# Improved Slide Attacks

Eli Biham*[1] Orr Dunkelman*[2] Nathan Keller**[3]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
biham@cs.technion.ac.il
[2]Katholieke Universiteit Leuven,
Dept. of Electrical Engineering ESAT/SCD-COSIC.
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
orr.dunkelman@esat.kuleuven.be
[3]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il

**Abstract.** The slide attack is applicable to ciphers that can be represented as an iterative application of the same keyed permutation. The slide attack leverages simple attacks on the keyed permutation to more complicated (and time consuming) attacks on the entire cipher.
In this paper we extend the slide attack by examining the cycle structures of the entire cipher and of the underlying keyed permutation. Our method allows to find slid pairs much faster than was previously known, and hence reduces the time complexity of the entire slide attack significantly. In addition, since our attack finds as many slid pairs as the attacker requires, it allows to leverage all types of attacks on the underlying permutation (and not only simple attacks) to an attack on the entire cipher.
We demonstrate the strength of our technique by presenting an attack on 24-round reduced GOST whose S-boxes are unknown. Our attack retrieves the unknown S-boxes as well as the secret key with a time complexity of about $2^{63}$ encryptions. Thus, this attack allows an easier attack on other instances of GOST that use the same S-boxes. When the S-boxes are known to the attacker, our attack can retrieve the secret key of 30-round GOST (out of the 32 rounds).

## 1 Introduction

Most of the modern block ciphers are constructed as a cascade of repeated keyed components, called rounds (or round functions). The security of such ciphers relies on applying the round function sufficiently many times. This is mainly due

* This work was supported in part by the Israel MOD Research and Technology Unit.
** The research presented in this paper was supported by the Adams fellowship.

to the fact that most attacks on block ciphers, e.g., differential cryptanalysis [3] and linear cryptanalysis [16], are statistical in nature and their effectiveness reduces as the number of rounds increases. Even non-statistical techniques, such as the SQUARE attack [7], are also affected by increasing the number of rounds of the cipher.

There are only few attacks on block ciphers that are independent of the number of rounds. One of them is the *related key attack* [1] presented by Biham in 1993. The attack uses encryption under unknown but related keys to derive the actual values of the keys.

In 1999, Biryukov and Wagner explored the framework of related key attacks and introduced the *slide attack* [4]. The slide attack is applied against ciphers that are a cascade of some identical "simple" functions, e.g., $E_k = f_k^l = f_k \circ f_k \circ \cdots \circ f_k$, where $f_k$ is a relatively weak keyed permutation. The attacker uses the birthday paradox to find *slid pairs* $(P_1, P_2)$ such that $P_2 = f_k(P_1)$. Due to the structure of $E_k$, the corresponding ciphertexts $(C_1, C_2)$ satisfy that $C_2 = f_k(C_1)$. Once a slid pair is found, the attacker uses the "simplicity" of $f_k$ and the input/output pairs $(P_1, P_2)$ and $(C_1, C_2)$ to attack $f_k$, and thus, to attack $E_k$. The attack can be used only if $f_k$ is "simple", i.e., can be broken using only two known input/output pairs.

In 2000, Biryukov and Wagner presented new variants of the slide attack, called *complementation slide* and *sliding with a twist* [5]. These variants allow for treating more complex functions in the slide framework. Nevertheless, there are no widely used ciphers that can be attacked using these techniques.

In addition to these new variants, the authors of [5] presented several techniques aimed at finding several slid pairs simultaneously, thus enabling to use the attack even if several input/output pairs are required to break $f_k$. One of these techniques, fully explored by Furuya [9], uses the fact that if $(P_1, P_2)$ is a slid pair then $(E_k^t(P_1), E_k^t(P_2))$ are also slid pairs for all values of $t$. This technique allows the attacker to transform any known plaintext attack on $f_k$ that requires $m$ known plaintexts to an attack on $E_k$. The data complexity of the attack is $O(m \cdot 2^{n/2})$ adaptively chosen plaintexts, where $n$ is the block size, and the time complexity of $O(2^n)$ applications of the known plaintext attack. Other attacks on the underlying function, e.g., chosen plaintext attacks, can be also leveraged to an attack on the entire cipher using the standard transformation to a known plaintext attack. However, in this case both the data complexity and the time complexity become very large, since the attack on the underlying function is applied $O(2^n)$ times during the attack on the whole cipher.

In this paper we present a new variant of the slide attack which enables to find slid pairs much more quickly. In our attack, instead of constructing slid pairs by the birthday paradox, we use the cycle structures of $E_k$ and $f_k$ for the detection of a large set of slid pairs. Then, we can use these pairs in any attack on $f_k$, whether it is a known plaintext attack, or if enough pairs are found, a chosen plaintext attack (or even an adaptive chosen plaintext and ciphertext attack). Unlike previous attacks, in our attack, the attack on $f_k$ is repeated only once (as the slid pairs are given), no matter what $f_k$ is.

The data complexity of our attack is high — it requires almost the entire codebook. On the other hand, the time complexity is surprisingly low — the first stage of the attack (finding the slid pairs) takes no more than $2^n$ encryptions, where $n$ is the block size, while the second stage is a (much faster) attack on $f_k$. Despite the large data requirements, the attack can be useful, as we demonstrate in our attack on 24-round reduced GOST with unknown S-boxes.

The block cipher GOST [10] is the Russian Encryption Standard. Since its publication in 1989 it withstood extensive cryptanalytic efforts. GOST has 64-bit blocks and keys of 256 bits. The main special feature of GOST is the S-boxes that were not published. Every industry was issued a different set of secret S-boxes. An example of a set that leaked is the set used in the Russian banking industry [19], and few attacks have been published on variants of GOST with these S-boxes [12, 15, 20]. There are also attacks on several variants of GOST with unknown S-boxes, but these attacks are either applicable to only a small number of rounds [20] or to relatively small classes of weak keys [4]. Another attack retrieves the unknown S-boxes by choosing a key from a set of weak keys for which it is possible to apply the slide attack [18].

In this paper we apply our technique to devise an attack on GOST reduced to its 24 first rounds with unknown S-boxes that allows to retrieve both the S-boxes and the secret key, a total of 768 key bits. The data complexity of our attack is $2^{63}$ adaptively chosen plaintexts and the time complexity is $2^{63}$ encryptions. A known plaintext variant of the attack has data and time complexities of a little less than $2^{64}$ encryptions.

A possible application of our attack is by an authorized user that has legitimate access to an instance of 24-round GOST that cannot be reverse engineered. Such a user can apply our attack to find the unknown S-boxes. As these S-boxes are shared by an entire industry, such an attack compromises the security of the entire industry.

When the S-boxes are known, it is possible to apply our attack to 30-round GOST (out of the 32 rounds), such that it is still faster than exhaustive key search. In addition, we present a new class of weak keys consisting of $2^{128}$ keys, for which our attack can break the entire GOST with unknown S-boxes with the same data and time complexities as the 24-round attack.

This paper is organized as follows: In Section 2 we give a brief description of the related key attacks and the slide attacks. In Section 3 we present our new technique. In Section 4 we present an attack on 24-round GOST with unknown

S-boxes and on 30-round GOST with known S-boxes. Appendix A outlines a 7-round truncated differential of GOST with probability 0.494. Finally, Section 5 summarizes this paper.

## 2 Related-Key Attacks and Slide Attacks

In this section we survey the previous results on related-key attacks. We start with describing the related-key attacks suggested by Biham [1] and Knudsen [14]. Then, we describe the slide attacks by Biryukov and Wagner [4]. We also present some of the extensions of the slide attack.

### 2.1 Related-Key Attacks

Related-key attacks, introduced in [1, 14], are attacks exploiting related-key plaintext pairs. The main idea behind the attack is to find instances of keys for which the encryption processes deploy the same permutation (or almost the same permutation). To illustrate the technique, we shortly present the attack from [1].

Consider a variant of DES in which all the rotate left operations in the key schedule algorithm are by a fixed number of bits.[1] For any key $K$ there exists another key $\hat{K}$ such that the round subkeys $KR_i, \widehat{KR}_i$ produced by $K$ and $\hat{K}$, respectively, satisfy:

$$KR_{i+1} = \widehat{KR}_i, \qquad \text{for } i = 1, \dots, 15.$$

For such pair of keys, if a pair of plaintexts $(P_1, P_2)$ satisfies $P_2 = f_{KR_1}(P_1)$, where $f_{sk}(P)$ denotes one round DES encryption of $P$ under the subkey $sk$, then the corresponding ciphertexts, $C_1$ and $C_2$, respectively, satisfy $C_2 = f_{\widehat{KR}_{16}}(C_1)$. Given such a pair of plaintexts (called in the sequel "a related-key plaintext pair"), the subkeys $KR_1$ and $\widehat{KR}_{16}$ can be easily extracted [1]. Due to the Feistel structure of DES, the related-key plaintext pairs can be detected easily. Therefore, this attack can be applied using only $2^{16}$ chosen plaintexts encrypted under $K$ and $2^{16}$ chosen plaintexts encrypted under $\hat{K}$, with a time complexity of $2^{17}$ encryptions.

In the more general case, the related-key attack has three parts: Obtaining related-key plaintexts, identifying the related-key plaintext pairs, and using them to deduce the key. In many cases, identifying the related-key plaintext pairs is best achieved by assuming for each candidate pair that it is a related-key plaintext pair, and then using it as an input for the key recovery phase of the

[1] Such a variant was proposed by Brown and Seberry [6].

attack. In other cases, the round functions' weaknesses allow the attacker to identify these pairs easily.

### 2.2 Slide Attacks

When a cipher has self-related keys, i.e., it can be written as $E_k = f_k^l = f_k \circ f_k \circ \cdots \circ f_k$, then it is susceptible to a variant of the related-key attack called the *slide attack* [4]. In this case, it is possible to apply the related-key attack to the cipher with $K = \hat{K}$, thus eliminating the key requirement of having two keys. The attacker looks for a slid pair, i.e., two plaintexts $(P_1, P_2)$ such that $P_2 = f_k(P_1)$. In this case, the pair satisfies $C_2 = f_k(C_1)$ as well. When the round function $f_k$ is simple enough, it is possible to use these two pairs in order to deduce information about the key.

In the slide attack the attacker obtains enough plaintext/ciphertext pairs to contain a slid pair, and has to check for each possible pair of plaintexts whether it is a slid pair by applying the attack on $f_k$. When dealing with a general block cipher this approach requires $O(2^{n/2})$ known plaintexts and $O(2^n)$ applications of the attack on $f_k$, where $n$ is the block size. For Feistel block ciphers, the attack can be optimized using $O(2^{n/4})$ chosen plaintexts and $O(1)$ applications of the attack. Note that as in the original related-key attacks, the main drawback of this approach is that the attack can be used only if $f_k$ can be broken using only two known input/output pairs, i.e., given one slid pair.

In 2000, Biryukov and Wagner [5] presented two variants of the slide attack, named *complementation slide* and *sliding with a twist*. These variants allow for treating more complex functions in a slide attack. Nevertheless, there are no widely used ciphers that can be attacked using these techniques.

The authors of [5] also presented several techniques aimed at finding several slid pairs simultaneously, enabling to use the attack even if several input/output pairs are needed for attacking $f_k$. One of these techniques, fully explored by Furuya [9], uses the fact that when $(P_1, P_2)$ is a slid pair then $(E_k^t(P_1), E_k^t(P_2))$ are also slid pairs for all values of $t$. This allows the attacker to transform any known plaintext attack on $f_k$ that requires $m$ known plaintexts to an attack on $E_k$ with a data complexity of $O(m \cdot 2^{n/2})$ adaptively chosen plaintexts. The time complexity of this approach is $O(2^n)$ applications of the known plaintext attack on $f_k$. It is worth mentioning that the technique can be easily improved when $f_k$ is one Feistel round, for which $O(2^{n/4})$ chosen plaintexts and $O(m)$ adaptive chosen plaintexts are sufficient to achieve $m$ slid pairs, which are easily identified.

This approach can be also used if there is only a chosen plaintext attack on $f_k$. The attacker can repeatedly generate slid pairs until the "plaintext" parts of the slid pairs contain a set of plaintexts satisfying the data requirements of the attack on $f_k$. However, the attack still requires $O(2^n)$ applications of the attack on $f_k$ (unless there is a special property that allows easy identification of the slid pairs).

## 3 Our New Technique

Before we start our discussion, let us recall the notations used in the previous section. In our discussion, $E_k = f_k^l$ is a block cipher composed of $l$ applications of the same keyed permutation $f_k$. The attacker looks for a slid pair, i.e., a pair of plaintexts $(P_1, P_2)$ such that $P_2 = f_k(P_1)$.

### 3.1 Studying the Cycle Structure

As noted earlier, in the previous variants of the slide attack there was no immediate indication which of the possible pairs is a slid pair (for a general cipher). Thus, an attacker had to try all possible pairs as candidates.

In our attack we use the cycle structure of $E_k$ and of $f_k$ to find a large amount of slid pairs. We emphasize that the attack uses only the cycle structure and is independent of any other properties of the functions $E_k$ and $f_k$. As a result, the attacker can detect the slid pairs even if there is no efficient attack on $f_k$. After finding the slid pairs, the attacker can use them to mount any attack she wishes on $f_k$, in order to retrieve the key material used in $f_k$.

First we recall several facts regarding the cycle structure of permutations that are used in our attack. Let $g : GF(2^n) \to GF(2^n)$ be a random permutation. For every $x \in GF(2^n)$ we denote $CycleLength(x) = \min\{k > 0 | g^k(x) = x\}$. The lengths of the cycles of $g$ are close to be uniformly distributed [8]. The expectation of the cycle length is $E[CycleLength(x)] = 2^{n-1}$. Therefore, we expect that the largest cycle $Cycle_{max}$ has a size of $O(2^{n-1})$ and for every $x \in GF(2^n)$, $\Pr[x \in Cycle_{max}] \approx 1/2$. Moreover, we expect that the second largest cycle has a size of $O(2^{n-2})$, and generally, the $i$-th largest cycle has a size of $O(2^{n-i})$. We note that for most of the permutations there are about $n$ cycles that are distributed according to a Poisson distribution (confirming the above claims for most of the cases) [11].

The main observation used in our attack is the following: Let $P$ be a randomly chosen plaintext. We consider the cycles of $P$ with respect to $f_k$ and $E_k$, denoted by $Cycle_{f_k}(P)$ and $Cycle_{E_k}(P)$, respectively. Denote the length of $Cycle_{f_k}(P)$ by $m_1$, i.e., $m_1 = \min\{t > 0 | f_k^t(P) = P\}$. Similarly, denote the length of $Cycle_{E_k}(P)$ by $m_2$. As $E_k = f_k^l$ the relation $E_k^{m_2}(P) = P$ can be written as $f_k^{l \cdot m_2}(P) = P$. Thus, from the definition of cycle lengths $m_1 | l \cdot m_2$. On the other hand, $m_2 = \min\{t > 0 | E_k^t(P) = P\} = \min\{t > 0 | f_k^{(t \cdot l) \bmod m_1}(P) = P\}$. Combining these two statements, we get $m_2 = \min\{t > 0 | tl = 0 \bmod m_1\}$. We conclude that

$$m_2 = m_1 / gcd(m_1, l).$$

In particular, if $gcd(m_1, l)$=1, then $m_1 = m_2$.

If $gcd(m_1, l) = 1$ we can find $1 \leq d_1 \leq l - 1$ and $d_2$ such that $d_1 \cdot m_1 = (-1) \pmod{l} = d_2 \cdot l - 1$ using Euclid's extended algorithm. These two values $d_1$ and $d_2$ satisfy both $f_k^{d_1 \cdot m_1 + 1}(P) = f_k(P)$ and $f_k^{d_1 \cdot m_1 + 1}(P) = f_k^{d_2 \cdot l}(P) = E_k^{d_2}(P)$. Therefore, the pair $(P, E_k^{d_2}(P))$ is a slid pair for $E$. Moreover, as already observed in [5], the pairs $(E_k^t(P), E_k^{d_2 + t}(P))$ are also slid pairs for every $t$.

When $gcd(m_1, l) \neq 1$, there is no general way to identify slid pairs as this case may be the product of several reasons. For example, when there is only one cycle (of size $2^n$), then the slid counterpart of any given plaintext $P$ is found in the cycle, but in an unknown location (or more precisely, any information that can be used to find the slid pair in the cycle, can be used without the need to construct the cycle).

In other cases, using the cycle structure can help a little bit. Assume that there are many cycles of different lengths, but a small group of cycles with the same size. It is very likely that this group contains slid pairs with one element in one cycle, and its slid counterpart in the other cycle. However, even if this is the case, the order between the different cycles, or even which is the counterpart of a given plaintext, is not necessarily disclosed by studying the cycle structure.

### 3.2 Using the Cycles in the Slide Attack

Our attack starts at some random plaintext $P_0$. We apply $E_k$ to it repeatedly until we get $P_0$ once again. When this happens, we have identified a cycle and the value of $CycleLength_{E_k}(P_0)$. Note that the attack cannot obtain the cycle of $f_k$, nor its length.

However, when the corresponding values $m_1$ and $l$ satisfy $gcd(m_1, l) = 1$ our attack can find the cycle length of $f_k$, as $m_1 = m_2$. Under this assumption we can calculate $d_2$, and then we get $m_2$ slid pairs in the cycle of $E_k$. These slid pairs can be used in any attack on $f_k$.

For a random value of $x$, $E[CycleLength(x)] = 2^{n-1}$, thus, we expect to get about $2^{n-1}$ slid pairs. This amount is sufficient for most possible attacks, including adaptive chosen plaintext and ciphertext attacks. In case the block ciphers ($f_k$ or $E_k$) have a different behavior, they can be easily distinguished from random permutations using this property.

In our attack, we assume that $m_1 = m_2$. In case this assumption is wrong, our attack fails. The probability of a failure for a random permutation $f_k$ is $1 - \varphi(l)/l$, where $\varphi(l) = |\{1 \leq d \leq (l-1) : gcd(d,l) = 1\}|$ is the Euler's function. In this case, we start with another plaintext $P_1 \notin Cycle_{E_k}(P_0)$ and repeat the attack. If the attack fails again, we take a new plaintext $P_2$ etc. Assuming that all the cycles lengths are independent of each other, after $t$ applications of the attack, the total success probability is $1 - (1 - \varphi(l)/l)^t$. This assumption is quite appropriate for random permutations, as long as $t$ is smaller than the total number of cycles.

The data complexity of the attack is very large — the attack requires $O(2^{n-1})$ adaptively chosen plaintexts. However, as we shall see in the attack on GOST, there are scenarios in which such an attack can be used despite the large data requirements.

We note that the attack can be transformed into a known plaintext attack that requires almost the entire code book. In that case, using birthday arguments we expect that few long cycles are found, suggesting a large amount of slid pairs.

## 4 Several Attacks on Reduced Round GOST

In this section we present a slide attack on 24-round GOST. The data complexity of the attack is about $2^{63}$ adaptive chosen plaintexts or $2^{64}$ known plaintexts (as required by our technique described in Section 3). The time complexity of the best attack we suggest is dominated by the time required to encrypt the plaintexts.

### 4.1 A Short Description of GOST

GOST [10] is a 64-bit block and 256-bit key cipher with a Feistel structure of 32 rounds. The round function accepts an input and a subkey of 32 bits each. The input and the subkey are added (modulo $2^{32}$), and the outcome is divided into eight groups of four bits each. Each such group enters a different S-box, where the least significant group enters $S1$, and the most significant group enters $S8$. The actual S-boxes that are used are kept secret, and are assigned by the government to a given set of users in each industry.[2] The outputs of all S-boxes are combined to 32 bits, which are then rotated to the left by 11 bits.

[2] We note that according to the published documentation [10], the S-boxes are not necessarily permutations. Hence, an S-box can be modeled as an unknown 64-bit key.

The key schedule algorithm takes the 256-bit key and treats it as eight 32-bit words, i.e., $K = K_1, \ldots, K_8$. The subkey $SK_r$ of round $r$ is

$$SK_r = \begin{cases} K_{(r-1) \bmod 8+1} & r \in \{1, \ldots, 24\}; \\ K_{33-r} & r \in \{25, \ldots, 32\}. \end{cases}$$

Thus, the 24 rounds of GOST we attack can be described as $f_k^3$, where $f_k$ is the first eight rounds of GOST.

There are few results on GOST with the S-boxes used in the banking industry: A differential attack on 13-round GOST requiring $2^{51}$ chosen plaintexts is described in [20] along with a related-key attack on 21-round GOST that requires $2^{56}$ related-key chosen plaintexts. A 24-round related-key attack (whose actual complexity and success rate depend on the S-boxes used) is described in [12]. A related-key attack on the full GOST with a data complexity of $2^{35}$ related-key chosen plaintexts and a time complexity of $2^{36}$ encryptions is presented in [15]. Only few results on GOST when the S-boxes are unknown were published before: A related-key distinguisher for the entire cipher that requires two related-key chosen plaintexts is presented in [15]. In [5] a slide attack on a 20-round variant of GOST in which the key additions are replaced by XOR with the key is described, along with a weak key class of $2^{128}$ keys that exists for that variant. A weak key class consisting of $2^{32}$ keys identified using a slide attack is presented in [9]. A chosen-key attack with a key in the weak key class that retrieves the unknown S-boxes is described in [18]. The attack uses $16 \cdot 2^{32}$ chosen plaintexts and has a running time of $2^{11}$ encryptions. The attack is a slide attack (that can be improved to an attack that requires $2^{17}$ chosen plaintexts), and the key can be any of the $2^{32}$ values $K = (K^*, K^*, K^*, K^*, K^*, K^*, K^*, K^*)$. However, we note that it is very easy to protect GOST against this attack, by preventing the usage of such keys.

### 4.2 Description of the Attack

As noted earlier, we attack 24 rounds of GOST for which $E_k = f_k^3$. Our attack first finds a large set of slid pairs $(P_a, P_b)$ and their corresponding ciphertexts $(C_a, C_b)$ such that $f_k(P_a) = P_b$ and $f_k(C_a) = C_b$. These slid pairs are found using the algorithm suggested in Section 3. Once these slid pairs are found, we apply a differential attack to $f_k$. We attack eight rounds of GOST using the following seven round differential that is independent of the actual S-boxes used:

$$P' = 00\ 01\ 00\ 00\ \ 00\ 00\ 00\ 00_x \rightarrow T' =??\ ??\ ??\ ??\ \ ?U_8\ L_9?\ ??\ ??$$

where ? denotes an unknown value, $L_9 \in \{0, 1_x, \ldots, 7_x\}$, and $U_8 \in \{0_x, 8_x\}$. The complete description of the differential is presented in Figure 1 in Appendix A. We note that as the differential is independent of the actual S-boxes used, we can apply the attack to any instance of GOST.

Roughly speaking, the attack takes pairs of slid pairs $(P_a, P_b)$ and $(P_c, P_d)$ for which the difference $P_a \oplus P_c$ equals the input difference of the differential, i.e., pairs for which $P_a \oplus P_c = P'$. As $P_b$ is the "encryption" of $P_a$ through $f_k$ and $P_d$ is the "encryption" of $P_c$ through $f_k$, then we treat the two plaintexts $P_b$ and $P_d$, as the ciphertexts in a differential attack on $f_k$. Once enough pairs of slid pairs are encountered, we partially decrypt the "ciphertexts" to find whether they satisfy the differential.

For the right guess of S-boxes and subkeys, the probability that a pair with input difference $P'$ has an output difference $T'$ is 0.494. This is a lower bound on the probability that the difference in the four bits 23–26 of the right half (left half after the swap operation) is zero. For a randomly selected pair of values this probability is $2^{-4}$. The way to check whether the differential holds, is to partially decrypt the pairs one round, and check whether the difference in these bits is 0 or not.

Our attack on GOST uses the following observations:

**Observation 1** *Each cycle suggests many pairs that can be used. Thus, we can impose more conditions on the ciphertexts we analyze, in order to reduce the time complexity of the attack.*

**Observation 2** *In order to determine the difference in bits 23–26 of the right half in round 7, it is sufficient to determine the output of $S4$ in round 8 (bits 12–15).*

**Observation 3** *It is possible to consider only ciphertexts that have predetermined inputs to the relevant S-box.*

Based on these observations, the attack algorithm is the following:

1. Obtain about $2^{43.5}$ slid pairs $(P_a, P_b)$ such that $f_k(P_a) = P_b$.
2. Identify $2^{22}$ pairs of slid pairs $(P_a, P_b)$ and $(P_c, P_d)$ such that $P_a \oplus P_c = P'$.
3. Consider the pairs of slid pairs $((P_a, P_b), (P_c, P_d))$ such that
   (a) The value of the right half of $P_b$ is $X0$ in bits 8–15, ($X$ is the value that enters $S4$ in round 8), where $X \in \{0, \ldots, F_x\}$ is a predetermined value.
   (b) The value of the right half of $P_d$ is $Z0$ in bits 8–15, ($Z$ is the value that enters $S4$ in round 8), where $Z \in \{0, \ldots, F_x\} \backslash \{X\}$ is a predetermined value.
4. For each remaining pair $((P_a, P_b), (P_c, P_d))$ of slid pairs:
   (a) Guess the four bits leaving $S4$ in round 8 for the pair $(P_a, P_b)$ (assuming that the value is the same for all the examined pairs), and guess the four bits leaving $S4$ in round 8 for the pairs $(P_c, P_d)$ (assuming that the value is the same for all the examined pairs).

(b) Partially decrypt $P_b$ and $P_d$ through $S4$, and check whether the difference in round 7 in bits 23–26 of the left half is 0.
(c) If the difference in these bits after the partial decryption is 0, increment a counter that corresponds to the specific guesses made (a total of 4+4=8 bits).
5. Output all the guesses whose corresponding counter has a value greater than 19.

### 4.3 Analysis of the Attack

For a wrong guess of the S-boxes outputs there is probability of $2^{-4}$ that a partially decrypted pair has zero difference in bits 23–26, thus incrementing the counter. For the correct guess, this probability is about 0.494.

Of the $2^{43.5}$ slid pairs, we expect that $(2^{43.5})^2/2 \cdot 2^{-64} = 2^{22}$ pairs of slid pairs have "plaintext" difference of $P'$. Out of these pairs, about $2^{22} \cdot (2^{-8})^2 = 2^6$ pairs are analyzed in Step 4 (due to the 8-bit condition on each of the ciphertexts).

In Step 4(a) and Step 4(b) the ciphertexts are partially decrypted through round 8. If the actual inputs to $S4$ in round 8 are fixed, then so do their outputs that compose the four output bits that are needed for the partial decryption. Thus, we choose a value for bits 12–15 (those entering S-box $S4$) for each of the two pairs, i.e., we require that one ciphertext has some arbitrary value $X$ in the S-box and that the second ciphertext has some other arbitrary value $Z$, where $X, Z \in \{0, \dots, F_x\}$. Setting these conditions does not necessarily imply that in all pairs the ciphertexts have the same four output bits (four from each ciphertext) due to the carry that the key addition may cause. Thus, we also require that in all the ciphertexts bits 8–11 have the value zero, and this reduces the probability that a carry changes the values entering S-box $S4$.[3]

For a wrong guess, the expected value of the counter is $2^6 \cdot 2^{-4} = 4$, while for the right guess it is $2^6 \cdot 0.494 = 31.6$. There are $2^8$ guesses, and the probability that one of them has a counter whose value is greater than 19 is less than $2^{-22}$, while the correct guess is suggested with probability of $1 - 2^{-7.6}$. We note that the attack returns two outputs of the S-box $S4$ in round 8. Using more pairs with different inputs to $S4$, enables mapping all the outputs of $S4$, up to the 16 combinations of key and S-box that are all equivalent (as the key bits are used to decide on the actual entries that lead to the output). The exact combination can be identified using different entries to the S-box and using auxiliary techniques.

[3] There is a difference in the carry when the four key bits that enter $S3$ are all 1's, and in addition there is a difference in the carry operation in the most significant bit that enters $S2$. If the attack fails, we deduce that this is the case, and we can continue to impose conditions on the ciphertexts, until either the attack succeeds, or we obtain the 12 least significant bits of $K_8$.

We conclude that our attack uses $2^{43.5}$ slid pairs. The time complexity of Step 4 in the attack is less than $2^8$ 24-round GOST encryptions for the retrieval of 8 bits of information about the key and $S4$. In this case, the time complexity of the attack is dominated mostly by the first step of the attack, i.e., finding the slid pairs, a step that requires $2^{63}$ encryptions.

After finding the entire $S4$ (up to the exact order, due to the addition with the key), we can use a rotated version of the differential (that may have a slightly smaller probability) to retrieve other S-boxes. The remaining key words can be found using auxiliary techniques (e.g., using the differential in the decryption direction).

As the attack finds the S-boxes along with the key it can be used by an authorized user who wishes to find the S-boxes used in a given implementation of GOST. In this case, the attacker can even know the encryption key (which can help in the process of the attack), and retrieve the unknown S-boxes in time complexity of $2^{63}$ encryptions (which as a valid user of GOST he can achieve). We note that the work of Saarinen in [18] addresses the same problem for a fixed key for which the slide properties are easily applied. However, it is easy to protect GOST from this attack by preventing the usage of keys of the form $(k, k, k, k, k, k, k, k)$.

We note that the data complexity is about $2^{63}$ adaptive chosen plaintexts, or about $2^{64} - 2^{18}$ known plaintexts. In the latter case, it is expected that three cycles of expected lengths of at least $2^{43.5}$ values are encountered. There is a high probability that one of these three cycles can be used in our attack (if the gcd of the cycle length with 3 is 1).

### 4.4 Other Results on GOST

Our attack can be extended for two cases: a weak key class of the full GOST that can be detected even when the S-boxes are unknown, and a 30-round attack when the S-boxes are known.

Our weak key class has $2^{128}$ keys of the form $K_1, K_2, K_3, K_4, K_4, K_3, K_2, K_1$, i.e., $K_{9-i} = K_i$. Thus, the last eight rounds define the same permutation as the first eight rounds. Hence, it is possible to treat the full GOST of this form as $GOST_K = f_K^4$, and apply our attack (even when the S-boxes are unknown). We note that this weak key class was suggested in [5] for a weakened variant of GOST where the key is XORed and not added.

The attack on 30-round GOST with known S-boxes is of the following nature:

- Ask for almost the entire code book

- For each guess of $K_3, K_4, \ldots, K_8$:
  - Partially decrypt all the ciphertexts through rounds 30–25
  - Apply a variant of the 24-round attack described earlier
  - If the attack succeeds, output the key guess of $K_3, \ldots, K_8$
- Exhaustively search over all possible values of $K_1, K_2$

The time complexity of this attack is equivalent to partially decrypting each ciphertext $2^{192}$ times. As the attack requires almost the entire code book, then the actual time complexity of the attack is almost $2^{256}$ partial decryptions which are equivalent to $2^{253.7}$ 30-round GOST encryptions.

## 5 Summary and Conclusions

In this paper we have presented a new variant of the slide attack. Our attack uses the relation between the cycle structure of the entire cipher and that of the underlying permutation, and allows to detect a large amount of slid pairs in an efficient way. These pairs are then used to mount various attacks on the underlying permutation.

The new technique allows us to attack 24-round GOST, even when the S-boxes are unknown, and to retrieve both the key and the unknown S-boxes. When the S-boxes are known, this attack can be extended to up to 30-round GOST. In addition, for a weak key class of GOST containing $2^{128}$ weak keys, the attack is applicable against the full GOST with unknown S-boxes. All the attacks have a data complexity of a little less than $2^{64}$ known plaintexts (or $2^{63}$ adaptively chosen plaintexts) with time complexity of $2^{64}$ (besides the 30-round attack whose time complexity is $2^{253.7}$). The 24-round attack reveals the S-boxes used in GOST, and thus it can be used by an authorized user who wishes to declassify the S-boxes he was given.

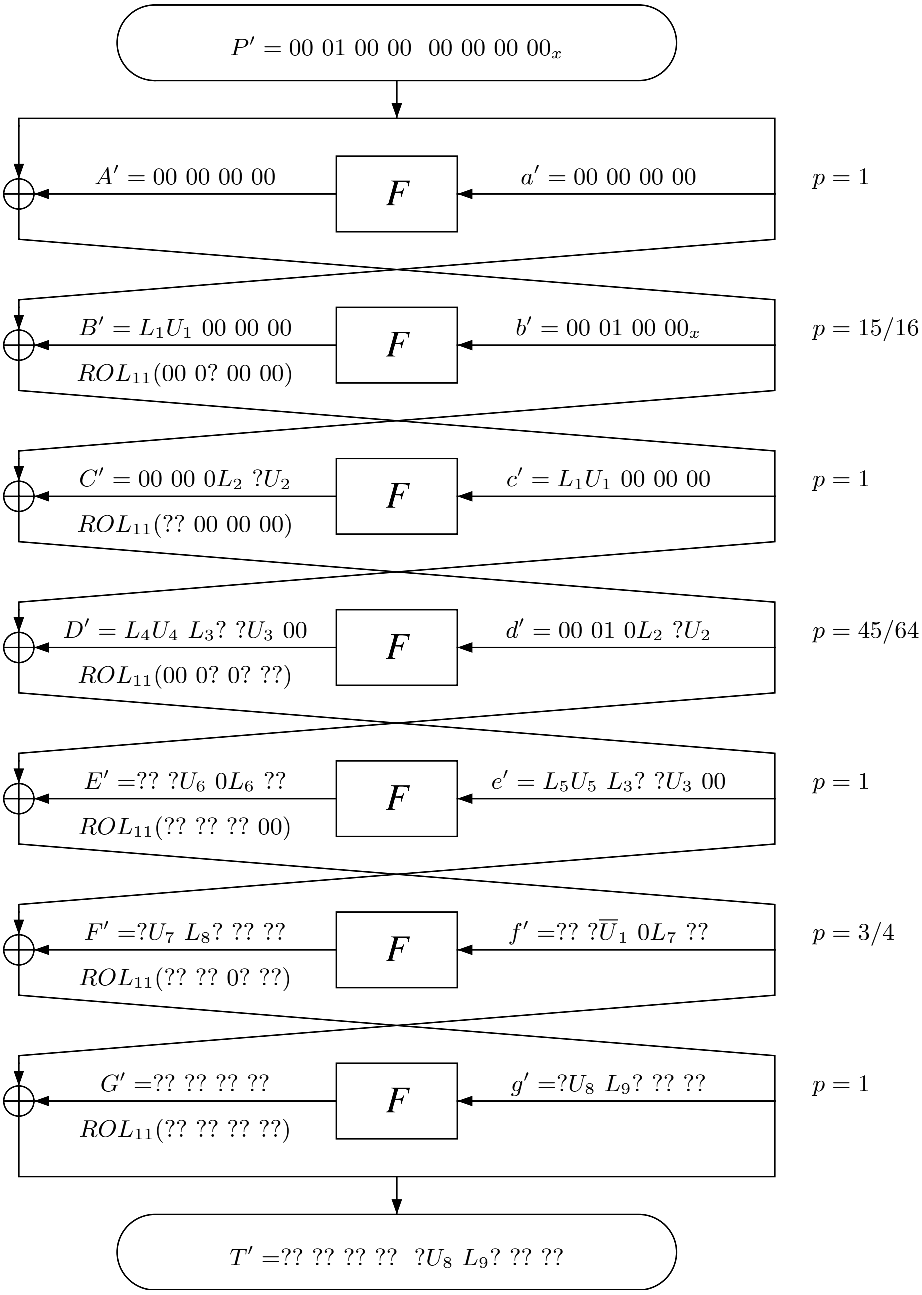
## References


1. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, Springer-Verlag, 1994.
2. Eli Biham, Alex Biryukov, Adi Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.
3. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
4. Alex Biryukov, David Wagner, *Slide Attacks*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 245–259, Springer-Verlag, 1999.

5. Alex Biryukov, David Wagner, *Advanced Slide Attacks*, Advances in Cryptology, proceedings of EUROCRYPT 2000, Lecture Notes in Computer Science 1807, pp. 586–606, Springer-Verlag, 2000.
6. Lawrence Brown, Jennifer Seberry, *Key Scheduling in DES Type Cryptosystems*, proceedings of AUSCRYPT 1990, Lecture Notes in Computer Science 453, pp. 221–228, Springer-Verlag, 1990.
7. Joan Daemen, Lars R. Knudsen, Vincent Rijmen, *The Block Cipher Square*, proceedings of Fast Software Encryption 4, Lecture Notes in Computer Science 1267, pp. 149–165, Springer-Verlag, 1997.
8. Donald W. Davies, Graeme I. P. Parkin, *The Average Cycle Size of the Key Stream in Output Feedback Encipherment (Abstract)*, Advances in Cryptology, proceedings of CRYPTO 1982, pp. 97–98, Plenum, New York, 1982.
9. Soichi Furuya, *Slide Attacks with a Known-Plaintext Cryptanalysis*, proceedings of Information and Communication Security 2001, Lecture Notes in Computer Science 2288, pp. 214–225, Springer-Verlag, 2002.
10. GOST, Gosudarstvennei Standard 28147-89, *Cryptographic Protection for Data Processing Systems*, Government Committee of the USSR for Standards, 1989.
11. Andrew Granville, *Cycle lengths in a permutation are typically Poisson distributed*, Electronic Journal of Combinatorics, Vol. 13, No. 1, R107, 2006. Available on-line at: *http://www.dms.umontreal.ca/∼andrew/PDF/CycleLengths.pdf*, 2006.
12. John Kelsey, Bruce Schneier, David Wagner, *Key-Schedule Cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, Advances in Cryptology, proceedings of CRYPTO '96, Lecture Notes in Computer Science 1109, pp. 237–251, Springer-Verlag, 1996.
13. John Kelsey, Bruce Schneier, David Wagner, *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA*, proceedings of Information and Communication Security 1997, Lecture Notes in Computer Science 1334, pp. 233–246, Springer-Verlag, 1997.
14. Lars R. Knudsen, *Cryptanalysis of LOKI91*, proceedings of Auscrypt 1992, Lecture Notes in Computer Science 718, pp. 196–208, Springer-Verlag, 1993.
15. Youngdai Ko, Seokhie Hong, Wonil Lee, Sangjin Lee, Ju-Sung Kang, *Related Key Differential Attacks on 27 Rounds of XTEA and Full-Round GOST*, proceedings of Fast Software Encryption 11, Lecture Notes in Computer Science 3017, pp. 299–316, Springer-Verlag, 2004.
16. Mitsuru Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 386–397, Springer-Verlag, 1994.
17. National Bureau of Standards, *Data Encryption Standard*, Federal Information Processing Standards Publications No. 46, 1977.
18. Markku-Juhani Saarinen, *A Chosen Key Attack against the Secret S-boxes of GOST*, 1998. Available on-line at *http://citeseer.ist.psu.edu/saarinen98chosen.html*.
19. Bruce Schneier, *Applied Cryptography, Second Edition*, John Wiley & Sons, 1996.
20. Haruki Seki, Toshinobu Kaneko, *Differential Cryptanalysis of Reduced Rounds of GOST*, proceedings of Selected Areas in Cryptography 2000, Lecture Notes in Computer Science 2012, pp. 315–323, Springer-Verlag, 2001.
21. Gideon Yuval, *Reinventing the ~~wheel~~ Travois: Encryption/MAC in 30 ROM Bytes*, Proceedings of Fast Software Encryption 4, Lecture Notes in Computer Science 1267, pp. 205–209, Springer-Verlag, 1997.

## A A 7-Round Differential of GOST with Unknown S-boxes

The input difference of the differential is of the form $P' = 00\ 01\ 00\ 00\ \ 00\ 00\ 00\ 00_x$. The zero difference enters the first round and has a zero output difference. In the second round a difference of $00\ 01\ 00\ 00_x$ enters the round function. As there is an addition, with probability 15/16 the differences in the carry of the addition operation (if there are such) do not affect other S-boxes. Thus, there is a non-zero input difference to one of the S-boxes of round 2, while all the rest have a zero input difference. The differential evolves, and after seven rounds there are four bits whose difference is known to be zero.




? denotes an unknown value.
$U_i \in \{0, 8_x\}, L_i \in \{0, 1_x, \ldots, 7_x\}, \overline{U}_1 \in \{1_x, 9_x\}$

**Fig. 1.** A 7-Round Differential of GOST with Probability $\frac{15}{16} \cdot \frac{45}{64} \cdot \frac{3}{4} = 0.494$.