Advanced Slide Attacks

Alex Biryukov[*] and David Wagner[**]

Abstract. Recently a powerful cryptanalytic tool—the slide attack—was introduced [3]. Slide attacks are very successful in breaking iterative ciphers with a high degree of self-similarity and even more surprisingly are independent of the number of rounds of a cipher. In this paper we extend the applicability of slide attacks to a larger class of ciphers. We find very efficient known- and chosen-text attacks on generic Feistel ciphers with a periodic key-schedule with four independent subkeys, and consequently we are able to break a DES variant proposed in [2] using just 128 chosen texts and negligible time for the analysis (for one out of every 2^{16} keys). We also describe *known-plaintext* attacks on DESX and Even-Mansour schemes with the same complexity as the best previously known *chosen-plaintext* attacks on these ciphers. Finally, we provide new insight into the design of GOST by successfully analyzing a 20-round variant (GOST$\oplus$) and demonstrating weak key classes for all 32 rounds.

1 Introduction

The *slide attack* is a powerful new method of cryptanalysis of block-ciphers introduced in [3]. The unique feature of this new cryptanalytic attack is its independence of the number of rounds used in the cipher of interest: when a slide attack is possible, the cipher can be broken no matter how many rounds are used. This capability is indispensable in a study of modern iterative block ciphers and hash functions. As the speed of computers grows, it is natural to use more and more rounds, which motivates our study of attacks that are independent of the number of rounds. While addition of a few rounds usually stops even a very sophisticated cryptanalytic attack (such as a differential or linear attack), in contrast a cipher vulnerable to slide attacks cannot be strengthened by increasing the number of its rounds. Instead, one must change the key-schedule or the design of the rounds.

In [3] it was shown that slide attacks exploit the degree of *self-similarity* of a block cipher and thus are applicable to iterative block-ciphers with a periodic key-schedule. It was also shown that slide attacks apply to auto-key ciphers (where the choice of the round subkeys is data-dependent). As an example an attack was presented on modified Blowfish [17], a cipher based on key-dependent S-boxes which so far had resisted all the conventional attacks.

[*] Applied Mathematics department, Technion - Israel Institute of Technology, Haifa, Israel 32000, and Computer Science department, The Weizmann Institute of Science, Rehovot 76100, Israel. Email: `albi@wisdom.weizmann.ac.il`

[**] University of California, Berkeley. Email: `daw@cs.berkeley.edu`

Table 1. Summary of our attacks on various ciphers.

Cipher	(Rounds)	Key bits	Best Previous Attack			Our Attack		
			Data	Type	Time	Data	Type	Time
2K-DES	(∞)	96	2^{32}	KP	2^{50}	2^{32}	KP	2^{33}
2K-DES	(∞)	96	2^{32}	KP	2^{50}	2^{17}	CP/CC	2^{17}
4K-Feistel	(∞)	192	—	–	—	2^{32}	KP	2^{33}
4K-Feistel	(∞)	192	—	–	—	2^{17}	CP/CC	2^{17}
4K-DES*	(∞)	192	—	–	—	2^{17}	CP/CC	2^{17}
Brown-Seberry-DES*	(∞)	56	—	–	—	128	CP/CC	2^{7}
DESX	(16)	184	2^{m}	CP	2^{121-m}	$2^{32.5}$	KP	$2^{87.5}$
DESX	(16)	184	2^{m}	CP	2^{121-m}	$2^{32.5}$	CO	2^{95}
Even-Mansour	(—)	$2n$	$2^{n/2}$	CP	$2^{n/2}$	$2^{n/2}$	KP	$2^{n/2}$
GOST⊕	(20)	256	—	–	—	2^{33}	KP	2^{70}

CO — ciphertext-only, KP — known-plaintext, CP — chosen-plaintext, CP/CC — chosen plaintext/ciphertext. * – Our attack on 4K-DES and Brown-Seberry-DES works for $1/2^{16}$ of all keys. Note that attacks on 2K-DES work for all the keys.

The existence of attacks which are independent of the number of rounds is perhaps counter-intuitive. To illustrate this consider a quote from [15]:

> "Except in a few degenerate cases, an algorithm can be made arbitrarily secure by adding more rounds."

Slide attacks force us to revise this intuition, and this motivates our detailed study of advanced sliding techniques.

In this paper we introduce advanced sliding techniques—*sliding with a twist* and the *complementation slide*—that result in a more efficient slide attacks and allow to attack new classes of ciphers. We illustrate these techniques on generic Feistel constructions with two- or four-round self-similarity as well as a Luby-Rackoff construction and also the example ciphers 2K-DES and 4K-DES, which differ from DES only by having 64 rounds, a 96- or 192-bit key, and a simplified (periodic) key-schedule. Analysis of these ciphers is of independent interest since it demonstrates the dangers of some ways to extend DES. Specifically we show a very efficient attack on a variant of DES proposed in [2]: our attack uses only 128 chosen texts and negligible time of analysis (for a 2^{-16} fraction of all keys).

We then apply the newly developed methods to the DESX and Even-Mansour schemes, and we show known-plaintext slide attacks with the same complexity as the best previously known chosen-plaintext attacks. We also apply slide attacks to the GOST cipher (a Russian equivalent of DES) obtaining insights on its design.

See Table 1 for a summary of our results. For each cipher a number of rounds that our attack is able to cover is presented; ∞ is shown if our attack is independent of the number of rounds of a cipher. The block size in bits is denoted by n, and the 'Key bits' column denotes the number of secret key bits of the cipher.

This paper is organized as follows: In Section 2 we briefly describe conventional slide attacks. We develop several advanced sliding techniques in Section 3, illustrating them on generic Feistel ciphers with periodic key-schedules. As a side effect we receive a distinguishing attack on the $\Psi(f, g, f, g, \ldots, f, g)$ Luby-Rackoff construction (see the end of Section 3.2). We then apply the newly developed techniques to the analysis of DESX and Even-Mansour schemes in Section 4. In Section 5 we turn advanced slide attacks to the analysis of GOST. Finally Section 6 summarizes some related work and Section 7 outlines some possible directions for further research.

2 Conventional Slide Attacks

Earlier work [3] described a simple form of slide analysis applicable to ciphers with self-similar round subkey sequences or autokey ciphers. We briefly sketch those ideas here; see [3] for full details and cryptanalysis of a number of ciphers, and Section 6 for other related work.

In the simplest case, we have an r-round cipher E whose rounds all use the same subkey, so that $E = F \circ F \circ \cdots \circ F = F^r$. Note that if the key schedule of a cipher is periodic with period p, we can consider F to be a "generalized" round consisting of p rounds of the original cipher. We call such ciphers **p-round self-similar**. Let $\langle P, C\rangle$ be a known plaintext-ciphertext pair for E. The crucial observation is

$$P' = F(P) \quad \text{implies} \quad C' = E(P') = F^r(F(P)) = F(F^r(P)) = F(C).$$

In a standard slide attack, we try to find pairs $\langle P, C\rangle$, $\langle P', C'\rangle$ with $P' = F(P)$; we call such a pair a **slid pair**, and then we will get the extra relation $C' = F(C)$ "for free."

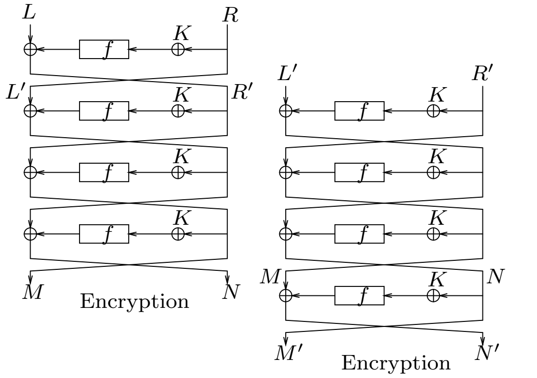

Fig. 1. A conventional slide attack on a generic Feistel cipher with one-round self-similarity. If $L' = R$ and $R' = L \oplus f(K \oplus R)$, the texts shown above will form a slid pair, and we will have $M' = N$ and $N' = M \oplus f(K \oplus N)$.

Slide attacks provide a very general attack on iterated product ciphers with repeating round subkeys. The only requirement on F is that it is very weak against known-plaintext attack with two pairs (we are able to relax this requirement later, in Section 3.5). More precisely, we call $F_k(x)$ a **weak** permutation if given the two equations $F_k(x_1) = y_1$ and $F_k(x_2) = y_2$ it is "easy" to extract the key k. Such a cipher (with a n-bit block) can be broken with only $2^{n/2}$ known texts, since then we obtain 2^n possible pairs $\langle P, C\rangle$, $\langle P', C'\rangle$; as each pair has a 2^{-n} chance of forming a slid pair, we expect to see one slid pair which discloses the key.

Feistel ciphers form an important special case for sliding, since the attack complexity can be substantially reduced from the general case. We depict in Figure 1 a conventional slide attack on a Feistel cipher with repeating round subkeys. The Feistel round structure gives us an n-bit filtering condition on slid pairs, which lets us reduce the complexity of analysis to about $2^{n/2}$ time and space, a significant improvement over the 2^n work required for the general attack listed above. Furthermore, there is a chosen-text variation which works against Feistel ciphers with about $2^{n/4}$ chosen plaintexts: we may simply use structures to 'bypass the first round'. See [3] for details.

In this paper, we focus on generalizing the slide attack to apply to a broader range of constructions.

3 Advanced Sliding Techniques

In this section we show several ways of extending the basic slide attack to apply to larger classes of ciphers. In the following subsections we introduce two new methods: the *complementation slide* and *sliding with a twist*.

We will describe these new techniques by applying them first to a generic Feistel cipher with a 64-bit block and self-similar round subkeys. (See Figure 1 for an example of such a cipher, where the subkeys exhibit one-round self-similarity. In this section, we consider up to four-round self-similarity.) For ease of illustration we will show graphically ciphers with only a small number of rounds, but we emphasize that the attacks described in this section apply to ciphers with any number of rounds. After describing the basic attack techniques we will show how to extend them to real ciphers.

3.1 The Complementation Slide

First we show a method to amplify self-similarity of Feistel ciphers with two-round self-similarity by exploiting its complementation properties, thus allowing for much better attacks. We call this approach the *complementation slide*.

In the conventional attack, to deal with two-round self-similarity one must slide by two rounds (thus achieving a perfect alignment of rounds with K_0 and K_1), but this yields inefficient attacks. In contrast, we suggest to slide by only one round. This introduces the difference $\Delta = K_0 \oplus K_1$ between slid encryptions in all the rounds. Notice that we have effectively amplified the self-similarity of the cipher from 2-round to 1-round self similarity. However together with amplified

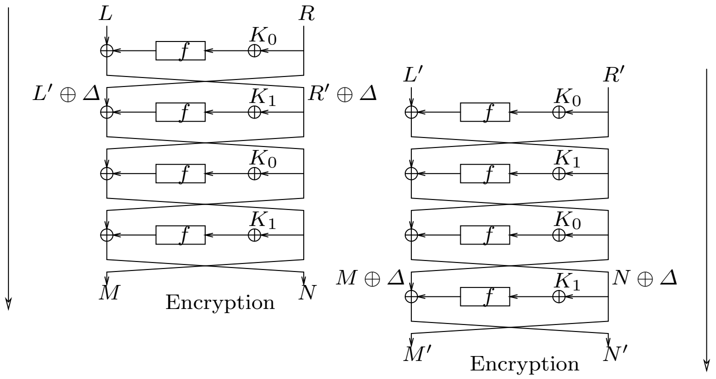

Fig. 2. A *complementation slide* attack on a Feistel cipher with two-round self-similarity. If $L' = R \oplus \Delta$ and $R' = L \oplus f(K_0 \oplus R) \oplus \Delta$, the texts shown above will form a slid pair, and we will have $M' = N \oplus \Delta$ and $N' = M \oplus f(K_1 \oplus N \oplus \Delta) \oplus \Delta$, where $\Delta = K_0 \oplus K_1$.

self-similarity we have introduced differences between rounds of encryption in a slid pair. How can the attack proceed?

Our answer is to choose a slid pair so that the plaintext differences will cancel the difference between the subkeys. Instead of searching for plaintexts with slid difference zero, we search for plaintexts with slid difference $\langle \Delta, \Delta \rangle$. (Note: We say that a pair of plaintexts P, P' has *slid difference* d if $F(P) \oplus P' = d$.) Such a slid difference will propagate with probability one through all the rounds, and thus will appear at the ciphertext. See Figure 2 for a pictorial illustration of the attack.

The slid pairs can be found in a pool of 2^{32} known plaintexts, as before. If we denote the plaintext by $P = \langle L, R \rangle$ and the ciphertext by $C = \langle M, N \rangle$, we get the following slid equations:

$$\langle L', R' \rangle = \langle R, L \oplus f(K_0 \oplus R) \rangle \oplus \langle \Delta, \Delta \rangle$$
$$\langle M', N' \rangle = \langle N, M \oplus f(K_1 \oplus N \oplus \Delta) \rangle \oplus \langle \Delta, \Delta \rangle.$$

Thus we have $L' \oplus M' = R \oplus N$ which is a 32-bit condition on a slid pair. Moreover the second equation suggests a 32-bit candidate for $\Delta = K_0 \oplus K_1$; if we have several slid pairs, this value should coincide for all of them (although we do not need the latter property in our attack). Thus the S/N ratio of this attack is very high. As soon as one slid pair is found, we derive $\Delta = K_0 \oplus K_1$. Then, if the round function f is weak enough, we will be able to derive the keys K_0 and K_1 themselves from the first and second equations. We will only need to examine 2^{31} pairs (due to the 32-bit filtering condition) and each pair suggests at most one candidate key, so the work-factor of the attack is very low.

To summarize, this gives a known plaintext attack on a generic Feistel cipher with two-round self-similarity. The complexity of the attack is quite realistic: we need just 2^{32} known texts and at most 2^{32} light steps of analysis. However, see Section 3.2 for an even better attack.

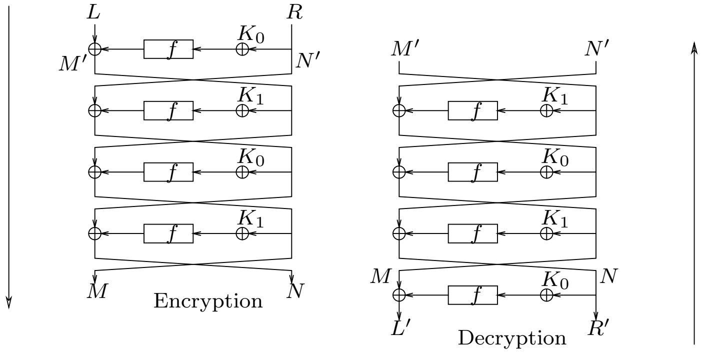

Fig. 3. *Sliding with a twist*, applied to a Feistel cipher with two-round self-similarity. If $N' = R$ and $M' = L \oplus f(K_0 \oplus R)$, the texts shown above will form a (twisted) slid pair, and we will have $R' = N$ and $L' = M \oplus f(K_0 \oplus N)$.

Even more interestingly: We can consider a variant with four independent subkeys, K_0, K_1, K_2, K_3, so that the key size is 128 bits. If we slide by two rounds we find that the XOR differences between subkeys are 2-round self-similar! A modified version of the above attack works, although the S/N ratio is not as high as before. Complementation sliding thus provides a powerful technique for amplifying self-similarity in iterated ciphers.

3.2 Sliding with a Twist

We next describe a novel technique of sliding with a *twist* on a Feistel cipher with two-round self-similarity. This allows for even better attacks than those presented above. See also our attack on DESX in Section 4 for an important application of sliding with a twist.

If we ignore the final swap for the moment, then decryption with a Feistel cipher under key K_0, K_1 is the same as encryption with key K_1, K_0[1]. Of course, Feistel encryption with key K_0, K_1 is very similar to encryption with key K_1, K_0: they are just out of phase by one round. Therefore, we can slide by one round a decryption process against an encryption process (the *twist*). This provides us with a slid pair with an overlap of all rounds except for one round at the top and one round at the bottom. Notice that due to the twist these rounds both use the same subkey K_0. See Figure 3 for a graphical depiction.

The attack begins by obtaining a pool of 2^{32} known texts, so that we expect to find one slid pair. For a slid pair, we have

$$\langle M', N' \rangle = \langle L \oplus f(K_0 \oplus R), R \rangle \qquad \langle L', R' \rangle = \langle M \oplus f(K_0 \oplus N), N \rangle$$

which gives us a 64-bit filtering condition on slid pairs (namely $N' = R$ and $R' = N$). Thus the slid pair can be easily found with a hash table and 2^{32} work, and it immediately reveals the subkey K_0.

[1] In [3] such cipher, based on DES was called 2K-DES.

The rest of the key material can be obtained in a second analysis phase with a simplified conventional sliding (by two rounds and without a twist) using the same pool of texts and with less than 2^{32} work. Pick a ciphertext from a pool, partially encrypt it with K_0 and search the pool of ciphertexts for one with coinciding 32 bits. If such a ciphertext is found perform a similar check on their plaintexts. If both conditions hold this is a slid pair that provides us with K_1. This attack requires just 2^{32} known texts and 2^{33} work.

Moreover, there is a chosen-plaintext/ciphertext variant that allows us to reduce the number of texts down to 2^{17} with the use of structures. We generate a pool of 2^{16} plaintexts of the form (L_i, R) and obtain their encryptions. Also, we build a pool of 2^{16} ciphertexts of the form (M'_j, N') and decrypt each of them, where the value $N' = R$ is fixed throughout the attack. This is expected to give one slid pair, and then the analysis proceeds as before.

This demonstrates that sliding with a twist is capable of attacking any n-bit Feistel block cipher with a two-round periodic key-schedule with $2^{n/2}$ known plaintexts and about $2^{n/2}$ time, or with about $2^{n/4}$ chosen plain-ciphertexts and about $2^{n/4}$ time. Also, sliding with a twist can be used to distinguish a Luby-Rackoff [13] construction with two alternating pseudo-random functions f and g and with an arbitrary number of rounds (an accepted notation is $\Psi(f, g, f, g, \ldots, f, g)$) from a random permutation with about $2^{n/2}$ known plaintexts and similar time (given that the block size is n bits), or with about $2^{n/4}$ chosen plaintext/ciphertext queries and similar time.

3.3 Better Amplification of Self-Similarity: Four-Round Periodicity

In this section we combine the *complementation slide* and *sliding with a twist* to amplify the self-similarity of round subkeys even further. Consider a Feistel cipher with key schedule that repeats every four rounds, using independent subkeys K_0, K_1, K_2, K_3, and suppose these keys are XORed at the input of the f-function. We call this generic cipher a *4K-Feistel* cipher.

One may naively slide by two rounds to amplify self-similarity, like this:

$$\begin{array}{l} K_0\ K_1\ K_2\ K_3\ K_0\ K_1\ \ldots \\ K_0\ K_1\ K_2\ K_3\ K_0\ K_1\ \ldots \end{array}$$

Then one may use a complementation slide technique using the slid difference $\langle K_1 \oplus K_3, K_0 \oplus K_2 \rangle$. However, there doesn't seem to be any way to make this attack work with less than $2^{n/2}$ texts, and the analysis phase is hard.

Better results are possible if one applies *sliding with a twist.* At a first glance, the twist may not seem to be applicable, but consider combining it simultaneously with the *complementation slide*, like this:

$$\begin{array}{l} K_0\ K_1\ K_2\ K_3\ K_0\ K_1\ K_2\ K_3\ K_0\ \ldots \\ K_3\ K_2\ K_1\ K_0\ K_3\ K_2\ K_1\ K_0\ K_3\ \ldots \end{array}$$

The top row represents an encryption, and the bottom represents a decryption (or, equivalently, encryption by K_3, K_2, K_1, K_0, due to the similarity between encryption and decryption in Feistel ciphers).

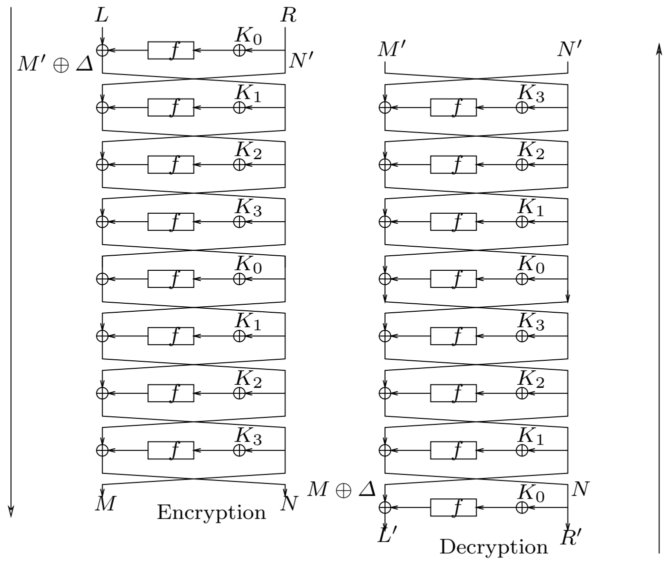

Fig. 4. Combining the complementation slide and sliding with a twist techniques in a single unified attack against a Feistel cipher with four-round self-similarity.

Now note that the odd rounds always line up, but the even rounds have the constant difference $K_1 \oplus K_3$ in the round subkeys. Therefore, we can apply the *complementation slide* technique, if we can get texts with a slid difference of $\langle 0, K_1 \oplus K_3 \rangle$. Then we get the attack shown in Figure 4.

Combining the two advanced sliding techniques provides a number of significant benefits. First, we obtain an n-bit filtering condition, so detecting slid pairs becomes easy. Consequently, the analysis phase is straightforward. Also, the combined approach makes it easier to recover key material from a slid pair. Finally, perhaps the most important improvement is that now we can reduce the data complexity of the attack to just $2^{n/4}$ texts, in the case where chosen-plaintext/ciphertext queries are allowed. Neither advanced sliding technique can—on its own—provide these advantages; in this respect, the whole is greater than the sum of the parts.

3.4 Attack on DES with Brown-Seberry Key-schedule

In [2] an alternative key-schedule for DES was proposed. This key-schedule was supposed to be "as effective as that used in the current DES" and was "suggested for use in any new algorithm" [2]. This variant of DES was already studied in [1] resulting in a related-key attack on it. In this section we show a chosen plaintext/ciphertext slide attack on this variant of DES, which uses only 128 chosen texts and negligible time for analysis. The attack works for 2^{40} out of 2^{56} keys.

To remind the reader: the DES key-schedule consists of two *permuted-choice* permutations PC1 and PC2, and a rotation schedule. The first permuted choice PC1 is used to reduce the key-size from 64 bits to 56 bits. Then the result is divided into two 28-bit registers C and D. Each round we cyclicly rotate both registers by one or two bits to the left. Permuted choice PC2 is applied to the result, which picks 24 bits from each 28-bit register and thus forms a 48-bit round subkey.

In [2] a key-schedule that rotates by 7 bits every round was proposed (instead of the irregular 1,2-bit rotations used in DES). Due to a larger rotation amount which spreads bits between different S-boxes the PC2 permutation was simplified to become an identity permutation which just discards the last 4 bits of each 28-bit register. We claim that for $1/2^{16}$ of the keys, this variant can be broken with our sliding with a twist techniques as follows: the known-plaintext attack will require $2^{32.5}$ texts, time and space; the chosen-plaintext/ciphertext, however, will require only 2^7 texts!

First of all notice that since the new rotation amount (7 bits) divides the size of the key-schedule registers (28 bits) the registers C, D return to their original state every four rounds. This results in a key-schedule with a period of four, which can be analyzed by the methods that we developed in the previous sections for the four-round self-similar Feistel ciphers. We will extend the standard attack even further by noticing that DES key-schedule is used and not four independent round subkeys as in our previous model. However, DES-like ciphers introduce one small complication: the DES round function XORs the subkey against the 48-bit expanded input rather than the raw 32-bit input, so the complementation slide only works if the 48-bit subkey difference is expressible as the expansion of some 32-bit text difference.

Let $J_i = \langle C \lll 7i, D \lll 7i \rangle$ so that $K_i = \mathrm{PC2}(J_i)$. For the sliding with a twist to work in the case of DES we need $K_1 \oplus K_3$ to have an 'expandable' form in order to pass through the 32 to 48 expansion of the DES round function. Note also that if $J_1 = \langle u, v, u', v' \rangle$ where u, v, u, v' are all 14-bit quantities, then $J_3 = \langle v, u, v', u' \rangle$ in a Brown-Seberry key-schedule, and thus for $Z = J_1 \oplus J_3$ we have $Z_i = Z_{i+14}$ for $i \in \{0, 1, \ldots, 13, 28, 29, \ldots, 41\}$. The PC2 just discards Z_i for $i \in \{24, 25, \ldots, 27, 52, 53, \ldots, 55\}$ to get the 48-bit quantity $Y = \mathrm{PC2}(Z) = K_1 \oplus K_3$.

If we insist $Y = \mathrm{Expansion}(X)$ for some X, we get 16 constraints on Y: namely, $Y_i = Y_{i+2}$ for $i = 6j + k$, $j \in \{0, \ldots, 7\}$, $k \in \{4, 5\}$ where subscripts are taken modulo 48. Thus we have

$$Z_i = Z_{i+2} \text{ for } i \in \{4, 5, 10, 11, 16, 17, 32, 33, 38, 39, 44, 45\};$$

and $Z_i = Z_{i+6}$ for $i \in \{22, 23, 50, 51\}$. Therefore $Y = K_1 \oplus K_3$ is expandable if and only if $Z = J_1 \oplus J_3$ has the form

$$Z = \langle abcdcd\ cdefgh\ ghabcd\ cdcdef\ ghgh\ \ efklkl\ klabmn\ mnefkl\ klklab\ mnmn \rangle$$

where $a, b, .., n$ are 12 arbitrary bits. we see that there are exactly 2^{12} expandable values of $K_1 \oplus K_3$ that satisfy the required constraints. Moreover, for each expandable value of $K_1 \oplus K_3$, there are 2^{28} possible values of J_1 for which $K_1 \oplus K_3$

has the given value (since we may choose u and u' arbitrarily, setting v and v' as required to ensure that $\langle u \oplus v, u \oplus v, u' \oplus v', u' \oplus v' \rangle$ has an appropriate value for $J_1 \oplus J_3$).

This shows that there are 2^{40} values of J_1 that lead to four-round self-similarity with an expandable value for $K_1 \oplus K_3$. In other words, $1/2^{16}$ of the keys are breakable with our standard attack. Note that the standard attack for the case of four independent round subkeys uses $2^{32.5}$ known texts, time and space, or 2^{17} chosen texts, time and space. However, we may use the special structure of $K_1 \oplus K_3$ to significantly reduce the complexity of the chosen-text attack.

In particular, we choose 2^6 plaintexts of the form $\langle L_i, R \rangle$ and 2^6 ciphertexts of the form $\langle M'_j, N' \rangle$, where $R = N'$ is fixed throughout the attack and

$$\begin{aligned} L_i &= \langle bcdc\ \ def0\ 0abc\ \ dcde\ \ f000\ 0ab0\ \ 0ef0\ \ 000a \rangle \\ M'_j &= \langle 0000\ 000g\ h000\ 0000\ 0klk\ \ l00m\ n00k\ \ lkl0 \rangle, \text{ so that} \\ L_i \oplus M'_j &= \langle bcdc\ \ defg\ habc\ dcde\ \ fklk\ \ labm\ nefk\ lkla \rangle \end{aligned}$$

and thus Expansion$(L_i \oplus M'_j) = K_1 \oplus K_3$ for some i, j, which immediately gives us a slid pair. (We assume for ease of description that the cipher includes the final swap and no IP or FP, so that Figure 4 in Section 3.2 applies.) We can recognize the slid pair by a 64-bit filtering condition on $\langle M, N \rangle, \langle L', R' \rangle$, and so the analysis phase is easy.

To sum up, this provides an attack on the cipher that breaks $1/2^{16}$ of the keys with 2^7 chosen texts, time and space.

3.5 Generalizations for a Composition of Stronger Functions

In Section 2 we have seen how a typical slide attack may work. However, in many cases this approach is too restrictive, since it may be desirable to analyze ciphers which decompose into a product of stronger functions; in particular, the round function may be strong enough that multiple input/output pairs are required to recover any key material. In this section we show several techniques to handle this situation.

One approach is to use a differential analysis. Denote by n the block size of the cipher. Suppose there is a non-trivial differential characteristic $\Delta X \rightarrow \Delta Y$ of probability p for the round function. We associate to each plaintext P the plaintext $P \oplus \Delta X$ and to each plaintext P' another plaintext $P' \oplus \Delta Y$. Then, if $P' = F(P)$, we will also have $P' \oplus \Delta Y = F(P \oplus \Delta X)$ with probability p (thanks to the characteristic $\Delta X \rightarrow \Delta Y$), which provides two slid pairs. In this way we may obtain four known input/output pairs for the function F. We can generate a set of $3 \cdot 2^{n/2} p^{-1/2}$ chosen plaintexts such that for plaintext P in the chosen set the plaintexts $P \oplus \Delta X$ and $P \oplus \Delta Y$ are also in the set; then we will expect to see one pair P, P' satisfying both the slide and the differential patterns.

The second approach (which is probably the simplest) works like this. Suppose to recover the key we need N known texts for the round function F. For each plaintext P, we suggest to get the encryption $E(P)$ of P, and the double-encryption $E^2(P) = E(E(P))$ of P, and so on, until we have obtained $E^{2N}(P)$. Then, if $P' = F(E^i(P))$, we find $2N - i$ slid pairs "for free" by the relation $E^j(P') = F(E^{j+i}(P))$ for $j = 1, .., 2N - i$. With $2^{(n+1)/2}N^{1/2}$ chosen texts, we expect to find about N slid pairs in this way (probably all in the same batch formed from a single coincidence of the form $P' = F(E^i(P))$). To locate the batch of slid pairs, one could naively try all 2^{n+2} possible pairings of texts (though in practice we would search for a more efficient approach); each pairing that gives N or more known texts for F will suggest a key value that can then be tested[2].

Normally this last attack would be classified as an adaptive chosen-plaintext attack. However, note that in many modes (CBC, CFB) it can be done with a non-adaptive chosen-plaintext attack. Furthermore, in the case of OFB mode, a known plaintext assumption suffices. However, these comments assume that re-encryption preserves the sliding property, which is not always the case.

Another possible generalization is in the case of Feistel-ciphers. In this case one can detect slid pairs even before trying to find the correct secret key k. In the case of a balanced Feistel cipher with block size n we have an $n/2$-bit condition on the ciphertexts of a slid pair. This increases the S/N ratio considerably, filtering out most of the incorrect pairs even before we start the analysis. This property allows an attacker to accumulate sufficient number of slid pairs before he starts an attack on a round-reduced variant of a cipher.

Notice also that if we use a technique for receiving many slid pairs in the case of a Feistel-cipher, we would need only $2 \cdot 2^{n/4}N$ chosen texts, and the S/N ratio will be excellent by comparing several halves of the ciphertexts.

Furthermore if $N^{1/2} > 2^{n/4}$, an absolutely different idea can be used. Choose a random starting point P. About $2^{n/2}$ times iterate the following operation $s \circ E$, where s denotes swap of the halves (the swap is needed only if E has no final swap at the last round). This way one can obtain more than $2^{n/2-\log r}$ slid pairs (here r denotes the number of rounds of a cipher). The S/N ratio is again excellent. The idea is that we essentially search for a symmetric point (A, A) of a round function, which happens after about $2^{n/2}$ rounds ($2^{n/2-\log r}$ encryptions). This does not necessarily happen in the middle of a cipher, so we may have to perform up to r times more encryptions before we reach a fixed point for E. In half of the cases (if the first symmetric point happened at an even round) we will receive an orbit "slidable" by two rounds, and in other half of the cases (symmetric point at odd rounds) an orbit will be "slidable" by one round. Even if an orbit is "slidable" only by two, and thus $n/2$-bit filtration will be unreachable to us, the encryption fixed point that ends our orbit helps us slide the orbit correctly (at most $r/2$ possibilities).

[2] If E were behaving like a random function, it would be enough to take $2^{n/2} + N$ encryptions, from an orbit of some arbitrarily chosen element P, but since E is expected to behave like a random permutation, an orbit of P will be a part of usually a very large cycle, leaving no place for collisions. Considering a few more orbits will not help either.

4 Cryptanalysis of DESX and Even-Mansour Schemes

DESX is an extension of DES proposed by Rivest in 1984. It makes DES more resistant to exhaustive search attacks by XORing two 64-bit keys: one at the input and another at the output of the DES encryption box[3]. See [10, 16] for theoretical analysis of DESX.

In this section we show the unexpected result that the DESX construction contains just enough symmetry to allow for slide attacks. These results are actually generally applicable to all uses of pre- and post-whitening (when applied using self-inverse operations like XOR), but for convenience of exposition we will focus on DESX.

The attacks presented here are another example of an application of the powerful new *sliding with a twist* technique. Our attacks on DESX are significantly better than the best previously known attacks: we need just $2^{32.5}$ *known* texts and $2^{87.5}$ time for the analysis, while the best generic attack reported in the literature is a *chosen*-plaintext attack with comparable complexity [10, 16][4]. Thus, sliding techniques allow one to move from the chosen-text attack model to the more realistic known-text attack model. Even more unexpectedly, our attack can also be converted to a ciphertext-only attack.

We briefly recall the definition of DESX. Let $E_k(x)$ denote the result of DES-encrypting the plaintext x under the key k. Then we define DESX encryption under the key $K = \langle k, k_x, k_y \rangle$ as $EX_K(p) = k_y \oplus E_k(p \oplus k_x)$. To set up the necessary slide relation, we imagine lining up a DESX encryption against a slid DESX decryption, as shown in Figure 5. More specifically, we say that the two known plaintext pairs $\langle p, c \rangle$ and $\langle p', c' \rangle$ form a slid pair if $c \oplus c' = k_y$. Consequently, for any slid pair, we will have

$$p' = k_x \oplus E_k^{-1}(c' \oplus k_y) = k_x \oplus E_k^{-1}(c)$$

as well as $p = k_x \oplus E_k^{-1}(c')$. Combining these two equations yields $k_x = p \oplus E_k^{-1}(c') = p' \oplus E_k^{-1}(c)$. As a result, we get a necessary property of slid pairs: they must satisfy

$$E_k^{-1}(c) \oplus p = E_k^{-1}(c') \oplus p'. \qquad (*)$$

To get a single slid pair, we obtain $2^{32.5}$ known plaintexts $\langle p_i, c_i \rangle$ and search for a pair which satisfies the sliding condition $(*)$. The pairs can be recognized efficiently with the following technique. We guess the DES key k. Next, we insert $E_k^{-1}(c_i) \oplus p_i$ into a lookup table for each i; alternatively, we may sort the texts by this value. A good slid pair $\langle p, c \rangle, \langle p', c' \rangle$ will show up as a collision in the table. Also, each candidate slid pair will suggest a value for k_x and k_y as above

[3] Note that an idea to use simple keyed transformations around a complex mixing transform goes back to Shannon [18, pp.713].

[4] One may apply differential or linear cryptanalysis to DESX, but then at least 2^{60}–2^{61} texts are needed [11]. In contrast, slide attacks allow for a *generic* attack with a much smaller data complexity.

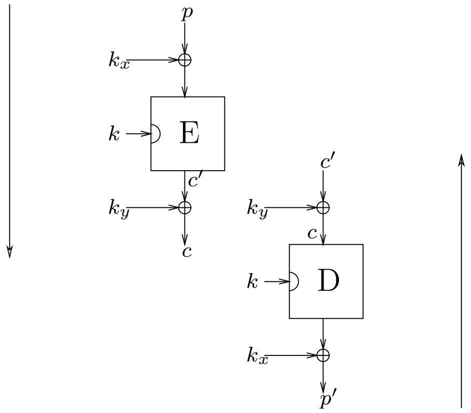

Fig. 5. Sliding with a twist, applied to DESX.

(e.g., $k_y = c \oplus c'$ and $k_x = p \oplus E_k^{-1}(c')$), so we try the suggested DESX key $\langle k, k_x, k_y \rangle$ immediately on a few known texts. With $2^{32.5}$ known texts, we expect to find one false match (which can be eliminated quickly) per guess at k, as well as one correct match (if our guess at k was correct). If this attack sketch is not clear, see the algorithmic description in Figure 6.

In total, the average complexity of our slide attack on DESX is $2^{87.5}$ offline trial DES encryptions, $2^{32.5}$ known texts, and $2^{32.5}$ space. The slide attack is easily parallelized. Compare this to the best attack previously reported in the open literature, which is a *chosen*-plaintext attack that needs 2^{121-m} time (average-case) when 2^m texts are available [10, 16]. Therefore, our attack converts the chosen-plaintext assumption to a much more reasonable known-plaintext assumption at no increase in the attack complexity.

Ciphertext-only attacks. Note that in many cases our slide attack on DESX can even be extended to a ciphertext-only attack. We suppose (for simplicity) that most plaintext blocks are composed of just the lowercase letters 'a' to 'z', encoded in ASCII, so that 24 bits of each plaintext are known[5]. For each i we calculate 24 bits of $E_k^{-1}(c_i) \oplus p_i$ and store the result in a lookup table. Due to the weak filtering condition, by the birthday paradox we expect to find about $2^{2 \cdot 32.5-1}/2^{24} = 2^{40}$ collisions in the table. Each collision suggests a value for k_y (as $k_y = c \oplus c'$) and for 24 bits of k_x, which we immediately try with a few DESX trial decryptions on other known ciphertexts. Therefore, for each guess of k the workfactor is 2^{40} DES operations.

This provides a simple ciphertext-only attack needing about $2^{32.5}$ ciphertexts and 2^{95} offline DES operations. The work-factor can be reduced somewhat to 2^{95} simple steps (where each step is much faster than a trial decryption), if 2^{33}

[5] The attack degrades gracefully if our model of the plaintext source is only probabilistic: for instance, if half of the texts follow the model, the attack will need only $\sqrt{2}$ times as many ciphertexts and only twice as much work.

```
ATTACK:
1. Collect $2^{32.5}$ known plaintexts $\langle p_i, c_i \rangle$.
2. For each $k \in \{0,1\}^{56}$, do
3.   Insert $\langle E_k^{-1}(c_i) \oplus p_i, i \rangle$ into a hash table keyed by the first component.
4.   For each $i \neq j$ with $E_k^{-1}(c_i) \oplus p_i = E_k^{-1}(c_j) \oplus p_j$, do
5.     Set $k_y = c_i \oplus c'_j$ and $k_x = p_i \oplus E_k^{-1}(c_i \oplus k_y)$.
6.     Test the validity of the guessed key $\langle k, k_x, k_y \rangle$ on a few more known texts.
```

Fig. 6. The DESX slide attack, in full detail. It is clear that—once discovered—the attack may be described without reference to sliding, but the *sliding with a twist* methodology made it possible to find the attack in the first place.

known ciphertexts are available, by considering candidate slid pairs two at a time and filtering on the suggested value of k_y, since then the correct value of k_y will be suggested at least twice and can therefore be recognized in this way before doing any trial decryptions. Note that these ciphertext-only attacks are applicable not only to ECB mode but also to most of the standard chaining modes, including CBC and CFB modes.

CRYPTANALYSIS OF THE EVEN-MANSOUR SCHEME. In [7], Even and Mansour studied a simple n-bit block cipher construction based on a fixed pseudo-random permutation and keyed n-bit XORs at the input and at the output. Due to the generic nature of our previous attack on DESX it can also be used to analyze the Even-Mansour construction[6]. In the case of Even-Mansour we replace E_k with an unkeyed mixing transformation E on n-bit blocks, so our slide attack succeeds with just $2^{(n+1)/2}$ known plaintexts and $2^{(n+1)/2}$ work. This provides a known-plaintext attack with the same complexities as the best previously-known chosen plaintext attack [6] and within a factor of $\sqrt{2}$ away from the Even-Mansour lower bound.

5 Analysis of GOST

GOST, the Russian encryption standard [19], was published in 1989.[7] Even after considerable amount of time and effort, no progress in cryptanalysis of the standard was made in the open literature except for a brief overview of a GOST structure in [4] and a related key attack in [9]. In this section we apply slide techniques to GOST and thus are able to produce cryptanalytic results that shed some light on its internal structure.

The GOST encryption algorithm is a block cipher with 256-bit keys and a 64-bit block length. GOST is designed as a 32-round Feistel network, with 32-bit round subkeys. See Figure 7 for a picture of one round of GOST.

[6] Of course, these attacks will apply with the same complexity to DESX when the DES key k is known somehow.

[7] It was translated into English in 1993 and since then became well known to open cryptographic community.

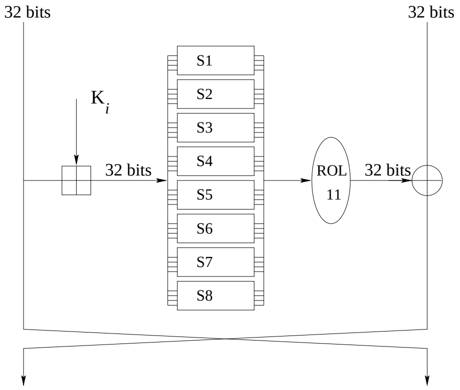

Fig. 7. One round of a GOST cipher.

The key schedule divides the 256-bit key into eight 32-bit words $K_0, \ldots, K_7$, and then uses those key words in the order $K_0, \ldots, K_7$, $K_0, \ldots, K_7$, $K_0, \ldots, K_7$, $K_7, K_6, \ldots, K_0$. Notice the 'twist' in the last 8 rounds.

The Analysis of GOST. GOST looks like a cipher that can be made both arbitrarily strong or arbitrarily weak depending on the designer's intent since some crucial parts of the algorithm are left unspecified. A huge number of rounds (32) and a well studied Feistel construction combined with Shannon's substitution-permutation sequence provide a solid basis for GOST's security. However, as in DES everything depends on the exact choice of the S-boxes and the key-schedule. This is where GOST conceptually differs from DES: the S-boxes are not specified in the standard and are left as a secondary key common to a "network of computers"[8].

The second mystery of GOST is its key-schedule. It is very simple and periodic with the period of eight rounds except for the last eight rounds where a twist happens. It is intriguing to find a reason for the twist in the last eight rounds of the key schedule. Moreover, in many applications we may wish to use shorter 64- or 128-bit keys, yet it is not clear how to extend these to a full 256-bit GOST key securely (fill the rest with zeros, copy the bits till they cover 256 bits, copy bits in a reversed order?).

Why the Twist? Consider a GOST cipher with a homogeneous key schedule, i.e., omitting the final twist (let us denote it GOST-H). Is this cipher less secure than GOST? We argue that, if one takes into account the slide attacks, it is. GOST-H can be decomposed into four identical transforms, each consisting of eight rounds of GOST. Furthermore, if one assumes that the round subkey is xored instead of being added, the cipher will have 2^{128} weak keys of the form $\langle A, B, C, D, A, B, C, D \rangle$ (here each letter represents a 32-bit GOST subkey). These keys are weak since they allow for a *sliding with a twist* attack.

[8] Contrary to common belief, the standard does not even require the S-boxes to be permutations.

There is a known plaintext attack with 2^{32} texts and time, and a chosen plaintext attack with 2^{16} texts and time; see Section 3.3 for more details.

Notice that the 2^{128} keys of the form $\langle A, B, C, D, D, C, B, A\rangle$ are also weak since GOST-H with these keys is an involution and thus double encryption will reveal the plaintext. Since these keys are invariant under a twist the same property holds for GOST itself. Also, there are 2^{32} fixed points for each key of this form, which demonstrates that there may be problems with using GOST to build a secure hash function.

The Attack on 20 rounds of GOST⊕. Suppose again that the round subkey is xored instead of being added, (we will denote this variant of GOST as GOST⊕). Here we show an application of *sliding with a twist* which results in an attack on the last 20 rounds of GOST⊕.

Applying *sliding with a twist*, we get a picture that looks like this:

$$\begin{array}{l} K_4\ K_5\ K_6\ K_7\ K_0\ K_1\ K_2\ K_3\ K_4\ K_5\ K_6\ K_7\ K_7\ K_6\ K_5\ K_4\ K_3\ K_2\ K_1\ K_0 \\ K_0\ K_1\ K_2\ K_3\ K_4\ K_5\ K_6\ K_7\ K_7\ K_6\ K_5\ K_4\ K_3\ K_2\ K_1\ K_0\ K_7\ K_6\ K_5\ K_4. \end{array}$$

Let F denote 4 rounds of GOST⊕ with key $K_4, \ldots, K_7$. With a pool of 2^{33} known texts, we expect to find two slid pairs, and each slid pair gives two input/output pairs for F. Breaking F with two known texts is straightforward, and can be performed in time comparable to about 2^9 evaluations of 4-round GOST (equivalent to 2^5 20-round trial encryptions). Thus in our attack we examine all 2^{65} text pairs; each pair suggests a value for 128 bits of key material, which we store in a hash table (or sorted list). The right key will be suggested twice, so we expect to be able to recognize it easily. By the birthday paradox, there will be only about two false matches, and they can be eliminated in the next phase.

Once we have recovered $K_4, \ldots, K_7$, it is easy to learn the rest of the key in a second analysis phase. For example, we can peel off the first four rounds and look for fixed points in the same pool of texts. Since the round subkeys are palindromic in the last sixteen rounds of GOST, there are 2^{32} fixed points, and each has the value $\langle x, x\rangle$ before the last eight rounds of encryption. Thus, given a fixed point, we can try the 2^{32} values of $\langle x, x\rangle$, encrypt forward and backward eight rounds, and obtain two candidate input/output pairs for 4 rounds of GOST⊕ with key $K_0, \ldots, K_3$, so that a value for $K_0, \ldots, K_3$ is suggested after 2^5 work; then the suggested 256-bit key value is tried on another known text pair.

In all, this gives an attack on the last 20 rounds of GOST⊕ that needs 2^{33} known texts, 2^{70} work, and 2^{65} space to recover the entire 256-bit key. Note that this attack is generic and works for any set of (known) S-boxes. The large memory requirements make the attack highly impractical, but we view it as a first step towards a better understanding of the GOST design.

6 Related Work

The first step in the "sliding" direction can be dated back to a 1978 paper by Grossman and Tuckerman [8], which has shown how to break a weakened Feistel cipher[9] by a chosen plaintext attack, independent of the number of rounds. We were also inspired by Biham's work on related-key cryptanalysis [1], and Knudsen's early work [12].

Some related concepts can be found in Coppersmith's analysis of fixed points in DES weak keys and cycle structure of DES using these keys [5]. This analysis was continued further by Moore and Simmons [14]. For a DES weak key, all round subkeys are constant, and so encryption is self-inverse and fixed points are relatively common: there are precisely 2^{32} fixed points. Note that this property will also be found in any Feistel cipher with *palindromic* round key sequences, so the slide attack is not the only weakness of ciphers with self-similar round subkey sequences.

7 Discussion

In this section we discuss possible extensions of slide attacks presented in this paper and possible directions of future research.

The most obvious type of slide attack is usually easy to prevent by destroying self-similarity in iterative ciphers, for example by adding iteration counters or fixed random constants. However more sophisticated variants of this technique are harder to analyze and to defend against. This paper is a first step towards advanced slide attacks which can penetrate more complex cipher designs.

One promising new direction is the *differential slide attack*. By sliding two encryptions against each other, we obtain new differential relations which in some cases are not available in the conventional differential analysis of a cipher. These might be very powerful, since they might for example violate the subtle design constraints placed on the system by its designer and thus result in unexpected differential properties. If key-scheduling is not self-similar or symmetric, differences in subkeys can cause constant XOR values to be introduced in the middle of the encryption process when slid pairs are considered. (In many cases, one can slide by different numbers of rounds and thus control the differences to some extent.) The drawback of this method is the same as in conventional methods: its complexity increases fast with the number of rounds, contrary to the general sliding technique, which works for arbitrary number of rounds.

Acknowledgments

We would like to thank Eli Biham for pointing to us the Brown-Seberry variant of DES key-schedule.

References

1. E. Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, J. of Cryptology, Vol.7, pp.229–246, 1994.

[9] They analyzed an 8-round Feistel cipher with eight bits of key material per round used to swap between two S-boxes S_0 and S_1 in a Lucifer-like manner: a really weak cipher by modern criteria.

2. L. Brown, J. Seberry, *Key Scheduling in DES Type Cryptosystems*, proceedings of AUSCRYPT'90, LNCS 453, pp.221–228, Springer Verlag, 1990
3. A. Biryukov, D. Wagner, *Slide Attacks*, proceedings of FSE'99, LNCS 1636, pp.245–259, Springer Verlag, 1999.
4. C. Charnes, L. O'Connor, J. Pieprzyk, R. Safavi-Naini, Y. Zheng, *Comments on Soviet Encryption Algorithm*, proceedings of EUROCRYPT'94, LNCS 950, pp.433–438, Springer Verlag, 1994.
5. D. Coppersmith, *The Real Reason for Rivest's Phenomenon*, proceedings of CRYPTO'85, pp.535–536, Springer Verlag, 1986.
6. J. Daemen, *Limitations of the Even-Mansour Construction*, proceedings of ASIACRYPT'91, pp.495–498, Springer-Verlag 1992.
7. S. Even, Y. Mansour, *A Construction of a Cipher from a Single Pseudorandom Permutation*, Journal of Cryptology, Vol.10, No.3, pp.151–161, 1997.
8. E. K. Grossman, B. Tucherman, *Analysis of a Weakened Feistel-like Cipher*, 1978 International Conference on Communications, pp.46.3.1–46.3.5, Alger Press Limited, 1978.
9. J. Kelsey, B. Schneier, D. Wagner, *Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, proceedings of CRYPTO'96, pp.237–251, Springer Verlag, 1996.
10. J. Kilian, P. Rogaway, *How to Protect Against Exhaustive Key Search*, proceedings of CRYPTO'96, LNCS 1109, pp.252–267, Springer Verlag, 1996.
11. B. Kaliski, M. Robshaw, *Multiple encryption: weighing security and performance*, Dr. Dobb's Journal, pp.123–127, Jan. 1996.
12. L. R. Knudsen, *Cryptanalysis of LOKI91*, proceedings of AUSCRYPT'92, LNCS 718, pp.196–208, Springer Verlag, 1993.
13. M. Luby, C. Rackoff, *How to Construct Pseudorandom Permutations from Pseudorandom Functions*, SIAM Journal of Computing, Vol. 17, pp.373–386, 1988.
14. J. H. Moore, G. J. Simmons, *Cycle Structure of the DES with Weak and Semi-Weak Keys*, proceedings of CRYPTO'86, pp.9–32, Springer Verlag, 1987.
15. B. Preneel, V. Rijmen, A. Bosselears, *Principles and Performance of Cryptographic Algorithms*, Dr. Dobb's Journal, Vol. 23, No. 12, pp.126–131, Miller Freeman, Dec. 1998.
16. P. Rogaway, *The Security of DESX*, RSA Laboratories' CryptoBytes, Summer 1996.
17. B. Schneier, *Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)*, proceedings of FSE'94, LNCS 809, pp.191–204, Springer Verlag, 1994.
18. C. Shannon, *Communication Theory of Secrecy Systems*, Bell Sys. Tech. J., Vol. 28, pp. 656–715, October 1949. (A declassified report from 1945.)
19. I. A. Zabotin, G. P. Glazkov, V. B. Isaeva, *Cryptographic Protection for Information Processing Systems. Cryptographic Transformation Algorithm*, Government Standard of the USSR, GOST 28147-89, 1989. (Translated by A. Malchik, with editorial and typographic assistance of W. Diffie.)