J. Cryptol. (2013) 26: 172–189
DOI: 10.1007/s00145-012-9118-5

Journal of
CRYPTOLOGY

A Single-Key Attack on the Full GOST Block Cipher*

Takanori Isobe
Sony Corporation, 1-7-1 Konan, Minato-ku, Tokyo 108-0075, Japan
Takanori.Isobe@jp.sony.com

Communicated by Willi Meier

Received 4 April 2011
Online publication 2 February 2012

Abstract. The GOST block cipher is the Russian encryption standard published in 1989. In spite of considerable cryptanalytic efforts over the past 20 years, a key recovery attack on the full GOST block cipher without any key conditions (e.g., weak keys and related keys) has not been published yet. In this paper, we show the first single-key attack, which works for all key classes, on the full GOST block cipher. To begin, we develop a new attack framework called *Reflection-Meet-in-the-Middle Attack*. This approach combines techniques of the reflection attack and the meet-in-the-middle (MITM) attack. Then we apply it to the GOST block cipher employing bijective S-boxes. In order to construct the full-round attack, we use additional novel techniques which are the effective MITM techniques using equivalent keys on a small number of rounds. As a result, a key can be recovered with a time complexity of 2^{225} encryptions and 2^{32} known plaintexts. Moreover, we show that our attack is applicable to the full GOST block cipher using any S-boxes, including non-bijective S-boxes.

Key words. Block cipher, GOST, Single-key attack, Reflection attack, Meet-in-the-middle attack, Equivalent keys.

1. Introduction

The GOST block cipher [24] is known as the former Soviet encryption standard GOST 28147-89 which was standardized as the Russian encryption standard in 1989. It is widely used in Russia for commercial and governmental uses, and also adopted as a national standard of surrounding countries such as Belarus, Kazakhstan and Ukraine.

GOST[1] is based on a 32-round Feistel structure with 64-bit block and 256-bit key size. The round function consists of a key addition, eight 4×4-bit S-boxes and a rotation. The GOST standard [24] does not specify a set of S-boxes, and each industry uses a different set of S-boxes given by the government. For example, one set of the S-boxes used in the Central Bank of the Russian Federation is known as in [32]. Although the

[1] For the remainder of this paper, we refer to the GOST block cipher as GOST.

* Solicited from FSE, 2011.

choice of the S-boxes can also be seen as a part of the secret key, Saarinen has pointed out that a chosen key attack reveals the set of S-boxes with a time complexity of 2^{32} encryptions [29].

In addition, GOST is well-suited for compact hardware implementations due to its simple structure. Poschmann et al. showed an extremely compact implementation requiring only 651 GE [26]. Therefore, GOST is considered as one of ultra lightweight block ciphers such as PRESENT [6], KATAN family [8], LED [16] and Piccolo [35], which are suitable for the constrained environments including RFID tags and sensor nodes.

Over the past 20 years, a number of attacks on GOST have been published. Kelsey et al. first analyzed related-key characteristics of GOST [20]. After that, a differential attack on 13-round GOST was proposed by Seki and Kaneko [33]. In the related-key setting, the attack is improved up to 21 rounds [33]. Ko et al. showed a related-key differential attack on the full GOST [21]. These results work only when GOST employs the S-boxes of the Central Bank of the Russian Federation [32]. Fleischmann et al. presented a related-key boomerang attack on the full GOST which works for any S-boxes [15]. However, Rudskoy pointed out flaws of this attack and modified it [28]. As another type of attack, Biham et al. showed slide attacks on reduced GOST [2]. Their attack utilizes self similarities among round functions of the encryption process, and does not depend on the values of S-boxes. In other words, even if an attacker does not know the S-boxes, 24-round GOST can be attacked by this approach. If the values are known, this attack can be extended up to 30 rounds. In addition, for a class of 2^{128} weak keys, the full GOST can be attacked by this approach. After that, Kara proposed a reflection attack on 30-round GOST [18]. This attack also uses self similarities among round functions, and works for any bijective S-boxes. The difference from the slide attack proposed by Biham et al. [2] is to use self similarities between the round function and its inverse function.[2] The reflection attack utilizes these similarities in order to construct fixed points of some round functions. Moreover, for a class of 2^{224} weak keys, the full GOST can be attacked by using the reflection technique.

In spite of considerable cryptanalytic efforts, a key recovery attack on the full GOST without any key assumptions (e.g., weak keys and related keys) has not been published so far. Furthermore, a weak-key attack and a related-key attack are arguable in the practical sense, because of their strong assumptions. A weak-key attack is generally applicable to a small fraction of the keys, e.g., in the attack of [18], the rate of weak keys in all keys is $2^{-32} (= 2^{224}/2^{256})$. Hence, almost all keys, $(2^{256} - 2^{224}) \approx 2^{256}$ keys, cannot be attacked by [18]. Besides, the attacker cannot even know whether a target key is included in a weak-key class or not. Only if the key is in the weak-key class, the complexity for finding the key is less than that of the exhaustive key search. A related-key attack requires an assumption that the attacker can access the encryption/decryption under multiple unknown keys such that the relation between them is known to the attacker. Though this type of attack is meaningful during the design and certification of ciphers, it does not lead to a realistic threat in practical security protocols which use the block

[2] Another difference between [2] and [18] is rounds to be attacked. Biham et al. [2] proposes an attack on the first 30 rounds while [18] proposes an attack on the last 30 rounds.

cipher in a standard way as stated in [14].[3] Therefore, the security under the single-key setting is the most important issue from the aspect of the practical security. In particular, an ultra lightweight block cipher does not need a security against related-key attacks in many cases. For example, in low-end devices such as a passive RFID tag, the key may not be changed in its life cycle as mentioned in [6,8]. Indeed, KTANTAN supports only a fixed key [8] and the compact implementation of GOST proposed by Poschmann et al. also uses a hard-wired fixed key [26].

Recently, the Meet-in-the-Middle (MITM) attack on KTANTAN [8] was presented by Bogdanov and Rechberger [7]. Their attack mainly exploits the weakness of the key schedule function of KTANTAN, which is that large parts of the cipher depend on a part of key bits. The MITM attack seems to be effective for block ciphers whose key schedules are simple, e.g., a bit or a word permutation, in the sense that the key dependency is not strong. In fact, the key schedule function of KTANTAN consists of only a bit permutation. Since GOST also has a simple key schedule function, which is a word permutation, the MITM attack seems applicable to it. However, it does not work well on the full GOST, because the key dependency of the full GOST is stronger than that of KTANTAN due to the iterative use of key words in many round functions.

Our Contributions In this paper, we first introduce a new attack framework called *Reflection-Meet-in-the-Middle* (*R-MITM*) *Attack*; it is a combination of the reflection attack and the MITM attack. The core idea of this combination is to make use of fixed points of the reflection attack to enhance the MITM attack. If some round functions have fixed points, we can probabilistically remove these rounds from the whole cipher. Since this skip using fixed points allows us to disregard the key bits involved in the removed rounds, the key dependency is consequently weakened. Thus, our attack is applicable to more rounds compared with the original MITM attack if fixed points can be constructed with high probability. Then we apply it to the GOST block cipher employing bijective S-boxes. Furthermore, to construct the full-round attack, we use additional novel techniques which are the effective MITM techniques using equivalent keys on a small number of rounds. As a result, we succeed in constructing the first key recovery attack on the full GOST block cipher in the single-key setting. It can recover the full key with a time complexity of 2^{225} encryptions, 2^{32} known plaintext/ciphertext pairs and memory of 2^{64} blocks. Moreover, by analyzing the properties of GOST exploited for the attack deeply, we show that this attack can be applied to the full GOST block cipher using any S-boxes, including non-bijective S-boxes, with the same data and time complexities, while the memory depends on the used S-boxes. These results are summarized in Table 1.

Outline of the Paper This paper is organized as follows. Brief descriptions of GOST, the MITM attack and the reflection attack are given in Sect. 2. The R-MITM attack is introduced in Sect. 3. In Sect. 4, we present the R-MITM attack on the full GOST employing bijective S-boxes. In Sect. 5, we discuss an attack without constraints on S-boxes. Finally, we present conclusions in Sect. 6.

[3] In nonstandard uses, there are cases where a related-key weakness leads to realistic threats such as the Microsoft's Xbox attack [34]. In this case, the block cipher TEA [36] is used as a compression function in a hash function.

Table 1. Key recovery attacks on GOST.

Key setting	Type of attack	Round	Time	Data	Paper
Single key	Differential*1	13	Not given	2^{51} CP	[33]
	Slide*3	24	2^{63}	2^{63} ACP	[2]
	Slide*3	30	$2^{253.7}$	2^{63} ACP	[2]
	Reflection*2	30	2^{224}	2^{32} KP	[18]
	R-MITM *3	32	2^{225}	2^{32} KP	This paper
Single key (Weak key)	Slide (2^{128} weak keys)*3	32	2^{63}	2^{63} ACP	[2]
	Reflection (2^{224} weak keys)*2	32	2^{192}	2^{32} CP	[18]
Related key	Differential*1	21	Not given	2^{56} CP	[33]
	Differential*1	32	2^{244}	2^{35} CP	[21]
	Boomerang *2	32	2^{224}	2^{10} CP	[28]

KP: known plaintexts; CP: chosen plaintexts; ACP: adaptive chosen plaintexts.

*1 Works for one specific set of S-boxes in [32].

*2 Works for any bijective S-boxes.

*3 Works for any S-boxes.

Table 2. Key schedule of GOST.

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

2. Preliminaries

In this section, we give brief descriptions of GOST, the MITM attack and the reflection attack.

2.1. *Description of GOST*

GOST is a block cipher based on a 32-round Feistel structure with 64-bit block and 256-bit key size.

The 256-bit master key K is divided into eight 32-bit words, i.e., $K = (k_1, k_2, \ldots, k_8)$, $k_i \in \{0, 1\}^{32}$. These words are used as round keys rk_i $(1 \le i \le 32)$ as

$$rk_i = \begin{cases} k_{i-8\times\lfloor\frac{i-1}{8}\rfloor} & (1 \le i \le 24), \\ k_{9-(i-8\times\lfloor\frac{i-1}{8}\rfloor)} & (25 \le i \le 32). \end{cases}$$

See Table 2.

Let a 64-bit data state be $L_i \| R_i$, $\{L_i, R_i\} \in \{0, 1\}^{32}$, where $\|$ is a concatenation, $L_0 \| R_0$ denotes a plaintext P, and $L_{32} \| R_{32}$ denotes a ciphertext C. In each round, the

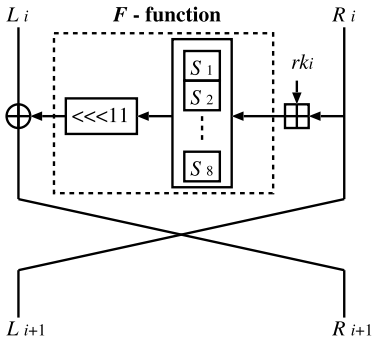

Fig. 1. One round of the GOST block cipher.

state is updated as

$$L_{i+1} = R_i,$$
$$R_{i+1} = L_i \oplus \big(F(rk_i \boxplus R_i)\big),$$

where $\boxplus$ is an addition modulo 2^{32} and $\oplus$ is a bitwise exclusive OR. The F-function consists of eight 4×4-bit S-boxes S_j $(1 \leq j \leq 8)$ and an 11-bit left rotation (see Fig. 1). Note that the GOST standard [24] does not specify a set of S-boxes, and each industry uses a different set of S-boxes.

2.2. *Meet-in-the-Middle Attack*

The Meet-in-the-Middle (MITM) attack was first proposed by Diffie and Helman [12]; it evaluates the security of the multiple encryptions with distinct keys $key1$ and $key2$ such that $C = E_{key2}(E_{key1}(P))$. The central idea is to compute $E_{key1}(P)$ and $E^{-1}_{key2}(C)$ independently by guessing $key1$ and $key2$. If the guessed key value is correct, the equation $E_{key1}(P) = E^{-1}_{key2}(C)$ holds. Due to the parallel guesses of $key1$ and $key2$, the attack is more effective than an exhaustive search in terms of time complexity. So far, the MITM attack has been applied to several block ciphers [9–11,13,14,17]. Furthermore, over the past few years, this attack has been improved in a line of preimage attacks on hash functions, and several novel techniques have been introduced, e.g., the splice and cut [1] and the initial structure [30].

The MITM attack consists of two stages: a MITM stage and a key testing stage. First, the MITM stage filters out some wrong key candidates and reduces the key space. Then the key testing stage finds a correct key from the surviving key candidates in a brute force manner.

Let $E_K : \{0,1\}^b \to \{0,1\}^b$ be a block cipher with an l-bit key K and a b-bit block. Assume that E_K is a composition of round functions as follows:

$$E_K(x) = F_{k_r} \circ F_{k_{r-1}} \circ \cdots \circ F_{k_1}(x), \quad x \in \{0,1\}^b,$$

where r is the number of rounds, $k_1, \ldots, k_r$ are round keys and F_{k_i} is the i-th round function, $F_{k_i} : \{0,1\}^b \to \{0,1\}^b$. The composition of $j - i + 1$ functions starting from

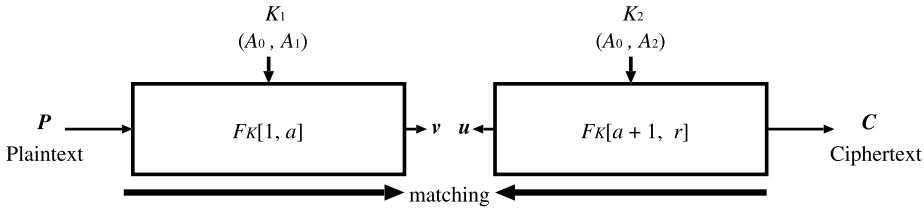

Fig. 2. Meet-in-the-middle stage.

i is denoted by $F_K[i, j]$ defined as

$$F_K[i, j](x) = F_{k_j} \circ \cdots \circ F_{k_i}(x), \quad 1 \le i < j \le r.$$

In the following, we give details of each stage of the MITM attack.

MITM Stage $E_k(X)$ is divided into two functions as $E_K(X) = F_K[a+1, r] \circ F_K[1, a]$, $1 < a < r - 1$.[4] Let K_1 and K_2 be sets of key bits used in $F_K[1, a]$ and $F_K[a + 1, r]$, respectively. $A_0 = K_1 \cap K_2$ is the common set of key bits used in both $F_K[1, a]$ and $F_K[a + 1, r]$. $A_1 = K_1 \setminus K_1 \cap K_2$ and $A_2 = K_2 \setminus K_1 \cap K_2$ are the sets of key bits used in only $F_K[1, a]$ and only $F_K[a + 1, r]$, respectively. In this stage, we use only one plaintext/ciphertext pair (P, C).[5]

The procedure of the MITM stage is as follows. Figure 2 shows the overview of the MITM stage.

1. Guess the value of A_0.
2. Compute $v = F_K[1, a](P)$ for all values of A_1 and make a table of (v, A_1) pairs. In this step, $2^{|A_1|}$ pairs are generated, where $|A_i|$ is the bit length of A_i and $2^{|A_i|}$ is the number of elements of A_i.
3. Compute $u = F_K^{-1}[a + 1, r](C)$ for all values of A_2. In this step, $2^{|A_2|}$ pairs are generated.
4. Add key candidates for which the equation $v = u$ is satisfied to the list of surviving keys. The number of surviving keys is $2^{|A_1|+|A_2|}/2^b$.
5. Repeat steps 2–4 for each different value of A_0 ($2^{|A_0|}$ times).

In this stage, 2^{l-b} key candidates survive, because $2^{|A_1|+|A_2|}/2^b \times 2^{|A_0|} = 2^l/2^b$.

Key Testing Stage We test surviving keys in a brute force manner by using additional plaintext/ciphertext pairs.

We evaluate the cost of this attack. The whole attack complexity C_{comp} is estimated as

$$C_{\text{comp}} = \underbrace{2^{|A_0|}\left(2^{|A_1|} + 2^{|A_2|}\right)}_{\text{MITM stage}} + \underbrace{\left(2^{l-b} + 2^{l-2b} + \cdots\right)}_{\text{Key testing stage}}.$$

[4] As in the attack of KTANTAN [7], by using the partial matching technique, E_K is divided into $F_K[1, a]$ and $F_K[a + t, r]$, $t > 1$. However, we consider only the case of $t = 1$, because we do not use partial matching.

[5] Though more such pairs can be used to reduce the key space, this paper only treats a single pair. This constraint is necessary for utilizing the equivalent-key technique in our attack.

The number of required plaintext/ciphertext pairs is $\lceil \frac{l}{b} \rceil$. The required memory is $\min(2^{|A_1|}, 2^{|A_2|})$ blocks,[6] which is the cost of the table used in the MITM stage. When $\min(|A_1|, |A_2|) > 1$, the attack is more effective than an exhaustive search. Therefore, the point of the MITM attack is to find independent sets of master key bits such as A_1 and A_2.

2.3. *Reflection Attack*

The reflection attack was first introduced by Kara and Manap [19]; it was applied to Blowfish [31]. After that, the attack was generalized by Kara [18]. In this section, we introduce the basic principle of the reflection attack used in our attack. See [18,19] for more details about the reflection attack.

The reflection attack is a kind of a self-similarity attack such as the slide attack [4,5]. Though the reflection attack utilizes similarities between the round function and its inverse function, the slide attack exploits similarities among the round functions of only the encryption process. The reflection attack utilizes these similarities in order to construct fixed points of some round functions.

Let $U_K(i, j)$ be the set of fixed points of the function $F_K[i, j]$ defined as follows:

$$U_K(i, j) = \left\{x \in \{0, 1\}^n \mid F_K[i, j](x) = x\right\}.$$

The basic principle of the reflection attack is given by the following lemma.

Lemma 1 [18]. *Let i and j be integers such that $0 \leq j - i < i + j < r$. Assume that $F_{k_{i-t}} = F^{-1}_{k_{j+t}}$ for all t: $1 \leq t < i$. If $F_K[i - t, i - 1](x) \in U_K(i, j)$, then $x \in U_K(i - t, j + t)$ for all t: $1 < t < i$. In addition, if $x \in U_K(i - t, j + t)$ for certain t: $1 < t < i$, then $F_K[i - t, i - 1](x) \in U_K(i, j)$.*

For the explanation of Lemma 1, we consider a 4-round Feistel structure with 64-bit block, E_K: $\{0, 1\}^{64} \rightarrow \{0, 1\}^{64}$. Assume that each round uses the same round key rk, then E_K is expressed as

$$\begin{aligned} E_K &= F_K[3, 4] \circ F_K[1, 2] \\ &= F_K^{-1}[1, 2] \circ S \circ F_K[1, 2], \end{aligned}$$

where S is the swap of the Feistel structure. Figure 3 shows these equivalent descriptions of the 4-round Feistel structure. S has 2^{32} fixed points, because the probability of that the right halves equal to the left halves is 2^{-32}. Thus, E_K also has 2^{32} fixed points, i.e., $|U_K(1, 4)| = 2^{32}$ due to the property of $F_K[3, 4] = F_K^{-1}[1, 2]$.

Lemma 1 shows that if the round functions hold the conditions, a local fixed point is expanded to previous and next rounds as shown in Fig. 4. Roughly speaking, such a cipher may have fix points of the long round functions if local fixed points are found. These fixed points enable us to probabilistically skip the round functions in a whole cipher.

[6] When $|A_1| > |A_2|$, it is possible to swap roles of A_1 and A_2.

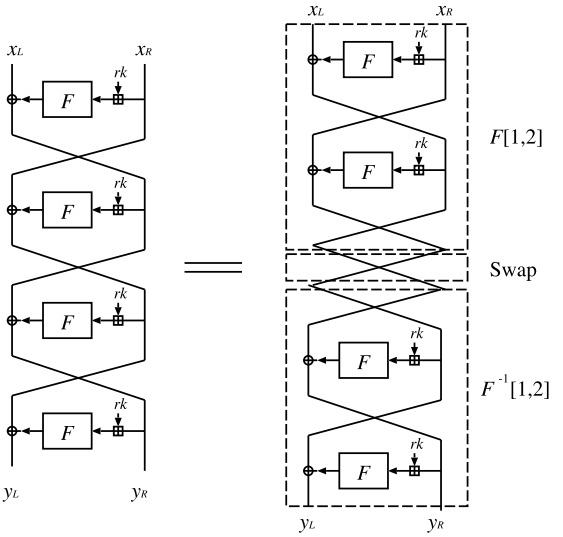

Fig. 3. Equivalent descriptions of the 4-round Feistel structure with the same round key rk in all round.

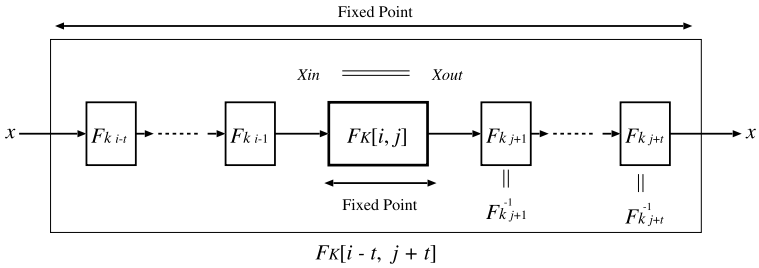

Fig. 4. Basic principle of the reflection attack.

We give an example to explain this skip in detail. Let i and j be integers such that $0 < j - i < i + j < r$. Assume that $F_{k_{i-t}} = F_{k_{j+t}}^{-1}$ for all t: $1 < t < i$, and $E_K(x)$ is expressed as follows:

$$\begin{aligned} E_K(x) &= F_K[j+i, r] \circ F_K[j+1, j+i-1] \circ F_K[i, j] \circ F_K[1, i-1](x) \\ &= F_K[j+i, r] \circ F_K^{-1}[1, i-1] \circ F_K[i, j] \circ F_K[1, i-1](x). \end{aligned}$$

Additionally, assume $F_K[1, i-1](x) \in U_K(i, j)$, then $F_K[1, j+i-1](x) = x$ (Lemma 1). Thus $E_K(x)$ is expressed as

$$E_K(x) = F[j+i, r](x).$$

In this case, the round functions $F_K[1, j+i-1]$ can be skipped from E_K. The probability P_{ref} of that above skip occurs for arbitrary x is $|U_K(i, j)|/2^b$. If $P_{\text{ref}} > 2^{-b}$ (i.e.,

$|U_K[i, j]| > 1$), this skip occurs at $F_K[1, j + i - 1]$ with higher probability than a random function.

3. Reflection-Meet-in-the-Middle Attack

We propose a new attack framework called *reflection-meet-in-the-middle (R-MITM) attack*, which is a combination of the reflection and the MITM attacks. As mentioned in Sect. 2.2, the point of the MITM attack is to construct independent sets of master key bits. In general, if the master key bits are used iteratively in each round and the use of key bits is not biased among rounds,[7] it seems to be difficult to find the independent sets of master key bits, because such a cipher has a strong key dependency even for a small number of rounds.

To overcome this problem, we utilize the technique of the reflection attack. In the reflection attack, some rounds satisfying certain conditions can be skipped from the whole cipher with probability P_{ref}. From now on, we call this skip a *reflection skip*. Since key bits used in skipped round functions can be omitted, it becomes easier to construct independent sets of master key bits. Thus, the R-MITM attack is applicable to more rounds compared with the original MITM attack when the reflection skip occurs with high probability. This is the concept of the R-MITM attack. In the following, we give a detailed explanation of the attack.

3.1. *Details of the R-MITM Attack*

Suppose that E_K is expressed as follows:

$$E_K(x) = F_K[a_3 + 1, r] \circ F_K[a_2 + 1, a_3] \circ F_K[a_1 + 1, a_2] \circ F_K[1, a_1](x),$$

where $2 < a_1 + 1 < a_2 < a_3 - 1 < r - 2$ and the reflection skip occurs at $F_K[a_2 + 1, a_3]$ with probability P_{ref}. Then E_K can be redescribed as follows and denoted by $E'_K(x)$:

$$E'_K(x) = F_K[a_3 + 1, r] \circ F_K[a_1 + 1, a_2] \circ F_K[1, a_1](x).$$

The R-MITM attack consists of three stages; a data collection stage, an R-MITM stage and a key testing stage. In the following, we explain each stage.

Data Collection Stage We collect plaintext/ciphertext pairs to obtain a pair in which the reflection skip occurs at $F_K[a_2 + 1, a_3]$. Since the probability of this event is P_{ref}, the number of required plaintext/ciphertext pairs is P_{ref}^{-1}.

After that, the R-MITM stage and the key testing stage are executed for all plaintext/ciphertext pairs obtained in the data collection stage.

R-MITM Stage We divide E_K into two functions: $F_K[1, a_1]$ and $F_K[a_1 + 1, r]$.[8] In this stage, we ignore $F_K[a_2 + 1, a_3]$ as follows:

$$F'_K[a_1 + 1, r] = F_K[a_3 + 1, r] \circ F_K[a_1 + 1, a_2],$$

[7] In KTANTAN [8], 6 bits of master key are not used in the first 111 rounds and another 6 bits of master key are not used in the last 131 rounds. The attack of [7] utilizes this bias of used key bits among rounds.

[8] Though there are many choices of divisions, we use it as an example.

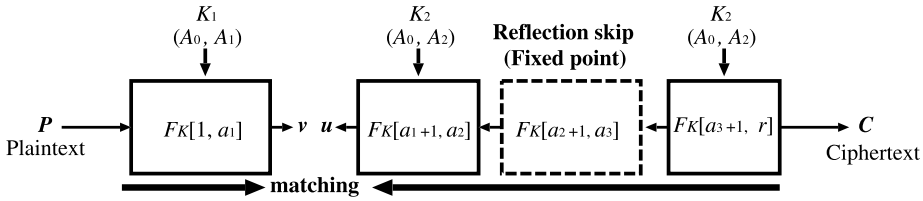

Fig. 5. Reflection-meet-in-the-middle stage.

assuming that the reflection skip occurs. Let K_1 and K_2 be sets of key bits used in $F_K[1, a_1]$ and $F'_K[a_1 + 1, r]$, respectively. $A_0 = K_1 \cap K_2$ is the set of key bits used in both $F_K[1, a_1]$ and $F'_K[a_1 + 1, r]$. $A_1 = K_1 \setminus K_1 \cap K_2$ and $A_2 = K_2 \setminus K_1 \cap K_2$ are the sets of key bits used in only $F_K[1, a_1]$ and only $F'_K[a_1 + 1, r]$, respectively. Figure 5 illustrates the R-MITM stage.

The procedure of the R-MITM stage is almost the same as the MITM stage of Sect. 2.2. The difference is that in the R-MITM stage, we assume that a reflection skip occurs, i.e., $F_K[a_2 + 1, a_3]$ is ignored. After this stage, 2^{l-b} key candidates survive.

Key Testing Stage We test surviving keys in a brute force manner by using several plaintext/ciphertext pairs.

3.2. *Evaluation of the R-MITM Attack*

We evaluate the cost of the R-MITM attack. The whole attack complexity C_{comp} is estimated as

$$C_{\text{comp}} = \underbrace{\left(\left(2^{|A_0|}\left(2^{|A_1|} + 2^{|A_2|}\right)\right)\right)}_{\text{R-MITM stage}} + \underbrace{\left(2^{l-b} + 2^{l-2b} + \cdots\right)}_{\text{Key testing stage}} \times P_{\text{ref}}^{-1}.$$

The number of required plaintext/ciphertext pairs is $\max(\lceil l/b \rceil, P_{\text{ref}}^{-1})$. The required memory is $\min(2^{|A_1|}, 2^{|A_2|})$ blocks, which is the cost of the table in the R-MITM stage. When $\min(2^{|A_1|}, 2^{|A_2|}, 2^b) > (P_{\text{ref}}^{-1})$, the attack is more effective than an exhaustive search.

Compared with the basic MITM attack in Sect. 2.2 for the R-MITM attack, the number of required plaintext/ciphertext pairs increases, because the R-MITM attack utilizes the probabilistic event, i.e., reflection skip. In addition, more independent key bits are needed for the successful attack. However, the R-MITM attack has a distinct advantage, which is the ability to skip some round functions by the reflection skip. Recall that it is the essential for the MITM attack to find independent sets of master key bits. Since the reflection skip enables us to disregard key bits involved in some rounds, it obviously becomes easier to construct such independent sets. Thus, this attack seems to be applicable to more rounds than the MITM attack when the reflection skip occurs with high probability.

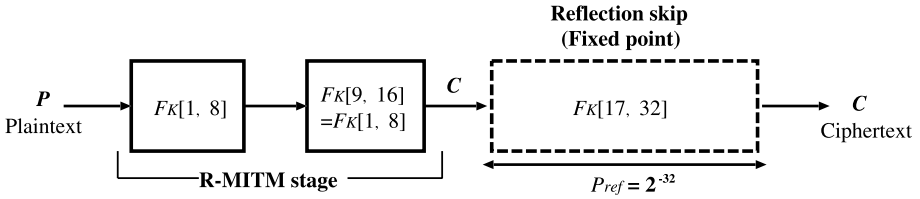

Fig. 6. Reflection skip of GOST.

4. R-MITM Attack on the Full GOST Block Cipher

In this section, we apply the R-MITM attack to the full GOST block cipher [24]. From Table 2, in the whole 32 rounds, the master key is iteratively used four times and all master key bits are used in the each 8-round units, i.e., 1–8, 9–16, 17–24 and 25–32 rounds. The basic MITM attack in Sect. 2.2 is not applicable to the full GOST, because independent sets of master key bits cannot be constructed in any divisions of 32 rounds. However, by using the R-MITM attack, we can construct independent sets and mount a key recovery attack on the full GOST. In this attack, we do not care about specific values of the S-boxes, and only assume that the S-boxes are bijective.

We first introduce the reflection property of GOST proposed by Kara [18] to construct the reflection skip. Next, we present an effective MITM techniques to enhance the R-MITM stage. These techniques make use of the equivalent keys of four round functions. Finally, we evaluate our attack.

4.1. *Reflection Property of GOST*

The reflection attack on GOST has been proposed by Kara [18].[9] The GOST block cipher $E_K : \{0,1\}^{64} \to \{0,1\}^{64}$ is expressed as

$$\begin{aligned} E_K &= S \circ F_K[25,32] \circ F_K[17,24] \circ F_K[9,16] \circ F_K[1,8] \\ &= F_K^{-1}[1,8] \circ S \circ F_K[1,8] \circ F_K[1,8] \circ F_K[1,8]. \end{aligned}$$

As mentioned Sect. 2.3, S has 2^{32} fixed points. From Lemma 1, $F_K^{-1}[1,8] \circ S \circ F_K[1,8]$ also has 2^{32} fixed points, i.e., $|U_K(17,32)| = 2^{32}$. Thus, with probability $P_{\text{ref}} = 2^{-32}$ $(= (2^{32}/2^{64}))$, $F_K[17,32]$ can be ignored. E_K is redescribed as follows and denoted by E'_K

$$E'_K = F_K[1,8] \circ F_K[1,8].$$

Figure 6 shows this reflection skip of GOST.

Therefore, in the data collection stage, we need to collect $P_{\text{ref}}^{-1} = 2^{32}$ plaintext/ciphertext pairs. In 2^{32} collected pairs, there is a pair in which the reflection skip occurs, i.e., last 16 rounds can be removed from E_K.

[9] The similar technique for constructing a fixed point is also used in the attacks on the GOST hash function [22,23].

4.2. *Effective MITM Technique Using Equivalent Keys on Four Rounds*

In the R-MITM stage, we mount the MITM approach on only $E'_K = F_K[1,8] \circ F_K[1,8]$ for all 2^{32} collected pairs.

As described in Sect. 3.2, we need to construct independent sets A_1 and A_2 which hold the condition, $\min(2^{|A_1|}, 2^{|A_2|}) > 2^{32}$. However, despite the reduction of rounds by the reflection skip, in the straightforward method, we cannot find such sets in any divisions of 16 rounds, due to the strict condition of independent sets.

We introduce effective MITM techniques which make use of equivalent keys of four round functions. The aim of these techniques is to ignore the first and the last four rounds and to mount the MITM approach in only the intermediate eight rounds. These techniques enable us to construct independent sets enough for the successful attack.

We treat E'_K as the following four round units:

$$E'_K = F_K[5,8] \circ F_K[1,4] \circ F_K[5,8] \circ F_K[1,4].$$

In the following, we first explain equivalent keys used in our attack. Then we present details of the R-MITM stage using the equivalent keys.

Equivalent Keys on Four Rounds Define a set of equivalent keys on $F_K[i,j]$ as follows:

$$Z\big(F_K[i,j], x, y\big) = \big\{ek \in \{0,1\}^{256} \mid F_{ek}[i,j](x) = y\big\},$$

where $(x, y) \in \{0,1\}^{64}$. Note that the class of keys defined above is the equivalent keys with respect to only one input/output pair. To put it more concretely, if equivalent keys $ek \in Z(F_K[i,j], x, y)$ are used, an input x is always transformed to y in $F_K[i,j]$. For other input/output pairs, these relations do not hold even if the same equivalent keys are used.

GOST has an interesting property regarding the equivalent keys on four rounds as described in the following observation.

Observation 1. *Given any x and y, $Z(F_K[1,4], x, y)$ and $Z(F_K^{-1}[5,8], x, y)$ can be easily obtained, and the number of equivalent keys for each pair (x, y) is 2^{64}.*

For $F_K[1,4]$, k_1, k_2, k_3 and k_4 are added in each round. Given the values of k_1 and k_2, the other values of k_3 and k_4 are determined from $F_K[1,2](x)$ and y as follows:

$$k_3 = F^{-1}(z_L + y_L) - z_R, \tag{1}$$

$$k_4 = F^{-1}(z_R + y_R) - y_R, \tag{2}$$

where F^{-1} is the inverse of F-function, y_L and y_R are left and right halves of y, and z_L and z_R are those of $F_K[1,2](x)$. Since the values of (k_1, k_2) are 64 bits, the number of $Z(F_K[1,4], x, y)$ is 2^{64}. Figure 7 shows this procedure. A similar property holds for $F_K^{-1}[5,8]$.

From Observation 1, we can easily obtain 2^{64} equivalent keys of the first and the last four rounds for any input/output pairs. Besides, $F_K[1,4]$ and $F_K^{-1}[5,8]$ use different

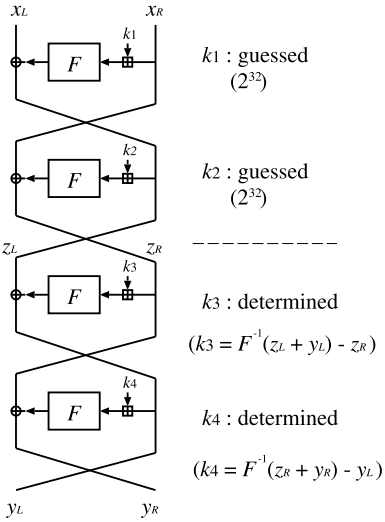

Fig. 7. Equivalent keys of four rounds.

key bits each other, $K_a = (k_1 \| k_2 \| k_3 \| k_4)$ and $K_b = (k_5 \| k_6 \| k_7 \| k_8)$, respectively. Thus, $Z(F_K[1,4], x, y)$ and $Z(F_K^{-1}[5,8], x, y)$ can be expressed by sets of only K_a and K_b as follows:

$$Z_{Ka}\big(F_K[1,4], x, y\big) = \big\{ek_a \in \{0,1\}^{128} \mid F_{ek_a}[1,4](x) = y\big\},$$
$$Z_{Kb}\big(F_K^{-1}[5,8], x, y\big) = \big\{ek_b \in \{0,1\}^{128} \mid F_{ek_b}^{-1}[5,8](x) = y\big\}.$$

Since K_a and K_b are independent sets of master keys, $Z_{Ka}(F_K[1,4], x, y)$ and $Z_{Kb}(F_K^{-1}[5,8], x, y)$ are also independent sets.

R-MITM Stage Using Equivalent Keys Let S and T be $F_K[1,4](P)$ and $F^{-1}[5,8](C)$, which are the input and output values of intermediate eight rounds, i.e., $F_K[5,12] = F_K[1,4] \circ F_K[5,8]$. From Observation 1, given the values of P, C, S and T, we can easily obtain two sets of 2^{64} equivalent keys, $Z_{K_a}(F_K[1,4], P, S)$ and $Z_{K_b}(F_K^{-1}[5,8], C, T)$.

When $Z_{K_a}(F_K[1,4], P, S)$ and $Z_{K_b}(F_K^{-1}[5,8], C, T)$ are used, S and T are not changed. Thus by using these equivalent keys, the first and the last four round can be ignored, and we can mount the MITM attack between $F_K[5,8](S)$ and $F_K^{-1}[1,4](T)$. The number of elements in each independent set is 2^{64}, which is enough for a successful attack.

The procedure of the R-MITM stage is as follows and illustrated in Fig. 8.

1. Guess the values S and T.
2. Compute $v = F_K[5,8](S)$ with 2^{64} K_b in $Z_{K_b}(F_K^{-1}[5,8], C, T)$ and make a table of (v, K_b) pairs.

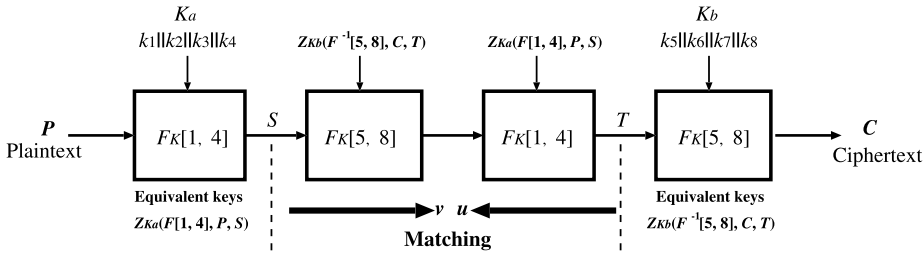

Fig. 8. R-MITM stage using equivalent keys.

3. Compute $u = F_K^{-1}[1,4](T)$ with 2^{64} K_a in $Z_{K_a}(F_K[1,4], P, S)$.
4. Add key candidates for which the equation $v = u$ is satisfied to the list of surviving keys. The number of surviving keys is $2^{64+64}/2^{64} = 2^{64}$.
5. Repeat 2–4 with the different values of S and T (2^{128} times).

After this procedure, 2^{192} ($= 2^{64} \times 2^{128}$) key candidates survive. These key candidates are evaluated in the key testing stage.

The R-MITM stage utilizes equivalent-key sets of $Z_{K_a}(F_K[1,4], P, S)$ and $Z_{K_b}(F_K^{-1}[5,8], C, T)$, $0 \le S,\ T < 2^{64}$, where each set includes 2^{64} elements. For $Z_{K_a}(F_K[1,4], P, S)$, $0 \le S < 2^{64}$, from (1) and (2), all elements of every set are surely different under the assumption that the S-boxes are bijective. Thus, $Z_{K_a}(F_K[1,4], P, S)$, $0 \le S < 2^{64}$ covers all 2^{128} ($= 2^{64} \times 2^{64}$) values of K_a. A similar property holds for K_b. Therefore, all possible values for the master key are tested and the set of surviving key candidates surely contain the correct key if the reflection skip occurs.

4.3. *Evaluation*

The whole attack complexity C_{comp} is estimated as

$$C_{\text{comp}} = \underbrace{\left(\left(2^{128}\left(2^{64} + 2^{64}\right)\right)\right)}_{\text{R-MITM stage}} + \underbrace{\left(2^{256-64} + 2^{256-128} + \cdots\right)}_{\text{Key testing stage}} \times 2^{32}$$
$$= 2^{225}.$$

The number of required known plaintext/ciphertext pairs is $\max(\lceil l/b \rceil, P_{\text{ref}}^{-1}) = \max(\lceil 256/64 \rceil, 2^{32}) = 2^{32}$. The required memory is $\min(2^{64}, 2^{64}) = 2^{64}$ blocks, which is the cost of the table used in the R-MITM stage. Therefore, this attack can recover a key with a time complexity of 2^{225} encryptions, 2^{32} known plaintext/ciphertext pairs and 2^{64} memory. It is more effective than an exhaustive attack in terms of time complexity.

5. Discussion: Attack Without Constraints on S-boxes

In this section, we consider an attack without constraints on the used S-boxes unlike the attack in Sect. 4 which works only when GOST employs bijective S-boxes. The

property of bijective S-boxes is utilized for finding equivalent keys of four rounds (Observation 1). Given a pair (x, y), the values of k_3 and k_4 are determined from the values of k_1 and k_2 by using (1) and (2). Under the assumption that the S-boxes are bijective, since the F-functions are also bijective, each k_3 and k_4 has only one solution with respect to arbitrary values of k_1 and k_2. Thus, the number of equivalent keys for each pair (x, y) is 2^{64}, because the joint length of k_1 and k_2 are 64 bits. Thus, the time complexity of the R-MITM stage is easily estimated as $2^{128}(2^{64} + 2^{64})$ encryptions.

For an arbitrary S-box, including non-bijective S-boxes, it is difficult to estimate the number of equivalent keys for each pair (x, y). When the F-functions is not bijective, it is not guaranteed that each k_3 and k_4 has only one solution for (1) and (2). In other words, there are cases where k_3 and k_4 have more than one solution or no solution for particular values of k_1 and k_2. Thus, the number of equivalent keys for each pair (x, y) strongly depends on the used S-boxes.

However, we are still able to evaluate the whole complexity of the attack. Define the number of equivalent keys in a set $Z(F_K[i, j], x, y)$ as $\#\, Z(F_K[i, j], x, y)$. In the R-MITM stage, for all 2^{128} values of S and T, v and u are computed by using sets of equivalent keys $Z_{K_b}(F_K^{-1}[5, 8], C, T)$ and $Z_{K_a}(F_K[1, 4], P, S)$, respectively. The complexity in the R-MITM stage is estimated as

$$\sum_{S=0}^{2^{64}-1}\left(\sum_{T=0}^{2^{64}-1}\left(\#Z_{K_b}\left(F_K^{-1}[5, 8], C, T\right) + \#Z_{K_a}\left(F_K[1, 4], P, S\right)\right)\right)$$
$$= 2^{64} \cdot \left(\sum_{T=0}^{2^{64}-1} \#Z_{K_b}\left(F_K^{-1}[5, 8], C, T\right)\right) + 2^{64} \cdot \left(\sum_{S=0}^{2^{64}-1} \#Z_{K_a}\left(F_K[1, 4], P, S\right)\right).$$

According to the definition of the equivalent keys in Sect. 4.2,

$$\sum_{T=0}^{2^{64}-1} \#Z_{K_b}\left(F_K^{-1}[5, 8], C, T\right) = \sum_{S=0}^{2^{64}-1} \#Z_{K_a}\left(F_K[1, 4], P, S\right) = 2^{128}.$$

Thus, the time complexity of the R-MITM stage is $2^{193}(= 2^{64} \cdot 2^{128} + 2^{64} \cdot 2^{128})$ encryptions. In addition, the condition for surviving keys is not changed, i.e., $v = u$.

Since the reflection property does not depend on the S-boxes, the number of required known plaintext/ciphertext pairs is also 2^{32}. The whole complexity is estimated as

$$C_{\text{comp}} = \Big(\underbrace{2^{193}}_{\text{R-MITM stage}} + \underbrace{\left(2^{256-64} + 2^{256-128} + \cdots\right)}_{\text{Key testing stage}} \Big) \times 2^{32}$$
$$= 2^{225}.$$

Therefore, even if arbitrary S-boxes, including non-bijective S-boxes, the complexity and required data are the same as those of the attack on GOST employing bijective S-boxes in Sect. 4.

However, the required memory size is different, and it depends on the used set of S-boxes. From the procedures of the R-MITM attack, the required memory, which is used for the matching, is estimated as the maximum value of the set

$\{\min(\#Z_{K_b}(F_K^{-1}[5,8], C, T), \#Z_{K_a}(F_K[1,4], P, S)) \mid 0 \leq S,\ T < 2^{64}\}$. The best case is when

$$\forall S, T \in \{0,1\}^{64}, \quad \#Z_{K_b}\left(F_K^{-1}[5,8], C, T\right) = \#Z_{K_a}\left(F_K[1,4], P, S\right) = 2^{64},$$

which holds if and only if all S-boxes are bijective. Then the required memory is 2^{64} blocks. The worst case is when

$$\exists S, T, \in \{0,1\}^{64}, \quad \#Z_{K_b}\left(F_K^{-1}[5,8], C, T\right) = \#Z_{K_a}\left(F_K[1,4], P, S\right) = 2^{128}.$$

Then the required memory is 2^{128} blocks.

Therefore, although the required memory depends on the values of the used S-boxes, our attack is applicable to the full GOST using any S-boxes with the same data and time complexities.

6. Conclusion

This paper has presented the first single-key attack on the full GOST block cipher without relying on weak-key classes. To build the attack, we introduced a new attack framework called *Reflection-Meet-in-the-Middle Attack*, which is the combination of the reflection and the MITM attacks. We then applied it to the full GOST block cipher employing bijective S-boxes. Furthermore, in order to construct the full-round attack, we utilize further novel techniques which make use of equivalent keys of four round functions. These techniques enable us to mount the effective MITM approach. As a result, we succeeded in constructing the first key recovery attack on the full GOST without any key conditions, which works for any bijective S-boxes. Moreover, by analyzing procedures of the attack, we showed that this attack can be extended to GOST employing any S-boxes, including non-bijective S-boxes. Our result shows that GOST does not have the 256-bit security for all key classes, even if a fixed key is used such as in [26].

The idea of the R-MITM attack seems applicable to other block ciphers in which the fixed point can be constructed with high probability and its key schedule is simple in the sense that the key dependency is not strong. Furthermore, the basic principle of the attack does not require the reflection property and fixed points. Other non-random properties of round functions may also be able to be utilized as the skip techniques, e.g., the strong correlations among round functions.

Acknowledgements

We would like to thank to Taizo Shirai, Kyoji Shibutani, Özgül Küçük, and anonymous referees for their insightful comments and suggestions.

References

[1] K. Aoki, Y. Sasaki, Preimage attacks on one-block MD4, 63-step MD5 and more, in *SAC*, ed. by R.M. Avanzi, L. Keliher, F. Sica. Lecture Notes in Computer Science, vol. 5381 (Springer, Berlin, 2008), pp. 103–119

[2] E. Biham, O. Dunkelman, N. Keller, Improved slide attacks, in *[3]* (2007), pp. 153–166
[3] A. Biryukov (ed.), *Fast Software Encryption, 14th International Workshop, FSE 2007*, Luxembourg, Luxembourg, 26–28 March, 2007, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4593 (Springer, Berlin, 2007)
[4] A. Biryukov, D. Wagner, Slide attacks, in *FSE*, ed. by L.R. Knudsen. Lecture Notes in Computer Science, vol. 1636 (Springer, Berlin, 1999), pp. 245–259
[5] A. Biryukov, D. Wagner, Advanced slide attacks, in *EUROCRYPT*, ed. by B. Preneel. Lecture Notes in Computer Science, vol. 1807 (Springer, Berlin, 2000), pp. 589–606
[6] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: an ultra-lightweight block cipher, in *CHES*, ed. by P. Paillier, I. Verbauwhede. Lecture Notes in Computer Science, vol. 4727 (Springer, Berlin, 2007), pp. 450–466
[7] A. Bogdanov, C. Rechberger, A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN, in *Selected Areas in Cryptography*, ed. by A. Biryukov, G. Gong, D.R. Stinson. Lecture Notes in Computer Science, vol. 6544 (Springer, Berlin, 2010), pp. 229–240
[8] C.D. Cannière, O. Dunkelman, M. Knezevic, KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers, in *CHES*, ed. by C. Clavier, K. Gaj. Lecture Notes in Computer Science, vol. 5747 (Springer, Berlin, 2009), pp. 272–288
[9] D. Chaum, J. Evertse, Crytanalysis of DES with a reduced number of rounds: sequences of linear factors in block ciphers, in *CRYPTO*, ed. by H.C. Williams. Lecture Notes in Computer Science, vol. 218 (Springer, Berlin, 1985), pp. 192–211
[10] H. Demirci, A.A. Selçuk, A meet-in-the-middle attack on 8-round AES, in *[25]* (2008), pp. 116–126
[11] H. Demirci, I. Taskin, M. Çoban, A. Baysal, Improved meet-in-the-middle attacks on AES, in *INDOCRYPT*, ed. by B.K. Roy, N. Sendrier. Lecture Notes in Computer Science, vol. 5922 (Springer, Berlin, 2009), pp. 144–156
[12] W. Diffie, M.E. Hellman, Exhaustive cryptanalysis of the NBS data encryption standard. *Computer* **10**, 74–84 (1977)
[13] O. Dunkelman, G. Sekar, B. Preneel, Improved meet-in-the-middle attacks on reduced-round DES, in *INDOCRYPT*, ed. by K. Srinathan, C.P. Rangan, M. Yung. Lecture Notes in Computer Science, vol. 4859 (Springer, Berlin, 2007), pp. 86–100
[14] O. Dunkelman, N. Keller, A. Shamir, Improved single-key attacks on 8-round AES-192 and AES-256, in *ASIACRYPT*, ed. by M. Abe. Lecture Notes in Computer Science, vol. 6477 (Springer, Berlin, 2010), pp. 158–176
[15] E. Fleischmann, M. Gorski, J. Hüehne, S. Lucks, Key recovery attack on full GOST. Block cipher with negligible time and memory, in *Western European Workshop on Research in Cryptology (WEWoRC)*. LNCS, vol. 6429 (Springer, Berlin, 2009) (to appear)
[16] J. Guo, T. Peyrin, A. Poschmann, M.J.B. Robshaw, The LED block cipher, in *[27]* (2011), pp. 326–341
[17] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, B. Preneel, A practical attack on KeeLoq, in *EUROCRYPT*, ed. by N.P. Smart. Lecture Notes in Computer Science, vol. 4965 (Springer, Berlin, 2008), pp. 1–18
[18] O. Kara, Reflection cryptanalysis of some ciphers, in *INDOCRYPT*, ed. by D.R. Chowdhury, V. Rijmen, A. Das. Lecture Notes in Computer Science, vol. 5365 (Springer, Berlin, 2008), pp. 294–307
[19] O. Kara, C. Manap, A new class of weak keys for blowfish, in *[3]* (2007), pp. 167–180
[20] J. Kelsey, B. Schneier, D. Wagner, Key-schedule cryptoanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES, in *CRYPTO*, ed. by N. Koblitz. Lecture Notes in Computer Science, vol. 1109 (Springer, Berlin, 1996), pp. 237–251
[21] Y. Ko, S. Hong, W. Lee, S. Lee, J.-S. Kang, Related key differential attacks on 27 rounds of XTEA and full-round GOST, in *FSE*, ed. by B.K. Roy, W. Meier. Lecture Notes in Computer Science, vol. 3017 (Springer, Berlin, 2004), pp. 299–316
[22] F. Mendel, N. Pramstaller, C. Rechberger, A (second) preimage attack on the GOST hash function, in *[25]* (2008), pp. 224–234
[23] F. Mendel, N. Pramstaller, C. Rechberger, M. Kontak, J. Szmidt, Cryptanalysis of the GOST Hash function, in *CRYPTO*, ed. by D. Wagner. Lecture Notes in Computer Science, vol. 5157 (Springer, Berlin, 2008), pp. 162–178
[24] National Soviet Bureau of Standards. Information Processing System—Cryptographic Protection—Cryptographic Algorithm GOST 28147-89 (1989)

[25] K. Nyberg (ed.), *Fast Software Encryption, 15th International Workshop, Revised Selected Papers, FSE 2008*, Lausanne, Switzerland, 10–13 February, 2008. Lecture Notes in Computer Science, vol. 5086 (Springer, Berlin, 2008)
[26] A. Poschmann, S. Ling, H. Wang, 256 bit standardized crypto for 650 GE-GOST revisited, in *CHES*, ed. by S. Mangard, F.-X. Standaert. Lecture Notes in Computer Science, vol. 6225 (Springer, Berlin, 2010), pp. 219–233
[27] B. Preneel, T. Takagi (eds.), *Proceedings Cryptographic Hardware and Embedded Systems—CHES 2011—13th International Workshop*, Nara, Japan, September 28–October 1, 2011. Lecture Notes in Computer Science, vol. 6917 (Springer, Berlin, 2011)
[28] V. Rudskoy, On zero practical significance of "Key recovery attack on full GOST block cipher with zero time and memory". Cryptology ePrint Archive, Report 2010/111 (2010). http://eprint.iacr.org/
[29] M.-J.O. Saarinen, A chosen key attack against the secret S-boxes of GOST. Unpublished manuscript (1998)
[30] Y. Sasaki, K. Aoki, Finding preimages in full MD5 faster than exhaustive search, in *EUROCRYPT*, ed. by A. Joux. Lecture Notes in Computer Science, vol. 5479 (Springer, Berlin, 2009), pp. 134–152
[31] B. Schneier, Description of a new variable-length key, 64-bit block cipher (Blowfish), in *FSE*, ed. by R.J. Anderson. Lecture Notes in Computer Science, vol. 809 (Springer, Berlin, 1993), pp. 191–204
[32] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edn. (Wiley, New York, 1995)
[33] H. Seki, T. Kaneko, Differential cryptanalysis of reduced rounds of GOST, in *SAC*, ed. by D.R. Stinson, S.E. Tavares. Lecture Notes in Computer Science, vol. 2012 (Springer, Berlin, 2011), pp. 315–323
[34] M. Steil, 17 Mistakes Microsoft Made in the Xbox Security System (2005)
[35] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, T. Shirai, Piccolo: an ultra-lightweight blockcipher, in *[27]* (2011), pp. 342–357
[36] D.J. Wheeler, R.M. Needham, TEA, a tiny encryption algorithm, in *FSE*, ed. by B. Preneel. Lecture Notes in Computer Science, vol. 1008 (Springer, Berlin, 1994), pp. 363–366