 128 \]

Derive new x
Weakness: “All Zero Output Difference”

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

![Diagram showing the structure of a block cipher with two rounds, involving S1 and S2 functions.](image-url)
“All Zero Output Difference” for DES?

Impossible for bijective functions.

The best we can hope: reproduction of small HW pattern $\Delta$. 
“All Zero Output Difference” for Round Functions

Possible for DES: not bijective.
Not easy
(3 or more boxes).

Impossible for GOST: bijective.
13.2. 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…

\[ \Delta = 0 \]

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).
13.3. DC on DES
DES:
Figure 1: An example of Differential Cryptanalysis
13.4.

Law Of Small Numbers = Poisson Distributions
Many events follow the Poisson distribution. Process without memory, like a lottery. Sometimes an event happens. Example: bitcoin mining. Every XXX minutes a block is created. Some are ours.

**Question:** if in 36 days I’ve produced #events=K blocks, what is the standard deviation?

**Answer:** \( \text{StDev} = \sqrt{K} \)

This what makes the miners mine in pools, e.g. if \( K=4 \), \( \sqrt{K} = 2 \)

variation of order of 50% on your monthly pay are really unbearable…

The same law applied to differential attacks and nearly any sort of attacks based of ‘events’ of some sort.

If some events happen: Events = \( 2^5 \) for a cipher

\( \text{StDev} = 2^{2.5} \)
****Application: Breaking AES with MD5

Breaking AES as a permutations generator with MD5 as a distinguisher

[Bernstein-Lange 2012] - NOT A JOKE!

Idea:

- AES
- AES
- RP
- RP
- vary k
- 128

pairs (x, x') on 256 bits

ONLY $2^{128}$ cases exist uniformly spans $2^{256} - 2^{128}$ points

MD5

output first bit only

count 1's

256

1

256

1

much larger sample => far smaller bias = 0

Standard distinguisher advantage in cryptography is:

$$Adv = |P(\text{Left MD5 bit } 0 \rightarrow 1) - P(\text{Right MD5 bit } 0 \rightarrow 1)|$$

Theorem: $Adv \geq 2^{-64.5}$ with proba 68%. $Adv \geq 2^{-65.5}$ with proba 95%.

NOT true that $Adv = 2^{-128}$ !!!!

Proof: Poisson distribution again: we expect $2^{127}$ events such that Left MD5 bit $0 \rightarrow 1$.

Then $\text{StDev} = 2^{63.5}$. The bias is about $2^{-64.5}$.

Inside 1 stdev with proba 68%. 2 stdev with proba 95%. 3 stdev with proba 99.7%. Etc…

Important remark: we still need time $\approx 2^{128}$ in order to “observe” such a bias!
13.5.
Classical DC
or How to Get Misled
13.5.1. History of DC
History of DC

New paper to appear soon:
History of DC

Differential Cryptanalysis (DC)

• based on tracking of changes in the differences between two messages as they pass through the consecutive rounds of encryption.

• one of the oldest classical attacks on modern block ciphers, if not the oldest.

• ALL ciphers should resist it…
History of DC

Coppersmith [IBM DES design team] have reported that this attack was already known to IBM designers around 1974. It was known under the name of T-attack or Tickle attack. It appears that

- DES have already been designed to resist to this type of attack
- IBM have agreed with the NSA that the design criteria of DES should not be made public. This precisely because it would “weaken the competitive advantage the United States enjoyed over other countries in the field of cryptography”
13.5.2.
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 $E(\text{event}) = 2^{64-Y}$ for a RP.
Propagation: $E(\text{event'}) = 2^{64-Y} + 2^{64-X}$ for XXX rounds.
Distinguishing between two Gaussian distributions.
  Q: How many standard deviations?
    Right key with proba? <= Gauss error function.
    StDev $= 2^{(64-Y)/2}$ => it is sufficient to obtain $2^{64-X} = C \times 2^{(64-Y)/2}$. Data Cxty follows.
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.

32 rounds, $2^{179}$
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 \( \frac{1}{2} \) 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.
14. Advanced DC
14.1. DC With Sets
Truncated Properties
Combinatorial Exploration
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 → 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 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 $1-3.5/16 = 2^{-0.36}$.
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 $2^{-28.4}$. 

© Nicolas T. Courtois, 2006-2013
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,
   => invention of new sets
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]

<table>
<thead>
<tr>
<th>Input Aggregated Differential</th>
<th>Output Aggregated Differential</th>
<th>Reference</th>
<th>Seki-Kaneko [38]</th>
<th>this paper and [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x70707070,0x07070707</td>
<td>0x80700700,0x80700700</td>
<td>Propagation 2 R</td>
<td>2^{-8.6}</td>
<td>2^{-7.5}</td>
</tr>
<tr>
<td>Propagation 4 R</td>
<td></td>
<td>2^{-16.7}</td>
<td></td>
<td>2^{-13.6}</td>
</tr>
<tr>
<td>Propagation 6 R</td>
<td></td>
<td>2^{-24.1}</td>
<td></td>
<td>2^{-18.7}</td>
</tr>
<tr>
<td>Propagation 8 R</td>
<td></td>
<td>2^{-28.4}</td>
<td></td>
<td>2^{-25.0}</td>
</tr>
<tr>
<td>Propagation 10 R</td>
<td></td>
<td>2^{-35}</td>
<td></td>
<td>2^{-31.1}</td>
</tr>
<tr>
<td>Propagation 12 R</td>
<td></td>
<td>2^{-43}</td>
<td></td>
<td>2^{-36}</td>
</tr>
<tr>
<td>Propagation 14 R</td>
<td></td>
<td>2^{-50}</td>
<td></td>
<td>2^{-42}</td>
</tr>
<tr>
<td>Propagation 16 R</td>
<td></td>
<td>2^{-56}</td>
<td></td>
<td>2^{-48}</td>
</tr>
<tr>
<td>Propagation 18 R</td>
<td></td>
<td>2^{-62}</td>
<td></td>
<td>2^{-54}</td>
</tr>
<tr>
<td>Propagation 20 R</td>
<td></td>
<td>2^{-70}</td>
<td></td>
<td>2^{-60}</td>
</tr>
<tr>
<td>Propagation 22 R</td>
<td></td>
<td>2^{-77}</td>
<td></td>
<td>2^{-66}</td>
</tr>
<tr>
<td>Output ∆ Occurs Naturally</td>
<td></td>
<td>2^{-40.0}</td>
<td></td>
<td>2^{-50.0}</td>
</tr>
</tbody>
</table>
Type 3+3: S836 + S836
14.3. Truncated Diff. Propagation
How To Find Such An Attack

Best differential property we ever found was found BY HAND.

Is a systematic approach possible?
Our Attack = Graph Walks With Costs
Figure 3: Propagation of (800000000, 000000000) after 7R
and 8R – still concentrates at few places

Figure 4: Propagation of \((80000000, 00000000)\) after 8R
Low Entropy!

Figure 5: The Entropy estimation and plot after 1-7 rounds of GOST starting from the input set 8000000000000000

<table>
<thead>
<tr>
<th>Round</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>2.81</td>
</tr>
<tr>
<td>3</td>
<td>5.61</td>
</tr>
<tr>
<td>4</td>
<td>5.72</td>
</tr>
<tr>
<td>5</td>
<td>8.19</td>
</tr>
<tr>
<td>6</td>
<td>10.92</td>
</tr>
<tr>
<td>7</td>
<td>12.31</td>
</tr>
</tbody>
</table>

14 for RP
Remark:

• the structure of this graph does NOT depend on the S-boxes
• only costs (probabilities) depend on the S-boxes and not always a lot
14.4.1. Truncated Differentials
As Collisions and Statistical Tests
“Truncated Differentials” == Double Collisions

Proposed as “collision tests” in:

Cryptographic Randomness Testing of Block Ciphers and Hash Functions

eprint.iacr.org/2010/564

Ali Doğanaksoy, Barış Ege, Onur Koçak and Fatih Sulak
For GOST

\[ P \xrightarrow{\text{2 rounds}} P', \text{identical on 50 bits} \]

\[ C \xleftarrow{\text{2 rounds}} C', \text{identical on 50 bits} \]

\[ 0x80700700, 0x80700700 \]

Cf.
What is Wrong?

WRONG approach, or WRONG philosophy, or at least wrong vocabulary…
What is Wrong?

\[
\begin{align*}
P & \xleftarrow{k} P' \\
C & \xleftarrow{k} C'
\end{align*}
\]

identical on 50 bits

\[
\begin{align*}
P & \xleftarrow{2 \text{ rounds}} P' \\
C & \xleftarrow{2 \text{ rounds}} C'
\end{align*}
\]

identical on 50 bits

**NOT** a property to be TESTED at random (average case random testing).
Efficient Testing vs. Painful Discovery
Painful Discovery!

\[ P \rightarrow P' \quad \text{identical on 50 bits} \]

\[ C \rightarrow C' \quad \text{identical on 50 bits} \]

**NOT** a property to be TESTED for.
This property must be studied as the BEST case.
Can be difficult to find even if it exists.
Moreover, size matters! As we will see later…
14.4.2. Existence of Interesting Attacks
Philosophy (1)

One perturbation is always diffused in DC.

"diffusion cone"

more rounds ➔ more active bits

ho hope it would be smaller…
Philosophy (2)

Can several perturbations converge somewhat? Like larger “channel capacity”.

© Nicolas T. Courtois, 2006-2013
Philosophy (3)

Not if we have TOO many sources!
Must **restrict** the input diversity.
14.4.3. From Existence to Discovery
Black Box Methods

Random guessing + several feedback/learning loops (evolutionary algorithm):

• flip few bits
• extend size
• decrease size
• use repeated “patterns” seen
• etc…
### Some Results – 19 bits,
“very good for 8R and good for 12R”

<table>
<thead>
<tr>
<th>S-box Set Name</th>
<th>Truncated differential set S</th>
<th>$P([S] \rightarrow [S])$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8R</td>
</tr>
<tr>
<td>0 GostR3411_94_TestParamSet</td>
<td>78001078 07070780</td>
<td>$2^{-24.9}$</td>
</tr>
<tr>
<td>1 GostR3411_94_CryptoProParamSet</td>
<td>08070780 78788030</td>
<td>$2^{-24.4}$</td>
</tr>
<tr>
<td>2 Gost28147_TestParamSet</td>
<td>84000707 E0787200</td>
<td>$2^{-23.6}$</td>
</tr>
<tr>
<td>3 Gost28147_CryptoProParamSetA</td>
<td>78780820 00070707</td>
<td>$2^{-25.2}$</td>
</tr>
<tr>
<td>4 Gost28147_CryptoProParamSetB</td>
<td>80707820 07000787</td>
<td>$2^{-25.9}$</td>
</tr>
<tr>
<td>5 Gost28147_CryptoProParamSetC</td>
<td>78780080 80070707</td>
<td>$2^{-25.5}$</td>
</tr>
<tr>
<td>6 Gost28147_CryptoProParamSetD</td>
<td>84000787 70707800</td>
<td>$2^{-25.4}$</td>
</tr>
<tr>
<td>7 GostR3411_94_SberbankHash</td>
<td>90000607 D4787800</td>
<td>$2^{-24.9}$</td>
</tr>
<tr>
<td>8 GOST ISO 18033-3 proposal</td>
<td>80000707 F0787800</td>
<td>$2^{-23.8}$</td>
</tr>
<tr>
<td>9 GOST-P proposal</td>
<td>F0707000 07000707</td>
<td>$2^{-27.0}$</td>
</tr>
</tbody>
</table>
14.5. Discovery of OPTIMAL Size [and Shape]


+new paper:
## Different Sizes

“very good for 8R”

<table>
<thead>
<tr>
<th>$a$</th>
<th>GOST S-box Set Name</th>
<th>Truncated differential set S</th>
<th>$P([S] \rightarrow [S])$ $8R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>F0780780 F0070781</td>
<td>$2^{-28.3}$</td>
</tr>
<tr>
<td>21 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>78780000 F0070783</td>
<td>$2^{-26.5}$</td>
</tr>
<tr>
<td>19 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>78001078 07070780</td>
<td>$2^{-24.9}$</td>
</tr>
<tr>
<td>17 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>D0707000 80000787</td>
<td>$2^{-23.7}$</td>
</tr>
<tr>
<td>15 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>80707800 80000707</td>
<td>$2^{-22.9}$</td>
</tr>
<tr>
<td>14 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>80707800 80000307</td>
<td>$2^{-22.6}$</td>
</tr>
<tr>
<td>12 0</td>
<td>GostR3411_94_TestParamSet</td>
<td>80707800 80000007</td>
<td>$2^{-22.8}$</td>
</tr>
<tr>
<td>9  0</td>
<td>GostR3411_94_TestParamSet</td>
<td>80700780 80000000</td>
<td>$2^{-25.2}$</td>
</tr>
<tr>
<td>24 3</td>
<td>Gost28147_CryptoProParamSetA</td>
<td>F0770700 F0700708</td>
<td>$2^{-31}$</td>
</tr>
<tr>
<td>21 3</td>
<td>Gost28147_CryptoProParamSetA</td>
<td>78780060 80070787</td>
<td>$2^{-25.4}$</td>
</tr>
<tr>
<td>19 3</td>
<td>Gost28147_CryptoProParamSetA</td>
<td>78780820 00070707</td>
<td>$2^{-25.2}$</td>
</tr>
<tr>
<td>17 3</td>
<td>Gost28147_CryptoProParamSetA</td>
<td>03070780 78008070</td>
<td>$2^{-24.2}$</td>
</tr>
<tr>
<td>14 3</td>
<td>Gost28147_CryptoProParamSetA</td>
<td>70780000 80030780</td>
<td>$2^{-23.8}$</td>
</tr>
<tr>
<td>12 3</td>
<td>Gost28147_CryptoProParamSetA</td>
<td>70780000 80080700</td>
<td>$2^{-26.7}$</td>
</tr>
</tbody>
</table>

© Nicolas T. Courtois, 2006-2013
### Comparison With DES

“very good for 4R”

<table>
<thead>
<tr>
<th>(a)</th>
<th>Reference Cipher</th>
<th>Truncated differential set (S)</th>
<th>(P([S] \rightarrow [S]))</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>U.S. Data Encryption Standard</td>
<td>D8081040 85308C06</td>
<td>(2^{-21.0})</td>
</tr>
<tr>
<td>14</td>
<td>U.S. Data Encryption Standard</td>
<td>10001040 85118C26</td>
<td>(2^{-18.8})</td>
</tr>
<tr>
<td>12</td>
<td>U.S. Data Encryption Standard</td>
<td>80521890 04200802</td>
<td>(2^{-18.8})</td>
</tr>
<tr>
<td>11</td>
<td>U.S. Data Encryption Standard</td>
<td>A05000B0 04200802</td>
<td>(2^{-16.4})</td>
</tr>
<tr>
<td>10</td>
<td>U.S. Data Encryption Standard</td>
<td>00802080 0C080A0A</td>
<td>(2^{-16.8})</td>
</tr>
<tr>
<td>9</td>
<td>U.S. Data Encryption Standard</td>
<td>08020000 80F88000</td>
<td>(2^{-16.7})</td>
</tr>
<tr>
<td>8</td>
<td>U.S. Data Encryption Standard</td>
<td>08020000 80B88000</td>
<td>(2^{-16.6})</td>
</tr>
<tr>
<td>7</td>
<td>U.S. Data Encryption Standard</td>
<td>08020000 80B80000</td>
<td>(2^{-16.8})</td>
</tr>
<tr>
<td>6</td>
<td>U.S. Data Encryption Standard</td>
<td>4000008B 00040000</td>
<td>(2^{-17.3})</td>
</tr>
<tr>
<td>5</td>
<td>U.S. Data Encryption Standard</td>
<td>005000B0 14200800</td>
<td>(2^{-18.1})</td>
</tr>
</tbody>
</table>
### Another Block Cipher

<table>
<thead>
<tr>
<th>$D$</th>
<th>Reference Cipher</th>
<th>Truncated differential set $S$</th>
<th>$P([S] \rightarrow [S])$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>TEA = Tiny Encryption Algorithm</td>
<td>008008A0 81009111</td>
<td>$2^{-13}$</td>
</tr>
<tr>
<td>10</td>
<td>TEA = Tiny Encryption Algorithm</td>
<td>00010012 90001131</td>
<td>$2^{-11}$</td>
</tr>
<tr>
<td>9</td>
<td>TEA = Tiny Encryption Algorithm</td>
<td>80600034 00800101</td>
<td>$2^{-11}$</td>
</tr>
<tr>
<td>8</td>
<td>TEA = Tiny Encryption Algorithm</td>
<td>4A000112 01000001</td>
<td>$2^{-11}$</td>
</tr>
<tr>
<td>6</td>
<td>TEA = Tiny Encryption Algorithm</td>
<td>00200020 001000A1</td>
<td>$2^{-13}$</td>
</tr>
</tbody>
</table>
**Some Results – 14 bits,**

“very good for 8R”

Table 1: Some recent results with sets of 14 bits and 8 rounds cf. [10]

<table>
<thead>
<tr>
<th>Set Name</th>
<th>Set</th>
<th>$P(8R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>default set [21]</td>
<td>78000078 07070780</td>
<td>$2^{-24.0}$</td>
</tr>
<tr>
<td>ISO 18033-3 proposal</td>
<td>80000707 20707000</td>
<td>$2^{-22.7}$</td>
</tr>
</tbody>
</table>
Best Differential Property of Any Size and Shape!
Best Differential for GOST = 14 Bits

⇒ 14 bit properties discovered earlier:


can be shown to be optimal!
⇒ 24 cannot be good
14.6.
DC -> Distinguishers
And 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

© Nicolas T. Courtois, 2006-2013
New Attacks

References:

   => invention of new sets

   => first simple attack (very slightly) faster than brute force $2^{254.6}$

   => progressive improved approach, heuristic and not very precise… $2^{226}$

   => symmetric + many further refinements + very careful work on individual bits + tight [barely working] distinguishers + justification of earlier results $2^{179}$


   => discovery that size matters in advanced differential attacks, work on various sets of S-boxes, search for optimal properties

© Nicolas T. Courtois, 2006-2013
Figure 6: The Alpha Property
Key Principles

Figure 6: The Alpha Property
### Key Principles

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Cipher Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xFFFFFFFF</td>
<td>0x00000700 0x80780000</td>
</tr>
<tr>
<td>(1 Round)</td>
<td>(1 Round)</td>
</tr>
<tr>
<td>0xFFFFFFFF</td>
<td>0x80780000 0xF0000787</td>
</tr>
<tr>
<td>(1 Round)</td>
<td>(1 Round)</td>
</tr>
<tr>
<td>0xFFFFFFFF</td>
<td>0x80780000 0xFF8787</td>
</tr>
<tr>
<td>(20 Rounds)</td>
<td>(1 Round)</td>
</tr>
<tr>
<td>0x80780000 0xF0000787</td>
<td>0x80780000 0xF0000787</td>
</tr>
<tr>
<td>(1 Round)</td>
<td>(1 Round)</td>
</tr>
<tr>
<td>0xFF8787 0x80780000</td>
<td>0xFF8787 0xFF8787</td>
</tr>
<tr>
<td>(1 Round)</td>
<td>(1 Round)</td>
</tr>
<tr>
<td>0xFF8787 0x80780000</td>
<td>0xFF8787 0xFF8787</td>
</tr>
<tr>
<td>(1 Round)</td>
<td>(1 Round)</td>
</tr>
<tr>
<td>0x80780000 0x00000700</td>
<td>0x80780000 0x00000700</td>
</tr>
<tr>
<td>(1 Round)</td>
<td>(1 Round)</td>
</tr>
</tbody>
</table>

**Figure 6: The Alpha Property**

- Unconstrained propagation, high proba!
- Constrained at 2 ends, arbitrary inside

© Nicolas T. Courtois, 2006-2013
14.6.2. Best Symmetric Result for 20 R (best known)
Propagation - Middle 20 Rounds

Propagation with probability???
What is Propagation???

0x80780000 0x00000700

20 rounds
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: $2^{15}$.  

\[
\begin{align*}
0x80780000 & \quad 0x00000700 \\
0x80700700 & \quad 0x80700700 \\
0x80700700 & \quad 0x80700700 \\
0x00000700 & \quad 0x80780000
\end{align*}
\]
14.7.
Distinguishers For 20R
**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 $2^{6.5}$.

---

**Figure 6: The Alpha Property**
Gauss Error

How many standard deviations?

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.
\]

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

Example: right key assumption rejected

= half of this number
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 $\text{Max}(2^{13}, 2^{11.9})$.
- 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 $\text{Max}(2^{15}, 2^{13.9})$.
- For more rounds we get $2^{15} + 2^{13.9}$.

We make an “artificial distinction” assumption which separates the two sets!

$20 = 7 + 6 + 7$
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.

Natural: $2^{15}$ XX rounds
Separation Failure: Less Rounds

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

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

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

+ 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.

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

With more rounds no reinforcement.
14.8. Building Distinguishers
New Paper [2013]

References:
1. Nicolas Courtois, Theodosis Mourouzis: 
   Enhanced Truncated Differential Cryptanalysis of GOST,
   => important notion: general open sets, more robust!
   => building detailed distinguishers

Summary:
We have refined the Knudsen general Truncated Differential attack and partition it into disjoint sets such that different parts of these sets are better defined and they give results which are more stable, more predictable and more ‘composable’.
Construction

Figure 6: Representation of construction of a general distinguisher for \( n \) rounds seen as a combinatorial problem.
Example – set0 (default S-boxes)

Result 5.1.1:

\[
\begin{align*}
8780070780707000 \\
\downarrow (10R) \\
[8070070080700700] \\
\downarrow (10R) \\
8070700087800707
\end{align*}
\]

is a 20 rounds distinguisher for this variant of GOST.

Justification: For a typical permutation on 64-bits (does not have to be a random permutation, it can be GOST with more rounds) we expect that there are \(2^{27.1}\) pairs \((P_i, P_j)\) with such differences. The distribution of this number can be approximated by a Gaussian with a standard deviation \(2^{13.55}\).

For 20 rounds of GOST and for a given random GOST key, there exists two disjoint sets of \(2^{27.1} + 2^{18.2}\) such pairs \((P_i, P_j)\).
\[ \text{set3} = \text{Gost28147-CryptoProParamSetA} \]

**Result 5.2.1:**

\[
\begin{align*}
07700700777777770 \\
\uparrow(10R) \\
[7007070070070700] \\
\downarrow(10R) \\
7777777007700700 \\
\end{align*}
\]

is a 20 rounds distinguisher for this variant of GOST, where \([7007070070070700]\) is a closed set.

**Justification:** For a typical permutation on 64-bits (does not have to be a random permutation, it can be GOST with more rounds) we expect that there are \(2^{55.1}\) pairs \((P_i, P_j)\) with such differences. The distribution of this number can be approximated by a Gaussian with a standard deviation \(2^{27.55}\).

For 20 rounds of GOST and for a given random GOST key, there exists two disjoint sets of \(2^{55.1} + 2^{33.0}\) such pairs \((P_i, P_j)\).
14.9.
Distinguishers =>$ \Rightarrow$ Key Recovery
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.

Level 2: guess more key bits, eliminate cases.

\[
2^{171} \quad 2^{171} \quad 2^{171} \quad 2^{171}
\]

\[
2^{175} \quad 2^{175} \quad 2^{175}
\]

\[
2^{55} \text{ cases per key}
\]

\[
2^{31} \text{ cases per key}
\]
Much Later:

Level 1:

116 bits

$2^{171}$

Level 2:

116+28 bits

$2^{175}$

$2^{55}$ cases per key

$2^{31}$ cases per key

Level 3:

X

© Nicolas T. Courtois 2006-2013
<table>
<thead>
<tr>
<th>guess key at S-boxes</th>
<th>correct</th>
<th>difference</th>
<th>new bits to cancel after outputs of</th>
<th>after round</th>
<th>new inactive bits (60 in total)</th>
<th>enumerate cases</th>
<th>per key</th>
<th>key bits assum.</th>
<th>time GOST encrypt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>all bits in R12</td>
<td>$2^{64}$</td>
<td>FFFFF8787</td>
<td>S7,S1</td>
<td>2,31</td>
<td>8</td>
<td>4-7, 12-15</td>
<td>$2^{127}$</td>
<td></td>
<td>birthday</td>
</tr>
<tr>
<td>S3*4567R3</td>
<td>$2^{20}$</td>
<td>807FFFF80</td>
<td>S45637</td>
<td>3,30</td>
<td>15</td>
<td>24-31,1-7</td>
<td>$2^{55}$</td>
<td>116</td>
<td>$2^{174}$</td>
</tr>
<tr>
<td>S812R3 S12345R4</td>
<td>$2^{32}$</td>
<td>F0000787</td>
<td>S2345</td>
<td>4,29</td>
<td>13</td>
<td>16-28</td>
<td>$2^{55}+116$</td>
<td>116</td>
<td>$2^{174}$</td>
</tr>
<tr>
<td>S6R4 S78R5</td>
<td>$2^{12}$</td>
<td>80780000</td>
<td>S8</td>
<td>5,28</td>
<td>4</td>
<td>8-11</td>
<td>$2^{171}$</td>
<td></td>
<td>$2^{179}$</td>
</tr>
<tr>
<td>S7R4 S1R5</td>
<td>$2^{8}$</td>
<td>80780000</td>
<td>S1</td>
<td>5,28</td>
<td>4</td>
<td>12-15</td>
<td>$2^{175}$</td>
<td>136</td>
<td>$2^{174}$</td>
</tr>
<tr>
<td>S8R4 S2R5</td>
<td>$2^{8}$</td>
<td>80780000</td>
<td>S2</td>
<td>5,28</td>
<td>4</td>
<td>16-19</td>
<td>$2^{175}$</td>
<td>144</td>
<td>$2^{174}$</td>
</tr>
<tr>
<td>S3R5 S415R6</td>
<td>$2^{9}$</td>
<td>00000700</td>
<td>S5</td>
<td>6,27</td>
<td>3</td>
<td>29,30,31</td>
<td>$2^{175}+9+1.2-3-3$</td>
<td>153</td>
<td>$2^{176}$</td>
</tr>
<tr>
<td>S4R5 S6R6</td>
<td>$2^{8}$</td>
<td>00000700</td>
<td>S6</td>
<td>6,27</td>
<td>4</td>
<td>32,1-3</td>
<td>$2^{179}+8-4-4$</td>
<td>161</td>
<td>$2^{178.2}$</td>
</tr>
<tr>
<td>$2^{18.2} + 2^{11.5}$</td>
<td>is</td>
<td>chosen</td>
<td>at $2^{2.4}$</td>
<td>standard deviations $\star 2^{-24}$</td>
<td>to survive $2^{179.2-24}$ or $2^{-6}$ per key only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>except for the right</td>
<td>161 bits</td>
<td>we have to remain</td>
<td>$2^{179.2-24}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S56R5 S781R6 S23R7</td>
<td>$2^{28}$</td>
<td>00000700</td>
<td>S8</td>
<td>6,27</td>
<td>1+</td>
<td>8</td>
<td>$2^{155.2+28-5-5}$</td>
<td>$28.2\star$</td>
<td>189</td>
</tr>
<tr>
<td>$2^{8.2} + 2^{10}$</td>
<td>is</td>
<td>chosen</td>
<td>at $2^{5.9}$</td>
<td>standard deviations $-$ certitude</td>
<td>total $2^{178.6}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All Steps
16. Strange Ideas...
16.1. Amplification Paradox
Involution $\Rightarrow$ Amplification

1 pair 16 R $\Rightarrow$
another pair for free

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

$$ \Rightarrow $$

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

can we continue?

<table>
<thead>
<tr>
<th>rounds</th>
<th>values</th>
<th>key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$X \downarrow$ $\mathcal{E}$ $T \downarrow$</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>$Y \downarrow$ $\mathcal{E}$ $Z \downarrow$</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>$Q \bowtie Q$ $\mathcal{E}$ $Q \bowtie Q$</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td>$Z \uparrow$ $\mathcal{D}$ $Y \uparrow$</td>
<td>256</td>
</tr>
</tbody>
</table>

bits $64$ $64$
Bad News

continue?

\[ Y = \mathcal{E}^2(X) \]
\[ \vdash \]
\[ Enc_k(X) = \mathcal{E}^2(Dec_k(Y)) \]
\[ \vdash \]
\[ Enc_k(Dec_k(Y)) = \mathcal{E}^2(Dec_k(Enc_k(X))) \]

<table>
<thead>
<tr>
<th>rounds</th>
<th>values</th>
<th>key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>↓</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>8</td>
<td>↓</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>8</td>
<td>$\downarrow$</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>$\overline{Q} \otimes Q$</td>
<td>$\varepsilon$</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>8</td>
<td>$\uparrow$</td>
<td>$\mathcal{D}$</td>
</tr>
</tbody>
</table>

bits \[64\]
16.2. Special Keys

Can be used for GOST hash attacks?
Palindromic Keys

- There are $2^{128}$ such keys.
- GOST encryption becomes an involution.
- We have exactly $2^{32}$ fixed points for the whole GOST.
- Cost to generate one: $\frac{1}{2}$ of GOST encryption.
  - Start in the middle.

<table>
<thead>
<tr>
<th>rounds</th>
<th>1</th>
<th>8</th>
<th>9</th>
<th>16</th>
<th>17</th>
<th>24</th>
<th>25</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>keys</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 7.** A Palindromic Key schedule in GOST
16.3. Twin Points…

sort of approximate reflection property…
which holds for ALL involutions!
**Involution**

**Definition:** Let $Q$ be a permutation on $n$ bits. We call Twin Points any Pair of Points $A$, $A'$ which differ by 1 bit such that $Q(A)=A'$ and $Q(A')=A$.

**Theorem:** Let $Q$ be an involution. For every bit position $i$ there exist on average 0.63 Twin Points and $0.63 \times n$ Twin Points overall.

**Proof:** Consider the function $X \mapsto X \oplus Q(X)$. Assume PRF. Any point including $e_i$ (all 0s except 1 at position $i$) is in its image with Probability=$\left(1-\frac{1}{e}\right)$ which is 63%. Then we have $Q(A)=A \oplus e_i$ so $A,A \oplus e_i$ is a valid pair; Likewise also $A \oplus e_i,A$ is a valid pair (because $Q$ is an involution).

**Remark:** this also works for any other non-zero difference… with several ones or arbitrary… 63 % are taken and lead to pairs… Some 28 % are taken twice or more… Etc…
16.4. Fixed Points Magic

In contrast this is an EXACT reflection property… which holds for ALL involutions!
Theorem: Let \( Q \) be an involution.

The expected number of fixed points is as large as \( 2^{n/2} \) instead of \( O(1) \) in a random permutation.

Proof:

see page 596 of Philippe Flajolet, Robert Sedgewick, Analytic Combinatorics, Cambridge University Press.

=> We already had this all over the place in our works, “semi-transparent cylinder” syndrome [Courtois], in ALL of our reflection, double reflection and triple reflection attacks… (tens of our attacks fall into this Thm.)

– EXCEPT that each time we could justify why a permutation has so many fixed points, however this also holds for more or less any involution, also some obscure ones…

14.X. or 16.X
More Distinguishers

[cf. Kara and Karakoç CANS 2012]
Symmetric Fixed Points

**Definition:** points with both halves equal. There are $2^{32}$ such points exactly.

**Fact:** They will be **inherited**.
From 8 rounds to 32 rounds:

\[ Enc_k = D \circ S \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E} \]

because A is symmetric
Symmetric Fixed Points

Taken out of $2^{32}$ such points $(x,x)$.

**Theorem 1:** A RP (Random Permutation) has one (or more) symmetric fixed points with proba = $1 - (1 - 1/2^{64})^{2^{32}} = 2^{-32}$

because $\text{taylor}( (1-(x^2))^{(1/x)}, x=0, 2 ) = 1 - x + \mathcal{O}(x^2)$

**Theorem 2:** Proba that GOST has one (or more) symmetric fixed points is twice bigger = $2 \times 2^{-32}$.

Proof: Second $2^{-32}$ is for inherited sym. fixed points from the first 8 rounds.
Two Symmetric Fixed Points

Taken out of $2^{32}$ such points $(x,x)$.

**Theorem 3:** A RP has two (or more) symmetric fixed points with proba $= \frac{1}{2} \times (2^{-32})^2 = 2^{-65}$

**Theorem 4:** Proba that GOST has two (or more) symmetric fixed points is $5 \times 2^{-65}$.

**Proof:**
Consider 9 cases wrt not(Event)1 and not(Event2),
5 have proba $2^{-65}$. Others have negligible or zero probability.
17. GOST Hash
GOST Hash

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

Obligatory part of Russian national Digital Signature standard.


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.

security proof?
works the same way
collision => collision on the compression function

extra component
(not much stronger...)

© Nicolas T. Courtois, 2006-2013
Collisions on The Compression Function

Sometimes called pseudo-collisions:
Because they may use intermediate values (IV or Hi) 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!

\[ \Delta \neq 0 \quad \Rightarrow \quad f \quad \Delta = 0 \]

\[ < 2^{128} \text{ time} \]
Pre-Images on The Compression Function

given $Y$ compute $X$

$X \xrightarrow{f} Y$

$\leq 2^{255}$ time
17.1.
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
$K, L$ are linear

$C_1, C_2$ are constants
GOST, Self-Similarity and Cryptanalysis of Block Ciphers

CICO

\[ x \in X \]

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

\[ y \in Y \]
Key Idea

- 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, $2^{128}$

\[
H_{i-1} \quad M_i
\]

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.

Obtain 64 linear equations, $2^{192}$
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, $2^{128}$

\[ m_0 \quad m_1 \quad m_2 \quad m_3 \]

Linear redundancy: dim=612

Obtain 64 linear equations, $2^{192}$
Method 1

[Mendel-Pramstaller-Rechberger]
FSE 2008

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

\[ m_0 \quad m_1 \quad m_2 \quad m_3 \]

linear redundancy \( \text{dim}=64 \)

\[ k_0 = P(h \oplus m) \]

Ps-coll./prei with \( x_0 = 0 \)

obtain 64 linear equations, \( 2^{192} \)
Method 2

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

$c_0 = c_1$

obtain $64$ linear equations, $2^{192}$

$h_0 = h_1$

$k_0 = k_1$
Why Do This?

Assume $256 + 64 + 64$ linear equations, $2^{128}$

Application 1:
find Ps-collisions $T = 2^{96}$

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

Obtain 64 linear equations, $2^{192}$
17.1.1. Pseudo-Collisions

\[ T = 2^{96} < 2^{128} \]
Ps-Collisions

[Mendel-Pramstaller-Rechberger]
FSE 2008 appendix

Also works with our Method 2!

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, $2^{128}$

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 $x_0=x_1$.

Method 3???: Cost??? to generate $2^{64}$ cases with $x_0=0$ and $x_1=0$. ???

\[ H_{i-1}, M_i \]
17.1.2. Ps-Pre-Images

\[ T = 2^{192} < 2^{255} \]

fewer methods
Method 1

[Mendel-Pramstaller-Rechberger]
FSE 2008

GOST, Self-Similarity and Cryptanalysis of Block Ciphers

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

$256 + 64$ det guess

$256$ + $64$ $k_0 = P(h \oplus m)$

$256$ $h_0$

$256$ $h_1$

$256$ $h_2$

$256$ $h_3$

$256 + 256$ $M_i$

$256 + 256$ $M_i$

obtain $64$ linear equations, $2^{192}$

$64$ $c_0$

correct?

$64$ $s_0$

$64$ $s_1$

$64$ $s_2$

$64$ $s_3$

linear redundancy (dim=612)

$1024$ $x_0$

det

$256$ $x_0$

$K$

© Nicolas T. Courtois, 2006-2013
Pre-Images

With Method 1 we can first of all choose $h_0, k_0$ 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 $2^{256+64+64}$ 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.

Assume $2^{256+64+64}$ linear equations, $2^{128}$

$H_{i-1}$ $M_i$ $H_i$
Method 2’
not equally good

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

H_{i-1} → 256+256 → 256+64 → 1024 linear redundancy

fix impose

m_0 m_1 m_2 m_3 linear

m_0 m_1 m_2 m_3 linear

h_0 = h_1 impose

64

h_0 h_1 h_2 h_3

k_0 + k_1

i_0

x_0 x_1 ...

impose correct?

obtain 64 linear equations, 2^{192}

© Nicolas T. Courtois, 2006-2013
Pre.Images

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

$H_{i-1}$ $M_i$

- With Method 2' we can fix $c_0 \oplus c_1$ such that $s_0 \oplus s_1 = 0$ 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.

obtain 64 linear equations, $2^{192}$
Conclusion

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.
17.2.
Real Collisions, Real Pre-Images
Collisions in the Strong Sense

Can be done with the same cost?

“Stronger” Collisions:
$H_{i-1}$ is arbitrary fixed,
use just $M_i$ to make
it collide nevertheless

\[
\begin{align*}
\Delta &\neq 0 \quad M_i \\
\Delta &\neq 0 \quad \text{cannot chose} \\
\Delta &\neq 0 \\
\end{align*}
\]
Better M1

Assume 256+64+64 linear equations, \(2^{128}\)

Get \(s_0 = h_0\) by selection of \(k_0\)

Obtain 64 linear equations, \(2^{192}\)
**Strong Collision**

*Assume* $h_0$ symmetric. Attack does NOT work if not symmetric. $P=2^{-32}$ over the fixed input $H_{i-1}$.

- **Fix** $c_0 \Rightarrow 64$ LIN. on $M_i$. 
  
- **Get** $s_0=h_0$ by selection of $k_0$. 
  Constrained by 64 linear equations above due to the choice of $c_0$.

Double MITM approach to generate $2^{64}$ keys on 256 bits.

For each $d_1$ key first $4R$ dim=$128 \rightarrow 64$. 

$T=2^{64}$ enum middle. Get $2^{64}$ matches on $L_4, R_4$. Get $2^{64}$ solutions $k_0$. $M=2^{64}$. 
Repeat $2^{32}$ times. 
Enum $2^{96}$ good $k_0$, out of $2^{192}$ existing.

© Nicolas T. Courtois, 2006-2013
Analysis

Cost = 4+4 rounds of GOST per each good $k_0$, generated. This $k_0$ gives $x_1$, cost = one GOST encryption. Early abort if $x_1$ is not the target.

Overall cost = $2^{96}$ GOST encryptions = $2^{94}$ GOST compression calculations.
Use fixed points for $h_0,k_0$: then $s_0 = h_0$.

In addition fix THE WHOLE of $H_i$
and in such a way that $h_0$ symmetric.

Fix $c_0 = 64$ equations on $M_i$, only works because is $H_i$ fixed and now these become also equations on $k_0$ alone!
As $k_0 = P(h \oplus m)$. If $H_i$ were not fixed this would NOT work!

$2^{32} \times \text{MITM}$: in time $2^{32+64}$ enumerate $2^{32+64}$ keys $k_0$ with these extra 64 eqs.
For this fixed $H_i$ we get $2^{32+64}$ messages where $x_0$ is the target we wanted.
Multi-Collisions

Cost = $128 \times (2^{32} + 2^{96})$

$h_0=0$ symmetric

$2^{128}$ collisions obtained
Strong Pre-images!

Crypto 2008

Works the same way: for every fixed $H_i$ s.t. $h_0$ is symmetric we construct lots of fixed points with linear constraints on $k_0$.

Time $= 2^{192}$. The whole space is used.
the end