New Linear Attacks on Block Cipher GOST

Yi LU

National Research Center of Fundamental Software, Beijing, P.R.China
Department of Informatics, University of Bergen, Bergen, Norway
luyi666@gmail.com

Abstract. Defined in the standard GOST 28147-89, GOST is a Soviet and Russian government standard symmetric-key block cipher. GOST has the 64-bit block size and a key length of 256 bits. It is a Feistel network of 32 rounds. In 2010, GOST was submitted to ISO 18033 to become a worldwide industrial encryption standard. GOST 28147-89 has also been published as informational RFC 5830 with IETF.

In this paper, we study linear attacks on GOST 28147-89. Prior to us, [14] did some analysis on the linear approximation of GOST without giving any detailed results. [14] claimed that the complexity of the linear attack on GOST is higher than 2^{256} after 5 rounds. In our work, we show that this is not true. First, we give the detailed bias analysis on the GOST round function for the first time. We show that the largest bias is 2^{-7}. Secondly, we proposed the first known linear attacks on GOST. The recent idea of synthetic linear analysis [9] is then successfully applied to improve the bias for the r-round linear approximation of GOST. In summary, our attack on 8-round GOST recovers the key in time 2^{37} with 2^{50} known plaintexts in the single-key setting. For the 16-round GOST with last 8 rounds using subkeys in reverse order, our distinguishing attack works in time 2^{85} using 2^{85} known plaintexts, in the plain multiple-key setting without the related-key assumption. That is, the plaintexts can be encrypted by arbitrary number of keys, with each key encrypting arbitrary number of plaintexts, as long as we have a total of 2^{85} known plaintexts. For the 32-round GOST with the slightly tweaked key schedule, i.e., assuming last 16 rounds using subkeys in reverse order, our distinguishing attack works in time $2^{170.8}$, given $2^{170.8}$ known plaintexts, in the plain multiple-key setting without the related-key assumption. To the best of our knowledge, our distinguishing attacks are the first known distinguishers on block ciphers in the plain multiple-key setting without the usual related-key assumption. Finally, for the 32-round GOST with the original key schedule, our distinguisher works in time $2^{173.8}$, given $2^{173.8}$ known plaintexts, in the related-key setting. This is the fastest attack known so far, compared with the best attacks [4, 3] on the full 32-round GOST.

Keywords: block cipher, GOST, Feistel network, bias, linear analysis, distinguishing attack, plain multiple-key setting

1 Introduction

Defined in the standard GOST 28147-89, the cipher GOST [12, Chapter 14.1] is a Soviet and Russian government standard symmetric-key block cipher. Note that the GOST hash function (c.f. [11]) is also based on this block cipher. Developed in the 1970s, the standard had been marked "Top Secret" and then downgraded to "Secret" in 1990. Shortly after the dissolution of the USSR, it was declassified and it was released to the public in 1994. GOST 28147 was a Soviet alternative to the United States standard algorithm DES. The two are very similar in structure. In 2010, GOST was submitted to ISO 18033 to become a worldwide industrial encryption standard. GOST 28147-89 has also been published [5] as informational RFC 5830 with the Internet Engineering Task Force (IETF), which is the main standardization body for Internet technology. The GOST cipher has a 64-bit block size and a key length of 256 bits. It is a Feistel network of 32 rounds. Recent attacks on GOST can be found in [8, 14, 2–4, 6, 7, 13].

In this paper, we study linear attacks on GOST 28147-89. Prior to us, [14] did some analysis on the linear approximation of GOST without giving any detailed results. It was claimed in [14] that GOST is secure against the linear analysis after 5 rounds out of 32 rounds (i.e., the complexity of the linear attack was considered to be higher than 2^{256}), with the S-boxes used in the Central Bank of the Russian Federation. In our work, we show that this is *not* true. First, we give the detailed bias analysis on the GOST round function for the first time. We show that the largest bias is 2^{-7}. Based on our bias analysis on the GOST round function, we proposed the first known linear attacks on GOST. Recently, the idea of synthetic linear analysis [9] was proposed to study the combined bias of multiple Boolean functions (such as multiple linear approximations), when some input terms of the Boolean functions are dependent and Piling-up Lemma might not be appropriate. We successfully apply the technique of synthetic linear analysis [9] to improve the bias for the r-round linear approximation of GOST, which is actually the combined bias of multiple linear approximations (from multiple rounds).

In summary, our linear attack on 8-round GOST recovers the key in time 2^{37} with 2^{50} known plaintexts in the single-key setting. For the 16-round GOST with last 8 rounds using subkeys in reverse order, our distinguishing attack works in time 2^{85}, given 2^{85} known plaintexts, in the

plain multiple-key setting[1] without the related-key assumption. For the 32-round GOST with the slightly tweaked key schedule, i.e., assuming last 16 rounds using subkeys in reverse order, our distinguishing attack works in time $2^{170.8}$, given $2^{170.8}$ known plaintexts, in the plain multiple-key setting without the related-key assumption. To the best of our knowledge, our proposed attacks are the first known distinguishing attacks on block ciphers in the plain multiple-key setting without the usual related-key assumption. Finally, for the 32-round GOST with the original key schedule, i.e., assuming last 8 rounds using subkeys in reverse order, our distinguishing attack works in time $2^{173.8}$, given $2^{173.8}$ known plaintexts, in the related-key setting. This is the fastest attack known so far, compared with the best attacks [4, 3] on 32-round GOST with the original key schedule.

The rest of the paper is organized as follows. In Sect. 2, we give short description on GOST. We give bias analysis on GOST round function in Sect. 3. We propose our first linear attacks on GOST in Sect. 4. We study the synthetic linear analysis on GOST in Sect. 5 and give the improved attacks. We conclude in Sect. 6.

2 Preliminaries on GOST

GOST has a 64-bit block size and a key length of 256 bits. GOST is a Feistel network of 32 rounds. Its round function is very simple. At each round, add a 32-bit subkey modulo 2^{32}, put the result through a layer of S-boxes, and rotate that result left by 11 bits. The result is the output of the round function. Let 32-bit L_i and R_i denote the left half and right half at Round i (let L_0, R_0 be the left half and right half of the plaintext). The subkey for Round i is k_i. The round function can be expressed by $L_i = R_{i-1}, R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$. Here,

$$f(R_{i-1}, k_i) = \text{S-Box}(R_{i-1} + k_i) \lll 11, \tag{1}$$

and the addition denotes the modulo addition.

There are eight different S-boxes in GOST. The S-boxes accept a 4-bit input and produce a 4-bit output. Each S-box is permutation of the numbers 0 through 15. The S-boxes are implementation dependent. For extra security, the S-boxes can be kept secret. Further, the GOST standard does discuss how to generate the S-boxes. Recently, a set of

[1] i.e., the plaintexts can be encrypted by arbitrary number of keys, with each key encrypting arbitrary number of plaintexts, as long as we have a total of 2^{85} known plaintexts.

S-boxes used in the Central Bank of the Russian Federation surfaced[2], according to [12, Chapter 14.1]. We give them in Table 4, Appendix. Note that this choice of S-boxes is exactly what most researchers call "the GOST cipher" in literature, according to [3]. The round subkeys are generated as follows. The 256-bit key is divided into eight 32-bit subkeys $k_1, k_2, \ldots, k_8$. Each subkey k_i is used four times in total for the full 32-round GOST; the first 24 rounds use the subkeys in order, while the last 8 rounds use them in reverse order. Table 1 shows the key schedule.

Table 1. The original key schedule for 32-round GOST

round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
subkey	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8

round	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
subkey	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_8	k_7	k_6	k_5	k_4	k_3	k_2	k_1

3 Bias Analysis on GOST Round Function

Given a a binary random variable X, we define the bias for X by

$$\text{bias}(X) \stackrel{\text{def}}{=} |\Pr(X = 0) - \frac{1}{2}|. \tag{2}$$

From Sect. 2, we note that the subkey mixing parts are different in Russian GOST from the famous DES. For DES, the subkey is XORed into the internal state and the subkey mixing is linear. In contrast, for GOST cipher, the subkey is added (modulo 2^{32}) to the internal state and thus the subkey mixing is nonlinear. Hence, for DES, linear cryptanalysis on the round function can be focused on the S-Boxes, which is the only nonlinear component within the round and which is not difficult to study due to the rather small input size of DES S-Boxes. For the round function of GOST cipher, however, we note that this subkey mixing difference makes linear cryptanalysis not an easy task.

Previously, [14] did some analysis on linear approximation of the GOST cipher without giving any detailed results. In this section, we will give the detailed analysis on the GOST round function f as defined in (1).

[2] These S-boxes are also used in the GOST hash function (c.f. [11]).

For simplicity, in this section, we let the two 32-bits inputs be denoted by R, k (i.e., we omit the subscripts i).

Assuming that the two inputs R, k are random and uniformly distributed, we can deduce that the output of f is random and uniformly distributed. The reason is as follows. First, we deduce that $(R + k)$ is random and uniformly distributed. As each S-box is a permutation over the 4-bit string, the 32-bit output of eight parallel S-boxes is also random and uniformly distributed. The final bit-wise rotation is a permutation over the 32-bit string, and so we deduce that the output of f is random and uniformly distributed. Consequently, we know that the bit $\beta \cdot f(R, k)$ is always balanced for all possible 32-bit nonzero output mask β.

From above, we see that for the round function f, it is necessary to examine the bias with the nonzero input masks. We propose to study the bias for the bit $\alpha \cdot R \oplus \alpha' \cdot k \oplus \beta \cdot f(R, k)$. Since the two inputs are mixed in f by the modulo addition operation, which is commutative, we consider the two input masks equal, i.e., $\alpha = \alpha'$. Further, to make it easy to iterate for consecutively many rounds of GOST, we focus on the bias analysis for the bit with equal input and output masks ($\alpha = \beta$), i.e., $\beta \cdot (R \oplus k \oplus f(R, k))$.

In order to find all sufficiently large biases, with N randomly and uniformly chosen pairs (R, k) (where the value of N is a parameter to be discussed later), we calculate the 32-bit $f'(R, k) = R \oplus k \oplus f(R, k)$ and we let $\mathcal{D}$ denote its distribution. The application of Walsh transform is known to compute all the biases simultaneously as follows. Recall that the Walsh transform for the function $F : GF(2)^\ell \to \mathbf{R}$ is defined as $\widehat{F}(x) = \sum_{x'} (-1)^{x \cdot x'} F(x')$, for $x \in GF(2)^\ell$. We can compute the Walsh transform for the distribution $\mathcal{D}$. And it is easy to see that $|\widehat{\mathcal{D}}(a)|$ is twice the value of the bias for $a \cdot f'(R, k)$, for any a. To show this, first, we have

$$\widehat{\mathcal{D}}(a) = \sum_b (-1)^{a \cdot b} \cdot \mathcal{D}(b) = \frac{1}{2^{64}} \sum_b (-1)^{a \cdot b} \cdot \sum_{R,k} 1_{f'(R,k)=b} \tag{3}$$

We swap the order of the two summations in (3), and we have

$$\widehat{\mathcal{D}}(a) = \frac{1}{2^{64}} \sum_{R,k} \sum_b (-1)^{a \cdot f'(R,k)} \cdot 1_{f'(R,k)=b} \tag{4}$$

$$= \frac{1}{2^{64}} \sum_{R,k} (-1)^{a \cdot f'(R,k)} \sum_b 1_{f'(R,k)=b} \tag{5}$$

We know that $\sum_b 1_{f'(R,k)=b} = 1$ holds true for any fixed (R, k). Consequently, we have

$$\widehat{\mathcal{D}}(a) = \frac{1}{2^{64}} \sum_{R,k} (-1)^{a \cdot f'(R,k)}, \tag{6}$$

which equals $\Pr(a \cdot f'(R,k) = 0) - \Pr(a \cdot f'(R,k) = 1)$, that is, $2(\Pr(a \cdot f'(R,k) = 0) - 1/2)$. By our definition on bias in (2), we have justified $|\widehat{\mathcal{D}}(a)|$ is twice the value of the bias for $a \cdot f'(R,k)$.

Now we discuss the sampling number N. Recall from coding theory, we know that if the bias of the bit A is d, then we can successfully distinguish the distribution of randomly and uniformly chosen $1/d^2$ samples of A from uniform distribution with probability of success higher than $1/2$. Hence, we choose $N = 2^{40}$. And all those biases which are sufficiently large (e.g., larger than 2^{-16}) can be directly detected by searching through our calculated big table of $\widehat{\mathcal{D}}$.

Our computations found out that the largest bias 2^{-7} is obtained when $\beta \in \{0\text{x}802001, 0\text{x}803007, 0\text{x}806001\}$. Other notably large biases are: the bias is $2^{-7.2}$ when $\beta = 0\text{x}806006$, and the bias is $2^{-7.3}$ when $\beta = 0\text{x}802007$. In Table 5, Appendix, we list all the biases, which are no smaller than 2^{-9}.

4 Our First Attacks on GOST

In this section, we will study attacks on GOST, based on our bias analysis results on GOST round function in last section.

4.1 Attacks on 8-round GOST

We can immediately transfer our results for attacks on 8-round GOST as follows. Let $R \oplus k \oplus f(R,k)$ have the bias d with the mask β, which were discussed following Sect. 3. Given the mask β, we can iterate this one-round linear approximation for 8-round GOST. This is illustrated in Figure 1, Appendix, which is shown in the untwisted Feistel structure. Note that the subkey mixing is omitted in Figure 1; and whenever the input mask is β for Round i in Figure 1, it means that the mask for the subkey of that round is also β. For the linear approximation of 8-round GOST, Round 1, Round 4 and Round 7 involve no linear approximation: the input and output masks are all marked "0" in Figure 1. By the famous Piling-up Lemma [10], we can check that the 8-round linear approximation

$$\beta \cdot (R_0 \oplus L_8 \oplus R_8) \approx \beta \cdot (k_2 \oplus k_3 \oplus k_5 \oplus k_6 \oplus k_8) \tag{7}$$

has bias $d' = (2d)^5/2 = 16 \cdot d^5$.

Thus, if we take the largest one-round bias $d = 2^{-7}$ by choosing the mask $\beta \in \{0x802001, 0x803007, 0x806001\}$, we know that the 8-round linear approximation (7) has the bias $d' = 2^{-31}$. This means that given $1/d'^2$ (i.e., 2^{62}) pairs of known plaintexts and ciphertexts for 8-round GOST, we compute the left-hand side of (7) for each pair, make a majority vote on them and we can successfully recover one key bit, i.e., the right-hand side of (7). Note that above is actually Matsui's Algorithm 1 [10], which uses 8-round linear approximation to attack 8-round GOST. If we use Matsui's Algorithm 2 [10], we can have better attack results on 8-round GOST. That is, we use 7-round linear approximation rather than 8-round approximation. It is easy to see from Figure 1 that the bias d'' for 7-round GOST is calculated as $d'' = (2d)^4/2 = 8 \cdot d^4 = 2^{-25}$ with $d = 2^{-7}$. Thus, with data amount 2^{50}, we can recover the subkey in the last round. This takes time $2^{32} \times 32 = 2^{37}$, by the optimized algorithm of Matsui's Algorithm 2 [1]. Then, we can recover the other subkeys sequentially with less data after that. The attacks are the first known linear attacks on GOST to the best of our knowledge.

4.2 Attacks on Higher Rounds of GOST

Obviously, we can iterate the one-round approximation for higher rounds of GOST. For instance, for the 10-round linear approximation of GOST, the same one-round linear approximation is iterated six times in total and the bias is calculated as $d' = (2d)^6/2 = 32 \cdot d^6$. With the largest one-round bias $d = 2^{-7}$, we have the bias $d' = 2^{-37}$ for 10-round approximation of GOST. It implies the data complexity $1/d'^2 = 2^{74}$, which is above the maximum limit 2^{64} for a single key. In this subsection, we propose attacks in the simple setting of multiple keys (without the related-key assumption), which allows to obtain more than 2^{64} known plaintext and ciphertext pairs (that are generated by multiple keys). First, we consider the distinguishing attack on 16-round GOST, whose last eight rounds use subkeys in reverse order. We can easily extend the 8-round linear approximation (7) for 16 rounds. We obtain the following 16-round linear approximation,

$$\begin{aligned} \beta \cdot (R_0 \oplus L_{16}) \approx \beta \cdot (&k_2 \oplus k_3 \oplus k_5 \oplus k_6 \oplus k_8 \oplus \\ &k_9 \oplus k_{11} \oplus k_{12} \oplus k_{14} \oplus k_{15}) \end{aligned} \tag{8}$$

where we let k_i denote the subkey for Round i, for $i = 1, 2, \ldots, 16$. As the last eight rounds of the reduced 16-round GOST use subkeys in reverse

order, i.e., $k_9 = k_8, k_{10} = k_7, k_{11} = k_6, \ldots, k_{16} = k_1$, the right-hand side of the linear approximation (8) is the constant 0, which does not depend on the key. This is *critical* for our distinguishing attack to work in the plain setting of multiple keys, where we do not need to make the related key assumption on these keys. Consequently, we know the 16-round linear approximation $\beta \cdot (R_0 \oplus L_{16}) \approx 0$ has the bias $d' = (2d)^{10}/2 = 2^9 \cdot d^{10} = 2^{-61}$ with $d = 2^{-7}$. Thus, we have the distinguishing attack on 16-round GOST with data amount 2^{122}, which can be obtained from multiple keys (and our attack is not affected by the aforementioned limit 2^{64} for the single key).

For 32-round GOST, we first consider the slightly tweaked key schedule, that is, the last 16 rounds use subkeys in reverse order. Similarly as we do for the 16 rounds, we have the 32-round linear approximation

$$\begin{aligned} & \beta \cdot (L_0 \oplus R_{32}) \qquad (9) \\ \approx\ & \beta \cdot (k_1 \oplus k_2 \oplus k_4 \oplus k_5 \oplus k_7 \oplus k_8 \oplus k_{10} \oplus k_{11} \oplus k_{13} \oplus k_{14} \oplus k_{16} \oplus \\ & k_{17} \oplus k_{19} \oplus k_{20} \oplus k_{22} \oplus k_{23} \oplus k_{25} \oplus k_{26} \oplus k_{28} \oplus k_{29} \oplus k_{31} \oplus k_{32}). \end{aligned}$$

As the last 16 rounds use subkeys in reverse order, we apply $k_{17} = k_{16}, k_{18} = k_{15}, \ldots, k_{32} = k_1$, and the right-hand side of above 32-round linear approximation is constant 0. So, we know the 32-round linear approximation $\beta \cdot (L_0 \oplus R_{32}) \approx 0$ has the bias $d' = (2d)^{22}/2 = 2^{21} \cdot d^{22} = 2^{-133}$ with $d = 2^{-7}$. Thus, we have the distinguishing attack on 32-round GOST with data amount 2^{266}, which are obtained from multiple keys.

Meanwhile, we note that for the full 32-round GOST with the original key schedule, in order for our linear attack to work, we need to make the related-key assumption. We can use the following 32-round linear approximation (which has larger bias than above as shown later), $\beta \cdot (R_0 \oplus L_{32} \oplus R_{32}) \approx \beta \cdot (k_1 \oplus k_3 \oplus k_4 \oplus k_6 \oplus k_7)$. Assuming that $k_1 \oplus k_3 \oplus k_4 \oplus k_6 \oplus k_7$ is a constant for all the keys, this 32-round linear approximation has bias $d' = (2d)^{21}/2 = 2^{20} \cdot d^{21} = 2^{-127}$. So, our distinguishing attack would work with 2^{254} known plaintext and ciphertext pairs generated by the multiple related keys. In the next section, we will study the improved bias analysis to improve the attack results.

5 Synthetic Linear Analysis on GOST

Recently, the idea of synthetic linear analysis [9] was proposed in order to study the combined bias of multiple Boolean functions (such as multiple linear approximations), when some input terms of the Boolean functions

are not statistically independent and Piling-up Lemma [10] might not be an appropriate approximation. When multiple (possibly dependent) Boolean functions are involved, [9] proposed to group dependent ones together and make independent groups. For each group, the combined bias is enlarged (compared with the total bias for all groups), which makes bias analysis easier (and possible). The total bias can be finally obtained from the combined bias of each independent group by Piling-up Lemma.

Recall that when we analyze the total bias for r-round linear approximation of GOST in Sect. 4, we assume that each round is independent of the others, and we apply the Piling-up Lemma to combine the biases for the linear approximations of each active round (i.e., the round with input mask $\beta \neq 0$). In particular, at each active round i, with the fixed mask β, the linear approximation is expressed by

$$\beta \cdot (R_i \oplus k_i \oplus f(R_i, k_i)) \approx 0. \tag{10}$$

Note that computing the individual bias for (10), assuming that R_i, k_i are random with uniform distribution, is discussed in Sect. 3.

In this section, we consider the dependency between multiple linear approximations of different active rounds and we apply the synthetic bias analysis [9] to improve the total combined bias of the r-round linear approximation. Let us start from the simple case of the combined bias for the two linear approximations (from two active rounds[3]). For the linear approximation (10) at the active round i, the input terms are (R_i, k_i). Given two active rounds i, j with $i \neq j$, we have the input terms (R_i, k_i) and (R_j, k_j) respectively. We focus on the dependency between the two input pairs. If we assume all the subkeys are random with uniform distribution, then, k_i, k_j are statistically independent. And it is most possible that R_i, R_j are dependent, which could make the combined bias stronger. We use the technique of Sect. 3 to first compute the bias for

$$R_i \oplus k_i \oplus f(R_i, k_i) \oplus R_j \oplus k_j \oplus f(R_j, k_j) \tag{11}$$

for all mask β, where $j = i+1$ (i.e., two consecutive rounds) and L_i, R_i, k_i, k_j are randomly chosen with uniform distribution. Our results show that Piling-up lemma gives fairly good approximation. We then tried for close i, j, (i.e., $j = i + 2, i + 3, i + 4$). And we did not find any improved bias results, compared with the simple Piling-up lemma approximation.

In above analysis, we assume that all the subkeys are random with uniform distribution. Now, we study the case in which this assumption is

[3] The two rounds may be not consecutive.

removed. Due to the simple key schedule of GOST, we notice that it is possible to have $k_i = k_j$ for $i \neq j$. This initiates us to compute the bias for (11) for all mask β, where $k_i = k_j$ and R_i, R_j, k_i are randomly chosen with uniform distribution. Interestingly, our results show that when $\beta =$ 0x806006, 0x400807, the bias is the largest $2^{-9.3}$. Note that Piling-up lemma approximation gives much smaller biases of $2d^2$, i.e., $2^{-13.4}, 2^{-16.2}$ respectively, (with $d = 2^{-7.2}, 2^{-8.6}$, which can be checked by Table 5). In Table 2, we compare the other notably large biases with the Piling-up lemma approximation results (in the last row of Table 2), where '-' indicates that the corresponding bias is smaller than 2^{-16}.

Table 2. Other large biases for (11), where $k_i = k_j$ and R_i, R_j, k_i are randomly chosen with uniform distribution, in comparison with Piling-up lemma approximation results

mask	0x400802	0x500802	0x500807	0x30080500	0x700802	0x700807	0x804003
bias	$2^{-10.7}$	$2^{-10.5}$	$2^{-10.5}$	$2^{-10.8}$	$2^{-11.1}$	$2^{-11.1}$	$2^{-11.1}$
d	$2^{-8.6}$	2^{-8}	-	$2^{-7.8}$	$2^{-9.7}$	-	-
$2d^2$	$2^{-16.2}$	2^{-15}	-	$2^{-14.6}$	$2^{-18.4}$	-	-

We have just studied the combined bias for two active rounds. For the combined bias for three active rounds, i.e., $R_i \oplus k_i \oplus f(R_i, k_i) \oplus R_j \oplus k_j \oplus f(R_j, k_j) \oplus R_m \oplus k_m \oplus f(R_m, k_m)$ with different i, j, m, we did similar analysis. Our results show that assuming $k_i = k_j = k_m$ and R_i, R_j, R_m, k_i are randomly chosen with uniform distribution, then, we have the (only) largest bias $2^{-13.2}$ with the mask $\beta =$ 0x806006. In comparison, note that Piling-up lemma approximation gives much smaller bias $4d^3$, i.e., $2^{-19.6}$ with $d = 2^{-7.2}$ (which can be checked by Table 5). Similarly as done for the case of two active rounds, we also analyzed the combined bias when $i < j < m$ and i, j, m are close, and L_i, R_i, k_i, k_j, k_m are randomly chosen with uniform distribution. We did not find any new results.

For the combined bias for four active rounds, i.e., $R_i \oplus k_i \oplus f(R_i, k_i) \oplus R_j \oplus k_j \oplus f(R_j, k_j) \oplus R_m \oplus k_m \oplus f(R_m, k_m) \oplus R_n \oplus k_n \oplus f(R_n, k_n)$, with different i, j, m, n, our analysis shows that assuming $k_i = k_j = k_m = k_n$ and R_i, R_j, R_m, R_n, k_i are randomly chosen with uniform distribution, we have two largest biases $2^{-15.5}, 2^{-15.3}$ with mask $\beta =$ 0x806006, 0x400807 respectively. On the other hand, the Piling-up lemma approximation gives much smaller biases of $8d^4$, i.e., $2^{-25.8}, 2^{-31.4}$ with $d = 2^{-7.2}, 2^{-8.6}$ respectively.

5.1 The Improved Attacks on GOST

We now apply our above results to our attacks in Sect. 4. For the 8-round attack in Sect. 4.1, we know that the subkeys are all random with uniform distribution, and so we deduce that the attack results cannot be improved.

We check our attack in Sect. 4.2 on the reduced 16-round GOST, where the last eight rounds use subkeys in reverse order as the first eight rounds. The 16-round linear approximation (8) involves linear approximations of 10 active rounds, i.e., round $2, 3, 5, 6, 8,\ 9, 11, 12, 14, 15$. We group the linear approximations of the 10 rounds into five groups: (round 2, round 15), (round 3, round 14), (round 5, round 12), (round 6, round 11), (round 8, round 9). The two linear approximations at the same group use the same subkey for the round function, and are strongly dependent by our above analysis. Meanwhile, the linear approximations with each from a different group, can be assumed to be (almost) independent, as all the subkeys involved now are random with uniform distribution[4], and our above analysis shows in this case we can assume the other inputs (i.e., R_i's) of the round function are independent even if their round numbers are close.

To get the combined bias for the not all independent ten linear approximations (i.e., the five groups), we first calculate the combined bias for each group. Recall that we have shown when we choose the mask $\beta \in \{0\text{x}806006, 0\text{x}400807\}$, the bias for each group is the largest $d = 2^{-9.3}$. Then, we apply Piling-up Lemma to calculate the total bias for the five groups, i.e., the bias for the 16-round linear approximation (8) is $(2d)^5/2 = 16d^5 = 2^{-42.5}$. This greatly improves our previous estimated bias 2^{-61} in Sect. 4.2, which assumes that the linear approximations are all independent. And it implies our improved distinguishing attack on 16-round GOST needs data 2^{85}.

For our 32-round attack with the slightly tweaked key schedule in Sect. 4.2, the 32-round linear approximation (9) involves 22 active rounds. We can group them into 8 groups: (round 1, round 32), (round 2, round 10, round 23, round 31), (round 11, round 22), (round 4, round 29), (round 5, round 13, round 20, round 28), (round 14, round 19), (round 7, round 26), (round 8, round 16, round 17, round 25). Similar arguments hold true to show that the linear approximations at the same group use the same subkey for the round function, and are strongly dependent; meanwhile, the linear approximations with each from a different group, can

[4] because each 32-bit subkey is extracted from a different (i.e., non-overlapping) part of the 256-bit key.

be assumed to be (almost) independent. When we choose the best mask β = 0x400807, the bias for the group with two rounds is $2^{-9.3}$ and the bias for the group with four rounds is $2^{-15.3}$, according to our previous synthetic bias analysis. Therefore, the bias for the 32-round linear approximation (9) can be calculated by combining the biases of the eight independent groups, $1/2 \times (2 \times 2^{-9.3})^5 \times (2 \times 2^{-15.3})^3 = 2^{-85.4}$. This means the 32-round attack with the improved complexity $2^{170.8}$. In comparison, note that our previous result (in Sect. 4.2) has much higher complexity 2^{266}, which is based on the assumption that the linear approximations are all independent.

For our proposed distinguishing attack (in the related-key setting[5]) on the full 32-round GOST with the original key schedule in Sect. 4.2, we can similarly improve the bias from previous estimate 2^{-127} to $2^{-86.9}$ by choosing the mask $\beta = 0x806006$. This gives the greatly reduced attack complexity $2^{173.8}$. It is the fastest attack known so far, compared with the best attacks [4, 3] on the full 32-round GOST with the original key schedule. Finally, we summarize our linear attack results in Table 3.

6 Conclusion

In this paper, we study linear attacks on GOST 28147-89. We give the detailed bias analysis on the GOST round function for the first time. We show that the largest bias is 2^{-7}. Secondly, we propose the first known linear attacks on GOST. Then, the recent technique of synthetic linear analysis [9] is successfully applied to improve the bias for the r-round linear approximation of GOST. Our proposed linear attacks are the first known distinguishing attacks on block ciphers in the *plain multiple-key setting* without the usual related-key assumption to the best of our knowledge. Finally, for the full 32-round GOST with the original key schedule, our distinguishing attack works in time $2^{173.8}$, given $2^{173.8}$ known plaintexts, in the related-key setting. This is the fastest attack known so far, compared with the best attacks [4, 3] on 32-round GOST with the original key schedule. Meanwhile, our results also show that the early statement [14] that GOST is secure against the linear analysis after 5 rounds out of 32 rounds is *not* true.

[5] i.e., assuming that $k_1 \oplus k_3 \oplus k_4 \oplus k_6 \oplus k_7$ is a constant for all the keys

Table 3. Summary of our linear attacks on r-round GOST

attack	r	type	setting	data	time	key schedule notes
[13]	13	differential key-recovery	single key (chosen-plaintext)	2^{51}	unknown	-
[13]	21	differential key-recovery	related key (chosen-plaintext)	2^{56}	unknown	-
[3]	32	differential key-recovery	single key	2^{64}	2^{178}	original key schedule
[4]	32	key-recovery	single key	2^{64}	2^{192}	original key schedule
[4]	32	key-recovery	single key	2^{32}	2^{224}	original key schedule
ours	8	linear key-recovery	single key	2^{50}	2^{37}	-
ours	16	linear distinguisher	multiple keys (no related-key)	2^{85}	2^{85}	last eight rounds use subkeys in reverse order
ours	32	linear distinguisher	multiple keys (no related-key)	$2^{170.8}$	$2^{170.8}$	last 16 rounds use subkeys in reverse order
ours	32	linear distinguisher	multiple keys (related-key)	$2^{173.8}$	$2^{173.8}$	original key schedule

References

1. B. Collard, F. -X. Standaert, Jean-Jacques Quisquater, *Improving the time complexity of Matsuis linear cryptanalysis*, ICISC 2007, LNCS vol. 4817, pp. 77-88, 2007.
2. N. Courtois, *Security evaluation of GOST 28147-89 in view of international standardisation*, IACR eprint, available online at `http://eprint.iacr.org/2011/211`, 2011.
3. N. Courtois, *An improved differential attack on full GOST*, IACR eprint, available online at `http://eprint.iacr.org/2012/138`, 2012.
4. I. Dinur, O. Dunkelman, A. Shamir, *Improved attacks on full GOST*, to appear in the proceedings of FSE 2012, preprint version available online at `http://eprint.iacr.org/2011/558`.
5. V. Dolmatov, Ed., *GOST 28147-89: encryption, decryption, and message authentication code (MAC) algorithms*, RFC 5830, IETF, ISSN: 2070-1721, available online at `http://www.ietf.org/rfc/rfc5830.txt?number=5830`, March 2010.
6. T. Isobe, *A single-key attack on the full GOST block cipher*, FSE 2011, LNCS vol. 6733, pp. 290-305, 2011.
7. O. Kara, *Reflection cryptanalysis of some ciphers*, INDOCRYPT 2008, LNCS vol. 5365, pp. 294-307, 2008.
8. J. Kelsey, B. Schneier, D. Wagner, *Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and tripe-DES*, CRYPTO 1996, LNCS vol. 1109, pp. 237-251, 1996.
9. Y. Lu, S. Vaudenay, W. Meier, *Synthetic Linear Analysis with Applications to CubeHash and Rabbit*, Cryptography and Communications, vol. 4, No. 3-4, Springer, pp. 259-276, 2012.

10. M. Matsui, *Linear cryptanalysis method for DES cipher*, EUROCRYPT 1993, LNCS vol. 765, pp. 386-397, 1994.
11. F. Mendel, N. Pramstaller, C. Rechberger, M. Kontak, J. Szmidt, *Cryptanalysis of the GOST hash function*, CRYPTO 2008, LNCS vol. 5157, pp. 162-178, 2008.
12. B. Schneier, *Applied Cryptography - Protocols, Algorithms, and Source Code in C*, second edition, John Wiley & Sons, 1996.
13. H. Seki, T. Kaneko, *Differential cryptanalysis of reduced rounds of GOST*, SAC 2000, LNCS vol. 2012, pp. 315-323, 2001.
14. Vitaly V. Shorin, Vadim V. Jelezniakov, Ernst M. Gabidulin, *Linear and Differential Cryptanalysis of Russian GOST*, Electronic Notes in Discrete Mathematics, vol. 6, pp. 538-547, April 2001.
15. A. J. Menezes, P. C. van. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.

Appendix

Table 4. GOST S-Boxes

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_1(x)$	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_2(x)$	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_3(x)$	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_4(x)$	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_5(x)$	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_6(x)$	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_7(x)$	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_8(x)$	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Table 5. Biases for GOST round function $R \oplus k \oplus f(R,k)$, which are no smaller than 2^{-9}

mask	0x802001	0x803007	0x806001	0x806006	0x802007	0x500802
bias	2^{-7}	2^{-7}	2^{-7}	$2^{-7.2}$	$2^{-7.3}$	2^{-8}

mask	0x500803	0x803006	0x804006	0x3004008	0x3006008	0x8040060
bias	2^{-8}	$2^{-7.8}$	$2^{-7.8}$	$2^{-7.8}$	$2^{-7.6}$	$2^{-7.8}$

mask	0x30080200	0x30080500	0x200801	0x200802	0x200803	0x400802
bias	$2^{-7.8}$	$2^{-7.8}$	2^{-8}	$2^{-8.4}$	$2^{-8.4}$	$2^{-8.6}$

mask	0x400803	0x400807	0x802002	0x802006	0x804007	0x806002
bias	$2^{-8.6}$	$2^{-8.6}$	$2^{-8.4}$	$2^{-8.8}$	$2^{-8.7}$	$2^{-8.4}$

mask	0xc01801	0xc06801	0x4008070	0x8030060	0x30080300	0x40080500
bias	$2^{-8.2}$	$2^{-8.6}$	$2^{-8.7}$	$2^{-8.8}$	$2^{-8.8}$	$2^{-8.2}$

mask	0x60030080	0xc01002	0x3003008	0x8040020	0x8060020	0x30030080
bias	2^{-8}	2^{-9}	2^{-9}	2^{-9}	2^{-9}	2^{-9}

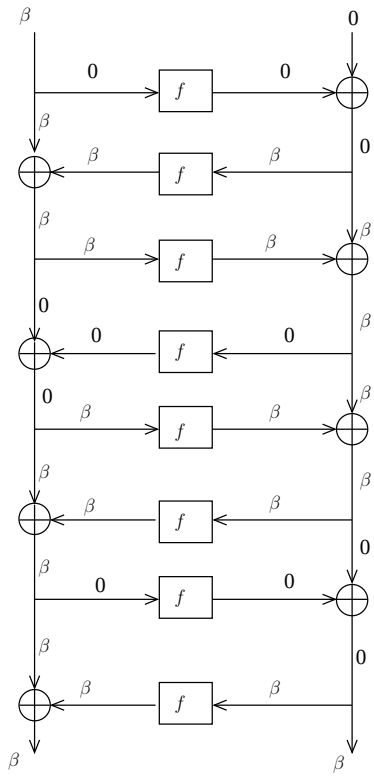

Fig. 1. The 8-round linear trails shown in untwisted structure, and the subkey mixing in f is omitted