Differential Cryptanalysis of GOST

Nicolas T. Courtois[1] and Michał Misztal[2]

[1] University College London, Gower Street, London, UK,
n.courtois@cs.ucl.ac.uk
[2] Military University of Technology, Kaliskiego 2, Warsaw, Poland,
mmisztal@wat.edu.pl

Abstract

GOST 28147-89 is a well-known block cipher and the official encryption standard of the Russian Federation. A 256-bit block cipher considered as an alternative for AES-256 and triple DES, having an amazingly low implementation cost and thus increasingly popular and used [22, 23, 30, 24, 41]. Until 2010 researchers unanimously agreed that: "despite considerable cryptanalytic efforts spent in the past 20 years, GOST is still not broken", see [30] and in 2010 it was submitted to ISO 18033 to become a worldwide industrial encryption standard. In 2011 it was suddenly discovered that GOST is insecure on more than one account. There is an amazing variety of recent attacks on GOST [8, 15]. We have reflection attacks [26, 15], attacks with double reflection [15], and various attacks which does not use reflections [15, 8]. All these methods follow a certain general framework called "Algebraic Complexity Reduction", a new general "umbrella" paradigm introduced in [15, 8]. The final key recovery step is in most cases a software algebraic attack [15, 8] and sometimes a Meet-In-The-Middle attack [26, 15].

In this paper we show that GOST is NOT SECURE even against (advanced forms of) differential cryptanalysis (DC), Previously Russian researchers postulated that GOST will be secure against DC for as few as 7 rounds out of 32 [18, 38] and Japanese researchers were already able to break about 13 rounds [37]. In this paper we show a first advanced differential attack faster than brute force on full 32-round GOST. This paper is just a sketch and a proof of concept. More results of this kind will be published soon.

Key Words: Block ciphers, GOST, differential cryptanalysis, sets of differentials, aggregated differentials, iterative differentials.

∗ This work was supported by Polish Ministry of Science and Higher Education under research project Nr O R00 0111 12.

1 Introduction on GOST

1.1 The Official Status of GOST

GOST 28147-89 was standardized in 1989 and first it became an official standard for the protection of confidential information but the specification of the cipher remained confidential [20]. In 1994, the standard was declassified, published and also translated to English [20, 21]. It is also described in several more recent Internet standards [23, 22]. Unlike DES which could only be used to protect unclassified information, and like AES, GOST allows to protect also classified and secret information apparently without any limitations, which is explicitly stated by the Russian standard, see the first page of [21]. Therefore GOST is much more than a Russian equivalent of DES, and its large key size of 256 bits make GOST a plausible alternative for AES-256 and 3-key triple DES. The latter for the same block size of 64 bits offers keys of only 168 bits. Clearly GOST is a very serious military-grade cipher designed with most serious applications in mind. At least two sets of GOST S-boxes have been explicitly identified as being used by the two of most prominent Russian banks and financial institutions cf. [38, 24]). The Russian banks in question need to securely communicate with tens of thousands of branches to protect assets worth many hundreds of billions of dollars against fraud.

1.2 GOST And Its S-boxes

GOST is a block cipher with a simple Feistel structure, 64-bit block size, 256-bit keys and 32 rounds. Each round contains a key addition modulo 2^{32}, a set of 8 bijective S-boxes on 4 bits, and a simple rotation by 11 positions. The GOST S-boxes can be secret and they can be used to constitute a secondary key which is common to a given application, further extending key size to a total of 610 bits. One set of S-boxes has been published in 1994 as a part of the Russian standard hash function specification GOST R 34.11-94 and according to Schneier [38] this set is used by the Central Bank of the Russian Federation. They appear in RFC4357 [23] as a part of the so called "id-GostR3411-94-CryptoProParamSet". This precise version of GOST 28147-89 block cipher is the most popular one, and it is commonly called just "the GOST cipher" in the cryptographic literature. In this paper we concentrate on this set of S-boxes. A basic reference implementation with only these S-boxes can be found in Crypto++ library [41].

Other S-boxes. Differential attacks with other sets of S-boxes will be published elsewhere. The most complete current reference implementation of GOST which is of genuine Russian origin and is a part of OpenSSL library, contains eight standard sets of S-boxes [24]. Other (secret) S-boxes could possibly be recovered from a chip or implementation, see [36, 17].

1.3 GOST Is Very Competitive

In addition to the very long bit keys GOST has a much lower implementation cost than AES or any other comparable encryption algorithm. It really costs much less than AES: for example in hardware GOST 256 bits requires less than 800 GE, while AES-128 requires 3100 GE, see [30]. More than 4 time more gates for a much lower level of security (nearly 10^{40} times lower).

Thus it is not surprising that GOST became an Internet standard [23, 22], it is part of many crypto libraries such as OpenSSL and Crypto++ [24, 41], and is increasingly popular also outside its country of origin [22, 23, 30]. In 2010 GOST was submitted to ISO to become a worldwide encryption standard. Very few crypto algorithms have ever become an international standard. ISO/IEC 18033-3:2010 specifies the following algorithms. Four 64-bit block ciphers: TDEA, MISTY1, CAST-128, HIGHT and three 128-bit block ciphers: AES, Camellia, SEED. GOST is intended to be added to the same standard ISO/IEC 18033-3.

Now it appears that never in history of industrial standardisation, we had such a competitive algorithm in terms of cost vs. claimed security level. GOST also has 20 years of cryptanalysis efforts behind it, and it appears that this claimed military-grade security level was never disputed, until now.

Update: In April 2011 [private communication] GOST was voted against by a majority of countries in an ISO vote in Singapore, but the result of this vote was later overthrown at the ISO SC27 plenary level, and thus ISO is still in the process of standardizing GOST at the time of writing.

2 Security of GOST

2.1 What Experts Have Once Said About GOST

Nothing in the current knowledge and literature about GOST ever suggested that it could be insecure. On the contrary, large keys and a large number of 32 rounds make that GOST seems a plausible encryption algorithm to be used for many decades to come.

Everyone familiar with the Moore's Law, understands that, in theory 256-bit keys should remain secure for at least 200 years. GOST was widely studied by the top cryptography experts active in the area of block cipher cryptanalysis such as Schneier, Biham, Biryukov, Dunkelman, Wagner, various Australian, Japanese, German and Russian scientists, ISO cryptography experts, and all researchers always seemed to agree that it could be or should be secure. While it is widely understood that the structure of GOST is in itself quite weak, for example compared to DES, and in particular the diffusion is not quite as good, it was however always stipulated that this should be compensated by a large number of 32 rounds cf. [18, 38, 37] and also by the additional non-linearity and diffusion provided by modular additions [18, 33]. In [4], Biryukov and Wagner write: "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." In the same paper we read: "after considerable amount of time and effort, no progress in cryptanalysis of the standard was made in the open literature". Thus, so far there was no significant attack on this algorithm from the point of view of communications confidentiality: an attack which would allow decryption or key recovery in a realistic scenario where GOST is used for encryption with various random keys. In contrast, there are already many many papers on weak keys in GOST [27, 4], attacks for some well-chosen number of rounds [27, 2, 37], attacks with modular additions removed [4], related-key attacks [29, 16, 35], reverse engineering attacks on S-boxes [36, 17], and at Crypto 2008 the hash function based on this cipher was broken [25]. In all these attacks the attacker has much more freedom than we would allow ourselves here. However, as far as traditional encryption applications with random keys are concerned, until 2011, no cryptographically significant attack on GOST was ever found, which was summarized in 2010 in these words: "despite considerable cryptanalytic efforts spent in the past 20 years, GOST is still not broken", see [30].

2.2 Linear and Differential Cryptanalysis of GOST

In the well known Schneier textbook written in the late 1990s we read: "Against differential and linear cryptanalysis, GOST is probably stronger than DES", see [38]. Then in 2000 Russian researchers claimed that "breaking the GOST with five or more rounds is very hard". and explain that as few as 5 to 7 rounds are sufficient to protect GOST against linear and differential cryptanalysis. In the same year, Japanese researchers [37], explain that in addition, such straightforward classical differential attack with one single differential characteristic are unlikely to work **at all** for a larger number of rounds. This is due to the fact that they only work for a fraction of keys, likely to rapidly decrease with the number of rounds, see [37]).

Yet in the same paper [37], more advanced differential attacks on GOST are described. They show how to break about 13 rounds of GOST and until now it was not clear if these attacks can be extended in any way to a larger number of rounds such as full 32 rounds, because partial internal differences generated in such attack become very hard to distinguish from differences which occur naturally at random. These questions are a central topic in this paper and we will come back to it later.

2.3 Sliding and Reflection Attacks

According to Biryukov and Wagner, the structure of GOST, and in particular the reversed order of keys in the last 8 rounds, makes it secure against sliding attacks [3, 4]. However the cipher still has a lot of self-similarity and this exact inversion of keys allows other attacks in which fixed points are combined with a so called "Reflection" property [25, 27]. The latter attack breaks GOST only for certain keys, which are weak keys.

2.4 Recent Developments, Algebraic and MITM Attacks

A new attack which also uses reflection, and finally breaks GOST, was very recently presented at FSE 2011, see [26]. A similar but different, simpler and faster attack, which is not the same attacks as sometimes claimed, appears in [15]. These two attacks require about 2^{132} bytes of memory which makes them arguably worse even than slower attacks with less memory.

Many new attacks which also use reflections and even simultaneous multiple reflections, which work for most GOST keys, and which allow to really break full-round GOST with 256-bit keys, not only for some weak keys like in [27] have been recently developed, see [15]. All these attacks require much less memory, and some are substantially faster, see [15]. These new attacks can be seen as examples of a new general paradigm for effective block cipher cryptanalysis called "Algebraic Complexity Reduction" which generalizes these attacks, and also generalizes many other known fixed point, slide, involution and other attacks. Importantly, there are now several attacks which allow to cryptanalyse GOST without any reflection, see [8, 15]. Finally, there is also many new weak key attacks way stronger than any previously found weak key attack [15]. Some classes of these weak keys are frequent enough to be able to occur in the real life. For some natural variants of GOST there are some nearly-practical attacks [15].

The name of "Algebraic Complexity Reduction" is also explained by the fact that most of these attacks could not be developed previously, because only in the recent 5 years it became possible to show the existence of an appropriate last step for many such attacks, which is a low data complexity attack. Most of the time it is a software algebraic attack [15, 8] and sometimes a Meet-In-The-Middle attack [26, 15].

3 Linear and Differential Cryptanalysis of GOST

3.1 Previous Research and Application to GOST

A basic assessment of the security of GOST against linear cryptanalysis (LC) and differential cryptanalysis (DC) has been conducted in 2000 by Gabidulin *et al*, see [19, 18]. The results are quite impressive: at the prescribed security of level of 2^{256}, 5 rounds are sufficient to protect GOST against LC. Moreover, even if the S-boxes are replaced by identity, and the only non-linear operation in the cipher is the addition modulo 2^{32}, the cipher is still secure against LC after 6 rounds out of 32. In [18] the authors also estimate that, but here only w.r.t. the security level of about 2^{128}, roughly about 7 rounds should be sufficient to protect GOST against DC.

Remark: The authors also claimed that "breaking the GOST with five or more rounds is very hard". However it is known that algebraic attacks allow to break more or less any cipher with such a small number of rounds, routinely and without human intervention. For example one can break 6 rounds of DES, [13] and about 8 rounds of GOST [15] quite easily.

3.2 Differential Cryptanalysis and Multiple Key Scenarios

Differential cryptanalysis (DC) [5] of GOST have attracted more attention than linear cryptanalysis (LC). It is possible to see that differential cryptanalysis is a much more "practical" attack than linear cryptanalysis: it does not require an astronomical quantity of data to be collected for one single key, which will never occur in practice because nobody encrypts such quantities of data. Differential cryptanalysis works also in a scenario where many different keys are used by different people. It will then allow to break one of these keys, see [5, 6]. However for simplicity, in this paper we still concentrate on the single key scenario. More advanced differential attacks on GOST will be published elsewhere.

We recall that in 2000 Russian researchers have claimed that 7 rounds are sufficient to protect GOST against differential cryptanalysis [19, 18]. This is correct in the classical understanding of this attack which comes from differential attack on DES [5, 6]. Moreover there is an additional difficulty which makes that against DC in this narrow sense GOST is, indeed probably "stronger than DES" as once claimed in [38].

3.3 Classical Biham-Shamir DC Attacks and GOST

The difficulty is explained by the Japanese researchers in 2000 [37]. If we consider the straightforward classical differential attack with one single differential characteristic it is in fact **unlikely to work at all** for a larger number of rounds.

This is due to the fact that when we study reasonably "good" iterative differential characteristics for a limited number of rounds (which already propagate with probabilities not better than $2^{-11.4}$ per round, cf. [37]), we realize that they only work for a fraction of keys smaller than half. For full 32-round GOST such an attack with a single characteristic would work only for a negligible fraction of keys of about 2^{-62} (and even for this tiny fraction if would propagate with a probability not better than 2^{-360}).

3.4 Advanced Differential Attacks on GOST

This however does not prevent more advanced differential attacks on GOST which are the central topic in this paper. They have been introduced in 2000 in the same already cited Japanese paper [37]. They can be seen in two different ways, either as attacks which use sets of differentials as they are formalized in [37], or more specifically, as it is the case for the most interesting attacks of this type known (from [37] and in this paper), as attacks in which differentials are truncated, so that the best sets of differentials actually follow certain patterns, for example certain bits or whole S-boxes have zero differentials, and only some bits are active and have non-zero differentials. These attacks differ considerably from the classical differential cryptanalysis [5, 6]. They depend strongly not only on the S-boxes but also rely on the the weak diffusion of GOST, and in the first approximation they are essentially distinguisher attacks on block cipher components which are not easy to transform into key recovery attacks.

3.5 Previous Advanced DC Attack on GOST

The best key recovery attack proposed in [37] allows to break between 12 and 17 rounds of GOST depending on the key, some keys being weaker. The four main components of this attack are:

1. An initial extension with initially a small number of only 3 active bits at the input of GOST. After 8 rounds, with some probability, the difference has 8 active differences on 3 bits, and 40 other bits are totally inactive.

2. An iterative set of differentials with 24 active bits which can propagate for an arbitrary number of rounds.

3. A final extension differential with again a much smaller number of active bits,

4. A method for key recovery with guessing some key bits.

Attacks with sets of differentials can be applied to other ciphers, for example Q [1] and more recently PP-1 [9]. However the exact method to transform the main iterative component (the only one which extends to an arbitrary number of rounds) into a key recovery attack will vary considerably. In general the topic of transforming such an iterative advanced differential attack into a key recovery attack is very complex and it is quite difficult to know what will be the best method or even how the best method should look like.

In this paper we concentrate mainly on the iterative part for GOST, then we will also propose an initial extension, and also a final extension. Our final key recovery uses the key scheduling of GOST and thus is different than in previously published works [37, 1, 9]. We expect that there will be a lot of work about this in the future and all the attacks we outline here can almost certainly be improved.

3.6 Key Recovery in Previous Advanced DC Attacks

Key recovery can be **very** difficult in this type of attacks. Sets of differentials occur naturally with higher probability, and when they occur they give much less exploitable information about the secret keys. In [37] the main iterative set of differentials occur naturally with higher probability of 2^{-50}, which is not negligible anymore like in DES [5, 6]. We are no longer dealing with exceptional events which never happen by accident, and which when they happen, happen for a specific reason and yield a lot of information when they happen, like in DES [5, 6]. For GOST, when some differentials in a set are attained, there is a lot of ambiguity about why exactly they are attained, and such events give much less exploitable information about the secret keys than in DC for DES [5, 6].

If we study the attack described in [37], we see that for 13 rounds, we get a differential property which holds with probability of about 2^{-49} over 13 rounds cipher and which occurs naturally, also for random data, with probability 2^{-50}. Observing one such difference allows to recover 32 bits of the key in the last round. However to recover any other key bits is much harder in this type of attack, therefore as the first approximation, we need to consider that the time complexity of the attack from [37] is 2^{224} GOST encryptions, and it is not clear if this can be improved.

Summary. The best advanced multiple differential attack proposed in [37] allows to break between 12 and 17 rounds of GOST depending on the key, some keys being weaker. However it is not clear at all, if these attacks can be extended in any way to a larger number of rounds such as full 32 rounds, because partial internal differences generated in the attack become very hard to distinguish from differences which occur naturally at random.

In this paper we will greatly improve on the state of the art and develop the first differential attack on full 32 rounds.

4 From GOST to New Differential Attacks on GOST

GOST is a block cipher with a simple Feistel structure, 64-bit block size, 256-bit keys and 32 rounds. Each round contains a key addition modulo 2^{32}, a set of 8 bijective S-boxes on 4 bits, and a simple rotation by 11 positions.

Differential characteristics in GOST, need to take into account not only the S-boxes, like in DES, but also the key addition modulo 2^{32}, which makes that their probabilities depend on the key. This is a major difficulty in differential cryptanalysis of GOST.

In this paper we summarize the state of the art and report some very important new results. The (very technical) explanation on how to obtain this type of results through extended computer simulations is outside the scope of this paper and will appear elsewhere. Also this is a research in progress and many similar and even better attacks on various versions of GOST can be developed and will be published soon.

In this paper we consider the "untwisted" representation of GOST where the left hand side is modified in odd rounds $1, 3, \ldots$, and the right hand side is modified in even rounds $2, 4, \ldots$.

5 Vocabulary: Aggregated Differentials

We define *an aggregated differential* A, B as the transition where any non-zero difference $a \in A$ will produce an arbitrary non-zero difference $b \in B$ with a certain probability.

In the previous work on GOST exactly the same sorts of differentials are exploited for GOST, see [37]. The are called "sets of differential characteristics" however this would suggest that any set of characteristics is possible, for example $a \Rightarrow b$ and $a' \Rightarrow b'$ could be permitted but not $a \Rightarrow b'$. This is an unnecessarily general notion. Our notion of Aggregated Differentials only allows "sets of differential characteristics" which are in a Cartezian direct product of two sets $A \times B$.

Similar sets of differentials are also called "almost iterative differentials" in [1], however the word "almost" can be seen as misleading, because here and elsewhere [1, 9] we will have "perfectly" iterative differentials, which are perfectly periodic, and can propagate for an arbitrary number of rounds, from set A, to exactly the same set A.

6 Multiple Differential Attacks on GOST

6.1 Previous Results From [37]

As already explained this type of differential attack on GOST was introduced in 2000 [37] under the name of "sets of differential characteristics". They exploit sets of differentials, which in addition follow certain patterns, for example certain bits have zero differentials, other bits are active and have non-zero differentials. Such sets of differentials do work extremely well while ordinary DC, as explained in Section 3.4, fails for GOST.

They work for more or less all possible keys, or with a high probability. They work for various S-boxes, and also when S-boxes are chosen at random, see [37]. Moreover it is easy to see that they will also propagate well and can be detected when the S-boxes are kept secret.

For example the difference of type 0x70707070,0x07070707, where each 7 means an arbitrary difference on 3 bits, plus extra rules to exclude all-zero differentials, will propagate for one round with a probability of about $2^{-5.3}$ and for any key chosen at random. In fact the result is slightly different for even and odd rounds and for specific fixed keys this probability will differ substantially.

Here what we report will already start to differ from the combination of theoretical probabilities given in [37]. This is because it is very hard to predict what really happens with complex sets of differentials by theory. In fact it is rather impossible for complex differentials which could propagate over many rounds, to enumerate all possible differential paths which could at the end produce one of the differentials in our set. Moreover they strongly depend on the key. Therefore the more rounds we have, the more the actual (experimental) results will differ from predictions. Moreover the difference is in our experience almost always beneficial to the attacker: as we will see below, better attacks than expected are almost always obtained.

For example, by theory, this differential set 0x70707070,0x07070707, described above and in [37], would propagate with a probability of about only 2^{-160} over 32 rounds, (though actually it is better in practice due to propagation through additional differential paths, as we will see below). This is not very good: there are only 2^{64+24} possible input differences for a block size of 64-bits. Moreover, an output difference of type 0x70707070,0x07070707 occurs naturally with probability about 2^{-40}. There will be really a lot of false positives in any potential attack using this differential. It is not clear how to deal with false positives, and it is generally considered that this type of attacks will not work after a certain number of rounds have been reached, see [37].

6.2 New Results

Many very good characteristics exist for GOST. Here we give one example. This example has been constructed by hand by the authors from differential characteristics of various S-boxes and already reported in one paper [9]. We expect that differentials which are better still can be found.

Consider the following differential set:

$$\Delta = 0x80700700$$

by which we mean all differences with between 1 and 7 active bits (but not 0) and where the active bits are contained within the mask 0x80700700. Similarly, an aggregated differential (Δ, Δ) means that we have 14 active bits, and that any non-zero difference is allowed. There are $2^{14} - 1$ differences in this set of ours. The following fact can be verified experimentally:

Fact 6.2.1 *The aggregated differential (Δ, Δ) with uniform sampling of all differences it allows, produces an element of the same aggregated differential set (Δ, Δ) after 4 rounds of GOST with probability about $2^{-13.6}$ on average over all possible keys.*

This probability is an average and it depends on the key, for example if all key bits are equal to 0 this probability is different and equal to $2^{-13.2}$.

Importantly, for 8 rounds the result is better than the square of $2^{-13.6}$ which would be $2^{-27.2}$. It is:

Fact 6.2.2 *The aggregated differential (Δ, Δ) (again with uniform sampling) produces the same aggregated differential (Δ, Δ) after 8 rounds of GOST with probability about $2^{-25.0}$ on average over all possible keys.*

Remark 1. Again, for some keys it will be smaller or bigger. For example if all key bits are equal to 0 a computer simulation gives the probability of $2^{-22.8}$. It appears also that the approximation gives similar results for most keys and we found no keys for which this probability would be significantly worse than $2^{-25.0}$.

Remark 2. The same kind of improvement, very hard to analyse by theory, but quite substantial and easily visible in practice, also exists for Japanese attacks from [37], see Figure 1.

6.3 Propagation for 16 Rounds

Fact 6.3.1 *The aggregated differential* (Δ, Δ) *produces the same aggregated differential* (Δ, Δ) *after 16 rounds of GOST with probability about* 2^{-48} *on average over all possible keys.*

Justification: Here is how this is estimated. In theory if we just compose 2 pieces of 8 rounds, we get 2^{-50}. However, the difference observed between Fact 6.2.1, Fact 6.2.2, is an improvement by a factor of $2^{+2.2}$ when the two pieces of GOST are joined together and a number of additional highly probable differentials can occur at the junction. Here the junction is done again, and very roughly we expect that the propagation probability will be about

$$2^{-25+2.2-25} \approx -2^{-48}.$$

A more precise result need to be obtained by computer simulations.

This needs to be compared to the probability that the output difference set (Δ, Δ) will also occur naturally. In this set there are exactly 50 inactive bits where the difference must always be 0. Therefore:

Fact 6.3.2 *The 64-bit output difference being a member of our set* (Δ, Δ) *occurs naturally with probability about* 2^{-50}.

6.4 Detailed Comparison New vs. Previous Results

We need to compare our result with the Japanese paper [37] from 2000. If we apply the probabilities found in [37], in theory, we expect that the difference of type 0x70707070,0x07070707 will propagate for 8 rounds with a probability of about $2^{-42.7}$. Our simulations show it is **much** higher. It is about $2^{-28.4}$ in practice. Similarly, the theory (according to [37]) says that this aggregated differential 0x70707070,0x07070707 will propagate for 16 rounds with a probability of about $2^{-85.3}$. However be decomposing it as 8+8 rounds we clearly see that it will be at most about 2^{-56} in practice. Unhappily, a differential of type 0x70707070,0x07070707 occurs naturally with probability about 2^{-40}. Here we are not able to distinguish 16 rounds from a random permutation.

Our aggregated differential (Δ, Δ) with $\Delta = 0x80700700$ occurs with a better probability of about 2^{-48} while it occurs naturally with probability of about 2^{-50}. Clearly with the new method we are able to distinguish 16 rounds of GOST from a random permutation. Some of these results were already reported in [9]. Other results are extrapolations.

Input Aggregated Differential	0x70707070,0x07070707	0x80700700,0x80700700
Output Aggregated Differential	0x70707070,0x07070707	0x80700700,0x80700700
Reference	Seki-Kaneko [37]	this paper and [9]
Propagation 2 R	$2^{-8.6}$	$2^{-7.5}$
Propagation 4 R	$2^{-16.7}$	$2^{-13.6}$
Propagation 6 R	$2^{-24.1}$	$2^{-18.7}$
Propagation 8 R	$2^{-28.4}$	$2^{-25.0}$
Propagation 10 R	2^{-35}	$2^{-31.1}$
Propagation 12 R	2^{-43}	2^{-36}
Propagation 14 R	2^{-50}	2^{-42}
Propagation 16 R	2^{-56}	2^{-48}
Propagation 18 R	2^{-62}	2^{-54}
Propagation 20 R	2^{-70}	2^{-60}
Propagation 22 R	2^{-77}	2^{-66}
Output Δ Occurs Naturally	$2^{-40.0}$	$2^{-50.0}$

Figure 1: Our results and further extrapolations vs. previous results

6.5 An Initial Extension

It appears that the best single differential suitable for the initial extension for (Δ, Δ) is one which does not modify anything in the first round, and one which will only affect the highest bit in the modular addition in the second round which does not generate any carries. This differential is $(0x80000000, 0x00000000)$.

In the following table we report some results which are computed as follows: for smaller number of rounds X they are computed experimentally. For a larger X and for all possible decompositions $X = Y + Z$ we consider that for the first Y rounds we have achieved the iterative set from the initial set, and for the remaining Z rounds we apply one of the exact results from Table 1 above. Due to the method these results are rather conservative estimations. We compare these results to those obtained assuming that one used the initial extension from [37] which again are exact results for up to 8 rounds, and beyond we again give lower bounds and based on a decomposition of $X = Y + Z$ rounds with Y rounds being as in Table 2 and Z being the main iterative piece as in Table 1, where we report the best lower bound obtain by such decomposition which again is a conservative estimation.

Input Aggregated Differential	0x00000700,0x00000000	0x80000000,0x00000000
Output Aggregated Differential	0x70707070,0x07070707	0x80700700,0x80700700
Reference	Seki-Kaneko [37]	this paper
Propagation 2 R	$2^{-1.4}$	$2^{-0.4}$
Propagation 4 R	$2^{-5.6}$	$2^{-6.2}$
Propagation 6 R	$2^{-11.5}$	$2^{-12.2}$
Propagation 8 R	$2^{-20.1}$	$2^{-20.5}$
Propagation 10 R	2^{-28}	$2^{-24.6}$
Propagation 12 R	2^{-34}	$2^{-30.3}$
Propagation 14 R	2^{-41}	2^{-36}
Propagation 16 R	2^{-46}	2^{-42}
Propagation 18 R	2^{-55}	2^{-47}
Propagation 20 R	2^{-61}	2^{-53}
Propagation 22 R	2^{-66}	2^{-59}
Output Δ Occurs Naturally	$2^{-40.0}$	$2^{-50.0}$

Figure 2: Some results with our initial extension

7 A Final Extension

In this section we propose a final extension with less active bits which seems to work well for various numbers of rounds. The current extension have been selected as one which works very well for 10 and more rounds. It does **not** work as well for less rounds and many other aggregated differentials are better for less rounds.

7.1 Final Extension

We have the following estimation:

Fact 7.1.1 *The ordinary differential* $(0x80000000, 0x00000000)$ *produces a non-zero differential in the set* $(0x00000100, 0x80600200)$ *after 20 rounds of GOST with probability roughly at least about* 2^{-58} *on average over all possible keys.*

Justification: Currently we have no good method to estimate this probability. We consider an affine model. Our simulations show that this differential works well for 8 rounds and more, for 6 or less rounds we would need to consider aggregated differentials of the form $(0x80000000, 0x00000000) \Rightarrow (0x80000100, 0x80600200)$ or similar. However these are not the best anymore for 8 and more rounds.

Experimental results obtained for our best aggregated differential which is $(0x80000000, 0x00000000) \Rightarrow (0x00000100, 0x80600200)$ and for 8, 10 and 12 rounds are respectively $2^{23.6}$, $2^{28.4}$, and 2^{33}. If we extrapolate from the last two values we get that maybe for 20 rounds the result could be $2^{-28.4-4.8\cdot(20-10)/2} \approx 2^{-52}$. Being conservative we postulate it is at least 2^{-58} due to the propagation (cases with many intermediate differentials of small Hamming weight) and another 2^{-59} which occur by accident because we have 59 inactive bits in $(0x00000100, 0x80600200)$. We do not have enough computing power to confirm this result. **This result is inexact.** A better method to estimate this probability needs to be developed in the future.

7.2 Our Distinguisher For 20 Rounds

We can note that the aggregated differential $(0x00000100, 0x80600200)$ has 5 active bits and occurs naturally with probability 2^{-59}. In contrast the propagation we predict is expected to occur with probability of about 2^{-58} (with many intermediate differentials of small Hamming weight).

We are finally able to distinguish 20 rounds from a random permutation. We study this distinguisher in more details.

For a random permutation, if we consider 2^{63} possible pairs with an input difference $(0x80000000, 0x00000000)$, about $2^4 = 2^{63-59}$ pairs will have an output difference in the desired form in $(0x00000100, 0x80600200)$ purely by accident (a collision on 59 bits in the last round, arbitrary differentials in the middle of the computation).

In the case of the actual 20 rounds of GOST we expect that there will be another and additional number of about $2^5 = 2^{63-58}$ "good" pairs and with output difference in $(0x80700700, 0x80700700)$ and with many intermediate differentials of small Hamming weight. It is essential to understand that the overlap between these sets of 2^4 and 2^5 pairs should be most of the time negligible, because we are well below the birthday paradox bound for two sets to overlap.

Therefore in our attacks on 20 rounds we expect to get about 2^4 pairs when the key guessed is incorrect, and about 2^4+2^5 when it is correct. Can we distinguish between these two cases?

The standard deviation for the number of cases (collisions on 59 bits which occur by accident) can be computed as a sum of 2^{63} independent random variables equal to 1 with probability 2^{-59} and the result is 2^2. Our additional 2^5 cases amounts to 8 standard deviations. By applying the central limit theorem and assuming that our sum of 2^{63} independent random variables is Gaussian, and by applying the Gauss error function [40] we obtain that the probability that by accident the number reaches a threshold of 2^4+2^5 when it is correct, which is outside 8 standard deviations, is about 2^{-50}. In contrast if the permutation has these additional pairs which come from the propagation of our differential, we expect to be above the threshold of $2^4 + 2^5$ with probability 2^{-1}.

This is the basis of our attack and this distinguisher will be used to filter out a large proportion of wrong assumptions on GOST keys.

Future Work: Our aggregated output differential is in fact a collection of 2^5-1 ordinary differentials, and the frequency of these differentials is not uniform. Thus better distinguishers can probably be developed. However this is **not** very easy because the distribution strongly depends on the key.

8 Key Recovery Attacks on Full 32-Round GOST

The key property which makes that we can substantially reduce the number of rounds in full 32-round GOST is that the order of 32-bit words in the key schedule is inversed in the last 8 rounds. In the fist 8 rounds we have:

$$k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7, \ldots$$

In the last 8 rounds we have:

$$\ldots, k_7, k_6, k_5, k_4, k_3, k_2, k_1, k_0$$

Thus for example if we guess $k_0, k_1, k_2, k_3, k_4, k_5$ we are left with only 20 rounds inside GOST. Then the simplest attack one can think of based on Fact 7.1.1 would work as follows:

8.1 First Differential Attack on GOST Faster Than Brute Force

For each 192-bit guess $k_0, k_1, k_2, k_3, k_4, k_5$ and for each of 2^{64} P/C pairs for 32 rounds, we compute the first 6 rounds forwards, and the last 6 rounds backwards. thus we get 2^{64} P/C pairs for 20 rounds. This would require a total time spent in this step of about $2^{192} \cdot 2^{64} \cdot 12/32$ GOST encryptions which is $2^{254.6}$ GOST encryptions, slightly less than brute force but not much less. Then we can discard a proportion about 2^{-50} of keys on 192 bits. Remaining key bits are found by brute force.

Summary. This attack requires 2^{64} KP and allows to break full 32-round GOST in time of about $2^{254.6}$ GOST encryptions for a success probability of 50 %.

8.2 Toward An Improved Differential Attack on GOST

In our first attack, we want to avoid the most costly step. We recall that for each of 2^{192} keys we check all possible P/C pairs for 32 rounds, we decrypt 6 rounds forwards and backwards. We can observe that but at the end the attacker has filtered some 2^4 or $2^4 + 2^5$ pairs for 20 rounds and most of the decryptions are not useful. We want to be able to compute the expected 2^4 or $2^4 + 2^5$ pairs for 20 rounds more efficiently by progressive filtering, knowing that if we guess less key bits, we can already narrow down the number of pairs which are able to have the desired differential set for 20 rounds, because due to the slow diffusion this implies differentials of certain form for an outer 22 and more rounds. To achieve this, we are basically going to guess 160 bits, then reduce the number of P/C pairs to less than 2^{64}, and only then guess additional 32 bits of the key. We proceed as follows.

We view GOST as 5+1+20+1+5 rounds.

We want to obtain a difference of the form $(0x80000000, 0x000000000) \Rightarrow (0x00000100, 0x80600200)$ in the 20 inner rounds. If such a difference holds, then it is easy to see that for the outer envelope of the 22 rounds we have differences which must be of type (we apply the "untwisted" convention) $(0x80000000, 0x00000780) \Rightarrow (0x00000100, 0x80600200)$ and this with a very good probability.

By good probability we mean at least $1 - 2^{-15}$ so that if the number of interesting differentials in our attack is less than 2^{10}, we don't lose any of them when we assume that for external 22 rounds the differences are of the form $(0x80000000, 0x00000780) \Rightarrow (0x00000100, 0x80600200)$. Thus in our attack we can assume that the following situation occurs for all the 2^4 or $2^4 + 2^5$ pairs we exploit in our distinguisher.

```
     (5 rounds)
0x80000000 0x00000780
     (1 round)
0x80000000 0x00000000
    (20 Rounds)
0x00000100 0x80600200
     (1 round)
0x00000100 0xFFF80207
     (5 rounds)
```

This set of differentials is used to filter out pairs with only 128 bits fixed, so that when we guess the additional 32 bits, there is less pairs. so that when we guess another 32 bits, there is still less pairs. We proceed as follows:

8.3 An Improved Differential Attack on GOST

Again GOST is viewed as 5+1+20+1+5 rounds. We proceed as follows:

1. We assume that the attacker has 2^{64} KP.

2. In the attack the attacker guesses k_0, k_1, k_2, k_3, k_4 which is 160 bits. We are left with only 22 rounds inside GOST. All the steps of the attack are repeated 2^{160} times.

3. For each 160-bit guess k_0, k_1, k_2, k_3, k_4 and for each of 2^{64} P/C pairs for 32 rounds, we compute the first 5 rounds forwards, and the last 5 rounds backwards. thus we get 2^{64} P/C pairs for 22 rounds. The total time spent in this step is about $2^{160} \cdot 2^{64} \cdot 10/32$ GOST encryptions which is $2^{222.3}$ GOST encryptions. Memory is about 2^{64}.

4. Now we know that if for the middle 20 rounds the input difference is $(0x80000000, 0x00000000)$, then at one round before, the difference is of the form $(0x80000000, 0x00000780)$ with an overwhelming probability. This set has 59=31+28 inactive bits.

5. Similarly if for the middle 20 rounds in the decomposition of GOST as 5+1+20+1+5 rounds, the output difference is $(0x00000100, 0x80600200)$, one round after, in full GOST at round 5+1+20+1, the output difference is contained within $(0x00000100, 0xFFF80207)$ with an overwhelming probability. This set has 31+15=46 inactive bits.

6. Now we want to filter pairs which can be "good" pairs for 20 rounds. For each out of 2^{64} plaintext P at the entry of 22 rounds we consider 2^5 plaintexts P' which coincide on the 59 inactive bits. We have a total of $2^{64} \cdot 2^5/2$ pairs P, P' such that for the middle 20 rounds the input difference could be $(0x80000000, 0x00000000)$.

   Out of these candidates, let C, C' be the ciphertext corresponding ciphertext after 22 rounds (located at round 5+22 of full GOST). We look at the values on the 46 inactive bits from $((0x00000100, 0xFFF80207)$ and expect to find about $2^{64+5-1-46} \approx 2^{22}$ pairs C, C' which collide on these 36 inactive bits.

   The total time spent in this step is about $2^{160} \cdot 2^{64+5-1} \cdot 10/32 \approx 2^{226.3}$ GOST encryptions. Memory is about 2^{64}.

7. Thus due to the moderate diffusion in the couple of outer rounds, we are left with only 2^{22} pairs for 22 rounds.

8. Now we are going to guess the 32-bit key for round 6. For each initial key guess on 160 bits we have filtered out 2^{22} pairs for 22 rounds. For each of these pairs 2^{160+22} pairs overall, we guess the 32-bit key for round 6 and obtain a pair for 20 rounds. Then we only keep it if it coincides on all 63 inactive bits in the input and on all the 59 inactive bits in the output. Most of the time we expect to obtain 2^4 pairs at this stage.

   The total time spent in this step is about $2^{160+22+32} \cdot 2/32 \approx 2^{210}$ GOST encryptions.

9. As in our distinguisher, we expect to get at the end about 2^4 pairs if the 192 bits are incorrect, and $2^4 + 2^5$ pairs if the 192 bits are correct where as explained before we fix a decision threshold at 2^4+2^5 and only accept the 192 bits, if the threshold is reached. We have previously

established that for incorrect keys we would be outside 8 standard deviations, which occurs with probability of about 2^{-50}. In contrast if the 192 bits are correct, we expect to be above the threshold of $2^4 + 2^5$ with probability 2^{-1}.

10. This allows to filter out which 192 bits are incorrect and we are left with $2^{192-50} = 2^{132}$ surviving keys on 192 bits, for which we will continue the attack.

11. Only in very rare cases where the threshold of $2^4 + 2^5$ was achieved the remaining $256 - 192$ key bits are found by brute force. Key candidates found need to be checked with additional P/C pairs for 32 rounds, most of the time just one such pair for 32 rounds allows to reject the key.

    The expected total time spent in this step is about $2^{132+256-192} \cdot 1 \approx 2^{198}$ GOST encryptions.

The total complexity, taking into account the two steps which take more time than all the other is about $2^{226.3}$ GOST encryptions.

Summary. Overall this attack requires 2^{64} KP and allows to break full 32-round GOST in time of about 2^{226} GOST encryptions for a success probability of 50 %.

This attack is just a sketch and a proof of concept. More results on differential attacks on GOST will be published soon.

9 Discussion

Frequently Asked Question. What's the point of recovering the key if the attacker is given 2^{64} KP?

Answer 1. First, the exact attack described here also works for less than 2^{64} with increasing complexity. This will be the topic for further papers, each time we can further improve our timings, we can also decrease the data complexity.

Answer 2. Secondly, even if the attacker is given the exact whole 2^{64} KP, the key recovery allows to see if all the P/C pairs he was able to obtain were encrypted with the same key or to see if they are authentic and allow to distinguish between genuine messages and fake messages. In Section 2 of [14] we can find a longer discussion with lots of interesting examples of practical applications of cryptographic attacks on block ciphers where the attacker disposes of a large part of the code-book or the whole code-book, and possibly with errors. Here we just give some examples.

Example 2.1. For example most practical systems using cryptography use key derivation and have master keys. If a master key can be recovered, its value will be much greater than of any individual key. A master key which allows to compute many other keys makes that the amortized cost of the attack per one compromised key or device will be much lower than expected.

Example 2.2. In many other cases, recovering or confirming or being able to forge or replay just one very important message, knowing what this message would mean, will be extremely valuable, see [14].

For example, a similar question apparently had a pivotal impact on winning the World War II, as reported by David Kahn, see [14, 28], where the United States needed to determine and confirm whether a certain specific cryptogram was the Midway Island. The certitude obtained allowed to strike with an overwhelming force and destroy the Japan's offensive power at sea, see [14, 28].

10 Conclusion

It was widely believed that against differential and linear cryptanalysis, one round of GOST is probably stronger than one round of DES, [38] and in 2000 Russian researchers claimed that as few as 7 rounds out of 32 are sufficient to protect GOST against differential cryptanalysis, see [19, 18]. In the same year two Japanese researchers [37], show that much more powerful differential attacks exist, if one joins several differential characteristics together [37]. This allows to break about 13 rounds of GOST.

In this paper we show that differential attacks with multiple differentials are much more powerful than expected. We show that the already known sets from [37] propagate with much higher probabilities than expected and allow for example to easily distinguish as many as 12 rounds from a random permutation, see Fig. 2. A distinguisher for 16 rounds was already described in [9]. We extend previous results with suitable initial and final extensions. Our best combined set of differentials allows to distinguish 20 rounds of GOST from a random permutation, see Fact 7.1.1. From here, given the particular ordering of round keys in GOST, which is very helpful for the attacker, we develop a first differential attack on full 32-round GOST. Our best current attack requires 2^{64} KP and allows to break full 32-round GOST in time of about 2^{226} GOST encryptions, which is faster than brute force.

Our current attack is just a sketch and a proof of concept. The exact complexity is not guaranteed. In the current attack, the dominating step is a filtering step in 2^{226} and we have been conservative in the extrapolations from our experimental results on aggregated differentials. For a broad range of changes in the probability obtained in Fact 7.1.1, the total complexity of our attack will not change. Therefore our result is presumably correct but it requires further work on Fact 7.1.1.

Better differential attacks on GOST with more detailed study and analysis and with more sophisticated distinguishers can be developed. It is also possible to see that the attacks such as described in this paper can also work with less than 2^{64} KP, this at the cost of increasing the time complexity.

Our result in 2^{226} is almost as fast as the fastest attack published as of today which is in 2^{225}, see [26, 8]. Our attack requires much less memory, 2^{64} instead of 2^{128}. Our new attack is faster than any attack on GOST which requires 2^{64} or memory which was published so far, However many other attacks which are faster, require less memory, or both, have already been announced and at this moment are not yet public, see [8, 15].

References

[1] Eli Biham, Vladimir Furman, Michal Misztal, and Vincent Rijmen: *Differential Cryptanalysis of Q,* In FSE 2002, LNCS 2355, Springer-Verlag, 2002.

[2] Eli Biham, Orr Dunkelman, Nathan Keller: *Improved Slide Attacks,* In FSE 2007, LNCS 4593 Springer 2007, pp. 153-166.

[3] A. Biryukov, D.Wagner: *Slide Attacks,* In proceedings of FSE'99, LNCS 1636, pp. 245-259, Springer, 1999.

[4] Alex Biryukov, David Wagner: *Advanced Slide Attacks,* In Eurocrypt 2000, LNCS 1807, pp. 589-606, Springer 2000.

[5] Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems,* Journal of Cryptology, vol. 4, pp. 3-72, IACR, 1991.

[6] Eli Biham, Adi Shamir, *Differential cryptanalysis of the full 16-round DES,* In Crypto'92, pp. 487-496, LNCS 740, Springer-Verlag, 1992.

[7] Christophe De Cannière: *GOST article,* In ENCYCLOPEDIA OF CRYPTOGRAPHY AND SECURITY 2005, pp. 242-243.

[8] Nicolas Courtois: *Security Evaluation of GOST 28147-89 In View Of International Standardisation,* document officially submitted to ISO in May 2011, In Cryptology ePrint Archive, Report 2011/211. 1 May 2011, `http://eprint.iacr.org/2011/211/`.

[9] Nicolas Courtois, Michał Misztal: *Aggregated Differentials and Cryptanalysis of PP-1 and GOST,* To appear in 11th Central European Conference on Cryptology, will be held in Debrecen, Hungary, on June 30 - July 2, 2011.

[10] Nicolas Courtois, Theodosis Mourouzis: *Black-Box Collision Attacks on the Compression Function of the GOST Hash Function,* accepted as SHORT paper, to appear in 6th International Conference on Security and Cryptography SECRYPT 2011, 18-21 July, Seville, Spain.

[11] Nicolas Courtois and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,* Asiacrypt 2002, LNCS 2501, pp.267-287, Springer.

[12] Nicolas Courtois: *General Principles of Algebraic Attacks and New Design Criteria for Components of Symmetric Ciphers*, in AES 4, LNCS 3373, pp. 67-83, Springer, 2005.

[13] Nicolas Courtois, Gregory V. Bard: *Algebraic Cryptanalysis of the Data Encryption Standard*, In Cryptography and Coding, 11-th IMA Conference, pp. 152-169, LNCS 4887, Springer, 2007. Preprint available at `eprint.iacr.org/2006/402/`. Also presented at ECRYPT workshop Tools for Cryptanalysis, Krakow, 24-25 September 2007.

[14] Nicolas Courtois, Gregory V. Bard and Andrey Bogdanov: *Periodic Ciphers with Small Blocks and Cryptanalysis of KeeLoq*, In Tatra Mountains Mathematic Publications, 41 (2008), pp. 167-188, post-proceedings of Tatracrypt 2007 conference, The 7th Central European Conference on Cryptology, June 22-24, 2007, Smolenice, Slovakia.

[15] Nicolas Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST*, 17 February 2011, 28 pages, original preprint submitted to Crypto 2011. MD5 Hash is d1e272a75601405d156618176cf98218. SHA-1 Hash is 6C16C46E 00AFD74B 3ED4949B 7766D5BF 6EC7DDBB. The fastest attack on full-round 256-bit GOST presented in this paper has a time complexity of 2^{216}. The paper also contains one nearly-practical attack on a well-known practical variant of GOST which allows to break some keys in practice.

[16] Fleischmann Ewan, Gorski Michael, Huehne Jan-Hendrik, Lucks Stefan: *Key recovery attack on full GOST block cipher with zero time and memory*, Published as ISO/IEC JTC 1/SC 27 N8229. 2009.

[17] Soichi Furuya: *Slide Attacks with a Known-Plaintext Cryptanalysis*, In ICISC 2001, LNCS 2288, 2002, pp. 11-50.

[18] Vitaly V. Shorin, Vadim V. Jelezniakov and Ernst M. Gabidulin: *Linear and Differential Cryptanalysis of Russian GOST*, Preprint submitted to Elsevier Preprint, 4 April 2001

[19] V.V. Shorin, V.V. Jelezniakov, E.M. Gabidulin *Security of algorithm GOST 28147-89*, (in Russian), In Abstracts of XLIII MIPT Science Conference, December 8-9, 2000.

[20] I. A. Zabotin, G. P. Glazkov, V. B. Isaeva: *Cryptographic Protection for Information Processing Systems*, Government Standard of the USSR,

GOST 28147-89, Government Committee of the USSR for Standards, 1989. In Russian, translated to English in [21].

[21] An English translation of [20] by Aleksandr Malchik with an English Preface co-written with Whitfield Diffie, can be found at `http://www.autochthonous.org/crypto/gosthash.tar.gz`

[22] Vasily Dolmatov, Editor, RFC 5830: *GOST 28147-89 encryption, decryption and MAC algorithms,* IETF. ISSN: 2070-1721. March 2010. `http://tools.ietf.org/html/rfc5830`

[23] V. Popov, I. Kurepkin, S. Leontie: *RFC 4357: Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms,* IETF January 2006. `http://tools.ietf.org/html/rfc4357`

[24] A Russian reference implementation of GOST implementing Russian algorithms as an extension of TLS v1.0. is available as a part of OpenSSL library. The file gost89.c contains eight different sets of S-boxes and is found in OpenSSL 0.9.8 and later: `http://www.openssl.org/source/`

[25] Florian Mendel, Norbert Pramstaller, Christian Rechberger, Marcin Kontak and Janusz Szmidt: *Cryptanalysis of the GOST Hash Function,* In Crypto 2008, LNCS 5157, pp. 162 - 178, Springer, 2008.

[26] Takanori Isobe: *A Single-Key Attack on the Full GOST Block Cipher,* In FSE 2011, Fast Software Ecnryption, Springer LNCS, 2011.

[27] Orhun Kara: *Reflection Cryptanalysis of Some Ciphers,* In Indocrypt 2008, LNCS 5365, pp. 294-307, 2008.

[28] David Kahn, *The Codebreakers,* The Comprehensive History of Secret Communication from Ancient Times to the Internet, First published in 1967, new chapter added in 1996.

[29] John Kelsey, Bruce Schneier, David Wagner: *Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES,* In Crypto'96, LLNCS 1109,

[30] Axel Poschmann, San Ling, and Huaxiong Wang: *256 Bit Standardized Crypto for 650 GE GOST Revisited,* In CHES 2010, LNCS 6225, pp. 219-233, 2010.

[31] J. Pieprzyk and L. Tombak, *Soviet Encryption Algorithm*, Preprint 94-10, Department of Computer Science, The University of Wollongong, 1994

[32] C. Charnes, L. O'Connor, J. Pieprzyk, R. Savafi-Naini, Y. Zheng: *Further comments on GOST encryption algorithm*, Preprint 94-9, Department of Computer Science, The University of Wollongong, 1994.

[33] C. Charnes, L. O'Connor, J. Pieprzyk, R. Savafi-Naini, Y. Zheng: *Comments on Soviet encryption algorithm*, In Advances in Cryptology - Eurocrypt'94 Proceedings, LNCS 950, A. De Santis, ed., pp. 433-438, Springer, 1995.

[34] Vincent Rijmen: *Cryptanalysis and design of iterated block ciphers*, PhD Thesis, October 1997, K.U.Leuven.

[35] Vladimir Rudskoy: *On zero practical significance of Key recovery attack on full GOST block cipher with zero time and memory*,

[36] Markku-Juhani Saarinen: *A chosen key attack against the secret S-boxes of GOST*, unpublished manuscript, 1998.

[37] Haruki Seki and Toshinobu Kaneko: *Differential Cryptanalysis of Reduced Rounds of GOST. In SAC 2000, Selected Areas in Cryptography, Douglas R. Stinson and Stafford E. Tavares, editors, LNCS 2012, pp. 315323, Springer, 2000.*

[38] Bruce Schneier: Section 14.1 GOST, in *Applied Cryptography*, Second Edition, John Wiley and Sons, 1996. ISBN 0-471-11709-9.

[39] Bruce Schneier, *The GOST Encryption Algorithm*, Dr. Dobbs Journal, Vol. 20, No. 2, 1995.

[40] Standard Deviation – wikipedia article, 13 June 2011, available at `http://en.wikipedia.org/wiki/Standard_deviation`.

[41] Wei Dai: Crypto++, a public domain library containing a reference C++ implementation of GOST and test vectors, `http://www.cryptopp.com`