Slide Attacks with A Known-plaintext Cryptanalysis

Soichi FURUYA

Systems Development Lab., Hitachi, Ltd.
soichi@sdl.hitachi.co.jp

Abstract. Although many strong cryptanalytic tools exploit weaknesses in the data-randomizinig part of a block cipher, relatively few general tools for cryptanalyzing on the other part, the key scheduling part, are known. A slide attack is an instance of attacks exploiting the key-schedule weakness. In this paper, currently proposed slide attacks can be still enhanced so that all currently published known-plaintext analytic technique can be applied to smaller part of a cipher with a weak key-scheduling part. As an example, we demonstrate applications of a slide attack to linear cryptanalysis, a DES variant case. In addition, we also show that our enhancement enables to declassify the unknown primitive used in a block cipher. We test a block cipher, GOST, and show how to de-classify the hidden 4-bit substitution tables.

1 Introduction

Many cryptanalyses of a block cipher are based on careful and elaborate observations on the data-randomizing part in a block cipher, and ignore the structure of corresponding key schedule. Since the differential cryptanalysis was presented by Biham and Shamir [1,2], the statistical aspects of a cryptosystem, typically the data-randomizing part, have been studied to cheek whether or not the cipher is secure against these attacks. Linear cryptanalysis [14] and other related works [9,10,12, 19] adopt this approach, however with other statistical characteristics.

In those attacks, round keys, i.e. the outputs of a key-scheduling part, are concerned and very little analyses on a key schedule are used to improve attacks. A related-key analyses[11] is one of major intersections between analyses on data-randomizing part and ones on key-scheduling part. However, the basic idea of the related-key analysis is particularly specialized not in general cryptanalytic techniques, but only in differential cryptanalysis.

A slide attack [3] is an attack based on a particular key-schedule weakness; if a key schedule has inherent cyclicity of round keys, there exist two distinct known plaintext pairs such that all intermediate values are conceptually identical but appears in different rounds. More precisely the coinciding intermediate values are *sliding* by one round. Biryukov and Wagner presented some novel improvements in

[4], e.g., sliding attacks in the adoptive chosen plaintext-ciphertext environments. In those attacks, the attacker basically exploits the weaknesses substantially in the key schedule.

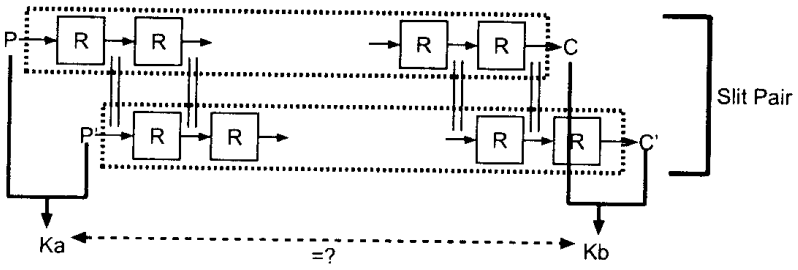

Fig. 1. Finding slit pair and deriving round key.

In this paper. we show that the slide attack can be used to enhance some types of current attacks of a cipher with a weak key—scheduling part. In fact Ins enhancement is applicable to any known—plaintext cryptanalytic tools, e.g. linear and partitioning cryptanalysis. In other words, our attack exploits time combination of the weakness in the key schedule and the Data—randomizing part.

More precisely, our extension allows the attack to be applicable not only to the target cipher with a *weak* round function but also a cipher with a sequence of rounds vulnerable to known-plaintext attacks. Note that the *weak* round function can be only one or two rounds of Feistel structure, whereas a sequence of round function allows four or more it iterations.

We apply this Technique to some block—cipher variants in order to demonstrate the effectiveness of the proposed enhancement. We treat the DES variant with four —round iterated round keys and describe its cryptanalysis that, is the combination of the linear cryptanalysis and the slide attack.

We also study the GOST block-cipher, which has both the simple key-scheduling algorithm the round function with confidential S boxes. Combining investigations on both parts, we point out interesting technique to de-classify the hidden S boxes.

This paper consists of the following sections. Section two prepares the preliminaries for representations and notations. Section three describes the original ideas of slide attacks. Section four proposes an extension to the slide attack. In section five, the extension is applied to a couple of example ciphers. In section six, we concludes our works.

2 Slide attacks

Let us assume that $K_i, (1 \le i \le r)$ are identical and the round function R has a structure such that the round key can be efficiently calculated out of the pair of

the input and the output.

The attacker tries to find a special pair of plaintexts, hereafter we call it a *slit* pair. (P, P') such that one (for example P') is the output of the first round function of the other's (P) encryption. Once he succeeds in finding the right pair, then he can efficiently calculate the round key out of two pairs of the input and the output for a round function.

To find the concerning plaintext pair, the naive way is to collect arbitrary $2^{b/2}$ known-plaintexts, where b is the block length in bits. Thanks to the birthday paradox, there exists such a slit pair with high probability.

Given the sufficient number of known-plaintexts, he identifies whether or not each pair in collected plaintexts (P_a,P_b) is the slit pair. The validation is to check the identity of two derived round keys, namely K_P (calculated out of P_a and P_b) and K_C (calculated out of C_a and C_b), where C_a and C_b are corresponding ciphertexts of P_a and P_b. If $K_P = K_C$, the pair (P_a,P_b) can be a slit pair. Otherwise they cannot be so and he tries the next plaintext pair.

This naive way to find a slit pair can be improved more effectively, depending on the structure or the characteristics of the round function. For instance, if the round function R is a Feistel structure, then a slid pair must share the same half of data, namely upper half of C_a and lower half of C_b. In this case the number of collected plaintexts are reduced to $2^{b/4}$.

The total complexity consists of two computations: collecting the sufficient number of known-plaintexts and searching the slit pair. Let n be the number of known-plaintexts. The former takes n computations and the latter takes $T \times n(n-1)/2$, where T is the computation for the slit-pair verification.

To calculate n the number of sufficient known-plaintexts, we take the birthday paradox into account. According to the rough estimation, about 50 % of the successful attack can be achieved by using about $2^{b/2}$ known-plaintexts. To obtain higher probability, say 80%, the increase of required known plaintext is no more than a factor of 2 to 4. An example calculation of probability that a birthday collides is depicted in Fig.2.

This basic idea is easily extended to plural rounds iteration. The original paper demonstrated the extended idea applied to their DES variants, 2K-DES, where K_1 is used in odd rounds, and K_2 in even rounds. In the slide attack of 2K-DES, two-round sliding property is used. More-round sliding has not been in open publications.

Remarks on Weak Round Function: In the original slide attack, Biryukov and Wagner introduced the concept of the *weak* round function with respect to the slide attacks. In order to apply the original slide attack, a sliding gap must be so *weak* that

sufficient key information is derived out of one (or two) input-output pairs of a round function. If the gap consists of more than two rounds, generally it becomes much more difficult to derive the key information.

3 Enhanced attacks with a known-plaintext attack

3.1 Our enhancement

In this section, we enhance slide attacks so that they are applicable not only ciphers with a *weak* round function but also ones with a sequence of a round function that is vulnerable to a known-plaintext cryptanalysis.

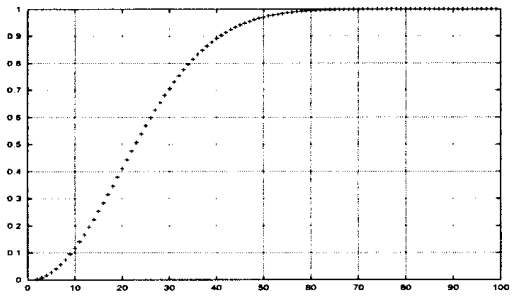

Fig. 2. Collision probability in choosing n out of 365.

At first we introduce the enhanced slide attack. In this context, the term "round" does not necessarily specify the exact definition of the round function of the block cipher. Instead, we intend to use the term "Round" to mean the unit of sliding, i.e., the gap. Typically the "Round" also includes the sequence of the round function.

A target cipher which we concern in this part consists of a sequence of Rounds. In addition, each Round is identically keyed, or equivalently the round-key generated in a cyclic manner. There also exist $D_R(t_d, q_d)$ and $A_R(t_a, q_a)$ that are the distinguisher and the key-deriving attack against the Round with q_d (or q_a) known-plaintexts and t_d (or t_a) computational time.

Because of the structure of the cipher, a slit pair exists as well as the original slide attack. At first, the attacker tries to find the slit pair by a Round. Instead of collecting a number of known-plaintexts, the attacker generates a number of (arbitrary) plaintexts and asks for $q_a - 1$ ciphertexts of multiple encryptions for each plaintext, i.e., the ciphertext, the ciphertext of the double encryption, triple-encryption and so on. If the pair of plaintexts (P_a, P_b) is a *slit* pair, then so is the corresponding pair of ciphertexts $(C_a^{(1)}, C_b^{(1)})$. Similarly the pair of the ciphertexts after n times encryption $(C_a^{(q_a-1)}, C_b^{(q_a-1)})$ also keeps the sliding

property (Fig.3). Note that each pair of ciphertexts $(C_a^{(k)}, C_b^{(k)})$ is also thought of as the pair of the input and the output of the Round. Therefore the attacker can obtain the q_a pairs (including plaintext pair) of the input-output pairs of the Round function, with which the attacker can mount the known-plaintext attack A_R.

To calculate the number of plaintexts, we apply the birthday paradox again. Consequently the naive method requires $2^{b/2}$ plaintexts to find a slit pair. An attacker use the distinguisher D_R to find a slit pair. For each possible pair of the collected plaintexts, the attacker invokes D_R. If D_R returns yes, then the attacker treats the pair as the slit pair. Otherwise he discards the pair and try the next pair. To invoke D_R the attacker has to prepare q_d pairs of the input output pairs of the Round. Hence the attacker prepares $q_d - 1$ ciphertexts for each plaintext.

We estimate the computational complexity for this attack. The attacker prepares $2^{b/2}$ plaintexts each of which is encrypted for $\max(q_d, q_a)$ times. In total, it takes $\max(q_d, q_a) \times 2^{b/2}$ computational time for generating data. The attacker makes $2^{b/2}(2^{b/2}-1)/2$ times D invocations, and a A invocation. In total, the computational complexity is estimated to be $\max(q_d, q_a) \times 2^{b/2} + 2^{b/2}(2^{b/2}-1)t_d/2 + t_a$.

In finding a slit pair, we roust exploit a certain weakness of a cipher element iterated in the target cipher. Using this weakness, an attacker can distinguish a slit pair (with the weak property) and others (expectedly which holds a random property). As an example, an attacker targeting four-round iterated Feistel cipher can be interested in a probabilistic linear property, with which he test pair-wise multiple ciphertexts to check if the probabilistic property of four rounds is detected. Practically, the technical way to find a slit pair is likely to relate to the way to attack the iterated cipher element. In the following, we demonstrate the typical analysis of our proposed enhancement.

3.2 Four-round iteration of DES

We consider a cipher holding four-round sliding property. An element of four-round iteration is not *weak* in the sense of original slide attacks by Biryukov and Wagner. Because of this reason the original slide attack cannot be applied to this cipher. Alternatively, they also proposed some advanced slide attacks where an attacker makes both encryption and decryption queries~4]. Their approach was to find a slit pair consisting of one plaintext-ciphertext pair and one ciphertext plaintext pair.

In this section, we apply our enhancement to the DES-cipher treated in[4] which is intensionally weakened to hold four-round subkey iteration, i.e. a DES cipher with four-round sliding property. We define the model to attack. However our enhancement achieves to attack the DES variant *without* decryption query, namely

adoptive chosen-plaintext attack in ECB, CBC. CFB or known-plaintext attack in GEB mode.

Let us demonstrate a simple example of a variant DES, ikDES4[1)], with a simple key schedule. In ikDES4, the key schedule works as follows. The secret key K whose length is $4\times48=192$, is divided into four 48-bit strings, K_t, (t = 1,2,3,4). Subkeys are set as follows:

$$K_{4i-3} = K_1 \text{ for } 1 \le i \le 8$$
$$K_{4i-2} = K_2 \text{ for } 1 \le i \le 8$$
$$K_{4i-1} = K_3 \text{ for } 1 \le i \le 8$$
$$K_{4i} = K_4 \text{ for } 1 \le i \le 8$$

The data-randomizing part is identical to DES cipher[6] except for the number of rounds and lack of initial and final bitwise permutations.

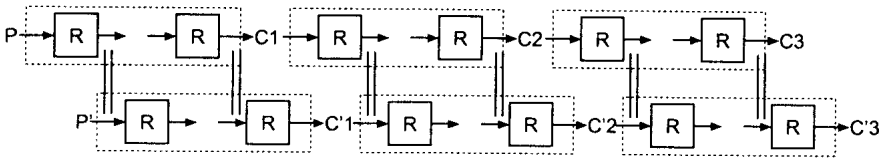

Fig. 3. A slide attack on multiple encryption.

First of all, a cryptanalytic tool exploited in the attack is introduced. This is the probabilistic linear relations of four rounds variant of DES data-randomizing part. This is approximated with $p_{DES4}=1/22-1.95\times2^{-5}$, which is detectable with $C\times(p_{DES4}^{(-2)})$. C depends on out of how many candidates the slit pair is detected[13]. Since in any case of slide attacks (and its variants), a statistical characteristics of the correct slit pair must be identified out of a large number of incorrect pairs, C must be large enough. We, in this paper, assume that $C=16$ gives enough to recognize out of less than 2^{64} corresponding random events.

It's easy to see that four rounds' slit pair, (P, C) and (P', C'), guarantees that each of those two pairs, (P, P') and (C, C') is a pair of input-output pair of DES-four rounds. Remember that in the original slide attack, an attacker can know no more than two input-output pairs even after he detects a slit pair. Consequently he cannot decide any information more than having about 2^{192-64} candidates, according to information theory. Although the result of this attack must be helpful to degrade the work effort of exhaustive search, this way of attacking does not use the full effect caused by iteration of round keys.

---

1) ikDESn stands for Iteratively-Keyed DES of v rounds. We do not care the number of rounds in a whole cipher as long as it is multiple of n.

Now, an attacker tries to attack the cipher, gathering ciphertexts in the multiple encryption of the target cipher. It's easy to see that ciphers in multiple encryption keeps to be slit if the pair of plaintexts does. In this case, an attacker continues to gather double encryption ciphertext, triple encryption, and so on until he gathers sufficient mount of pairwise data. If a pair of plaintext, (P_A, P_B), is a slit pair, the ciphers in single encryption, (C_A^1, C_B^1), the ciphers in double encryption, (C_A^2, C_B^2) and the ciphers in N times encryption, (C_A^n, C_B^n), each of which is a pair of input-output pair of four rounds. N must be the number for necessary plaintext pairs for linear cryptanalysis on four rounds DES, so that he exploits linear cryptanalysis on four round DES, stripping one or two rounds out of four applying maximum-likelihood-method on subkeys. Then the number of required input—output pair after finding a slit pair is

$$C(p'_{DES2})^{-2} = 8 \times (-20/64)^{-2} = 81.9,$$

for stripping two rounds. Note that in this analysis, we use $C = 8.0$ instead of 16, for a correct key in 2^{12} candidates in $S5$ box in round one, $S1$ box in round four. In successful attack, another parity of key bits in round three will be known to the attacker. Thirteen bits in total can be found in an attack (Fig. 4).

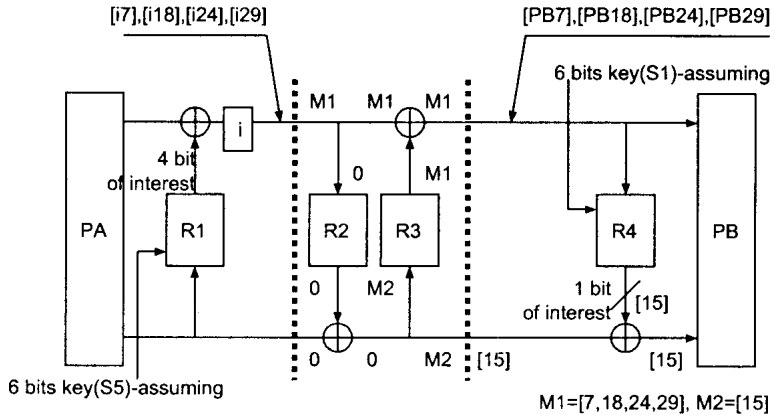

Fig. 4. Attack strategy in ikDES4.

We omitted explanation of how to distinguish a slit pair from pools of pairs. In this case, he can use four round linear characteristic for each slit pair, (P_A, P_B), (C_A^1, C_B^1), (C_A^2, C_B^2) and so on, and check correctness of a testing pair. If all the pairs above are correctly input-output pair of DES four rounds, they must show explicit bias in statistics, whereas if they not, they will not. In this case, meeting the most bias with the correct pair requires

$$C \times (p'_{DES4})^{-2} \approx 4294.96,$$

$$p'_{DES4} = 2^2 \times (1/2 - 12/64)^2(1/2 - 22/64).$$

This is the number of multiple encryptions enough to recognize the slit pair.
In terms of computational complexity in this attack, an attacker expects one slit pair in 2^{32} known-plaintexts pool. For each plaintext, he gathers 4294 ciphertext blocks, each of which is a resultant ciphertext of i-times multiple encryption for $1 \le i \le 4294$, storing a bit information of parity masked according to linear approximation of four rounds DES. In this first stage of an attack, it takes $4294 \times 2^{32} \approx 2^{44}$ encryptions to gather data and 2^{44} times masked parity calculation. In order to save whole encryption results, about $2^{44} \times 64$-bit memory space is required. Nevertheless, 2^{44}-bit space is enough if an attacker can apply chosen plaintexts queries after finding a slit pair.
In the next stage, he tries to find a slit pair, checking bias in distribution of parity bit. For each pair out of about 2^{63} possible pairs, 4295-bit exclusive-or and bit increment for a counter. In 32-hit processor, 4295-bit exclusive-or operation will take 4295/32 clocks, since storing 4295 bits in $|4295/32|$ is possible during the first stage. $2^{63} \times |4295/32| \approx 2^{70}$ exclusive-or operations approximately correspond to 2^{62} encryption, since a DES encryption takes $45 \times 8 = 360$ cycles on a Pentium processor[18]. As for the memory space, this stage requires negligible memory space since all he needs are the maximum bias and its pair information.
In the third stage, deriving 13-bit key information takes small amount of time, like $2^{13} \times 82$ counter increments, in comparison with those on above two stages. In this stage, 2^{12} counters are required.
In total, work efforts equivalent to 2^{62} encryptions with 2^{44} chosen-plaintexts, enable to crack ikDES4, which is independent of the number of rounds as well as original slide attacks. The minimum memory requirement is about 2^{44} bit, i.e. 2000 GByte.
We summarize the results on It's dependent on the applying known-plaintext attack. It must be very light calculation so that the mount of checking is less than the work effort of key exhaustive search.

4 Key-schedule analyses on block ciphers

We describe our observations on key schedules in a couple of block ciphers, discussing applicability of our attack and effectiveness. The necessary conditions to apply our attack are very simple: a vulnerable cipher should have a sliding property

(but not only ones with weak round functions); a cipher with a sliding property must be structured by a number of Round iterations and its non- negligible key space generate cyclic round keys synchronizing to iteration of round function. Most of the currently proposed block ciphers iterate identical round functions. Then our major observation begins with the structure of key schedule.

4.1 GOST

COST is a 64-bit block cipher proposed from the former Soviet Union [7,17], keyed with 256-bit secret key and equips eight secret *S*-boxes. However, an example of *S*-boxes for COST is disclosed and actually used in some applications.

We initially show a brief description of COST cipher. A plaintext block, P is divided into two 32-bit words, L_0 and R_0, and iterates a round function $R(L_i, R_i, K_i)$ for 32 times ($1 \le i \le 32$), where K_i is expanded keys generated by very simple key schedule. The ciphertext, C, is $L_{32}PVERR_{32}$. The round function, R, is very simple. The input data, R, is added with the key, K_i. The result, $R_{i-1} + K_i$, is divided into eight four-bit data, each of which becomes a input of one of eight *S*-boxes. The results of *S*-boxes are concatenated to make a 32-bit word. Then eleven-bit left rotation is executed on the result and exclusive-ored with L_i, which generates the output for R_i. The other output, L_i, is R_{i-1}. In terms of key schedule of GOST, it adopts very simple one. A 256-bit key is divided into eight 32-bit words, $S_1, \ldots, S_8$. The round key, K_i is decided as follows:

$$K_i = S_{i \bmod 8} \text{ for } 1 \le i \le 24$$
$$K_i = S_{33-i} \quad \text{for } 25 \le i \le 32 \text{ v}$$

Due to its key schedule, a slide attack chooses secret keys to hold the sliding property.

In this attack, the key to be attacked must he very particular. All the words in a key, K_i, $1 \le i \le 32$, are identical. In total, 2^{32} keys are vulnerable against our attack. Now all the 32 round functions arc keyed with an identical key, so that the original slide attack is applicable to check the slit pair, that just sees identity of two halves of data of ciphertext pairs.

In this case, the original slide attack is applicable if the S boxes are known. More interestingly, our attack allows user of GOST with unknown S boxes to de-classify his cipher. The same approach has already been described in [16]. We briefly revisit the result.

Saarinen's Algorithm

1. Set K to be zero vector (namely all subkeys are zero, too),

2. Encrypt a zero-vector plaintext (0,0) and find (z,0) formatted sliding plaintext as the original slide attack, where (X, Y) is denoted the left and right halves of plaintext or ciphertext data. Let a to be the common half data of both ciphertexts, namely a right half of zero plaintext's and a left half of sliding plaintext's.

3. The S box disclosure consists of $2^4 \times 2^4$ queries for each possible input V and output U, whether or not V is the output of the input U. Each query determines (a, b) values and the answer of queries should be given by sliding ciphertexts of $(a, 0)$ and (b, a) as plaintext. Repeating these queries for all eight S boxes, 2^{11} queries are required to disclose whole contents of hidden S boxes.

As Saarinen mentions, a and b are defined by interested query of V and U. However it is very unlikely to hold *sliding property* for those two plaintexts $(a, 0)$ and (b, a) even if the attacker knows z such that $z = f(0)$, where f is the zero-keyed F function of GOST. Consequently we claim that the Saarinen's algorithm, which still finds some elements of S box, lacks of flexibility to complete whole S box entries.

We introduce our enhancement to add the flexibility to the Saarinen's attack in order to fulfill the objective. Set one of 2^{32} vulnerable keys against one round sliding attack, and find a slit pair in the original way. For the next step, the attacker calculates n times multiple encryption each of plaintext of a slit pair. In each time of encryption, two ciphertexts are saved. With sufficient number of input-output pairs, all the elements in each S box are easily calculated since he knows the round key.

We consider the sufficient number of t in order to know all sixteen elements. We get the equation of the probability of choosing all sixteen elements after t times picking:

$$p_t = \sum_{i=0}^{16} (-1)^i \binom{i}{16} \left(1 - \frac{i}{16}\right)^t .$$

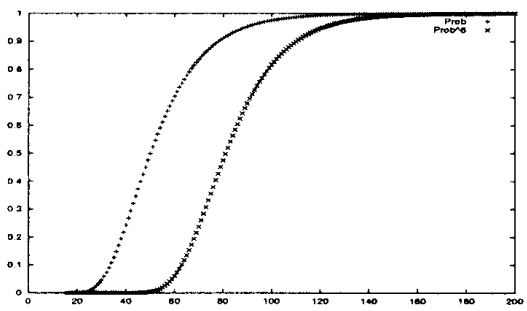

Fig. 5. Probabilities to choose all 16 elements.

In order to know all the elements in all eight S boxes, the probability to know all

the elements is p_t^8. We show the graphical image of both two curves, p_t and p_t^8 in Fig. 5. According to the probability, about 128 samples are enough to provide high probability to know all the elements in all eight S boxes.
The work effort of this attack includes computations for finding the slit pair (2^{32} encryptions and 2^{64} 32-bit data matching) and multiple encryptions of the slit pair to disclose S boxes (2×128). Since total amount is the sum of those two computations, that is approximated to 2^{32}.

4.2 MISTY

MISTY is a block cipher whose key schedule is designed relatively simple. After our detailed observation of MISTY's key schedule, we could find the very small key space of MISTY without *FL* functions which holds the sliding properties. However even with our enhancement of the slide attack, the key schedule of MISTY without *FL* functions is still resistant against slide attacks.
MISTY is based on provable security against differential and linear crypt-analyses [15]. If round keys are independent and uniformly distributed, three rounds of MISTY requires more than 2^{56} chosen(or known)-plaintext pairs for those two attacks.
In our study, we focused on the simplicity of the key-scheduling algorithm and investigated the possibility of our enhancement of slide attacks.
The key scheduling of MISTY is relatively simple. A 128-bit secret key, or eight 16-bit keys, K_i for $1\leq i\leq8$, is used to generate eight other 16-bit keys, K_i' for $1\leq i\leq8$, as follows:

$$K_i' = FI(K_i, K_{i+1}), 1 \leq i \leq 8,$$

where the index, 9, is reduced to 1. In the data-randomizing part, these 16 keys are used according to the following key schedule.
The first observation would agree with a specific characteristics of extended keys of a secret key that consists of identical 16-bit strings, i.e. eight K_is, for which resultant eight subkeys K_i's are identical. Therefore, for all rounds, KO_{i1}, KO_{i2}, KO_{i3}, and KO_{i4} are generated as the same 16-bit key, while KI_{i1}, KI_{i2}, and KI_{i3} are generated as other identical keys. There are 2^{16} keys, each of which serves the same keys for all rounds, apart from KL_{i1} and KL_{i2} functions. If we consider a modified model of MISTY, i.e. removing KL_{i1} and KL_{i2}, these 2^{16} keys allow for sliding properties.
In the sense of the original slide attacks, FO functions are not *weak*. However our extension of the slide attack could allow one to apply a known-plaintext attack on

a round function, e.g. linear cryptanalysis to FO function. Note that the linear probability of FO function is proven to be less than 2^{-28}, whereas the input-output size is 32 bits.

The question is still open as to whether there exists a sufficiently effective known-plaintext cryptanalysis on FO round function. However, in comparison with the work effort required to find a slit pair, no less than $2^{32/2}$ to apply the birthday paradox, the effective key size for the sliding property, 2^{16}, is much smaller. From this reason, the key-scheduling algorithm of MISTY is still secure against our enhancement of slide attacks.

We also note cryptographic importance of FL functions. As a consequence of our study, the existence of FL functions and the keying rules to these functions make it very hard to find sliding property. It looks that full specification of MISTY with FL functions are most unlikely to be vulnerable against any kinds of slide attacks. In MISTY, FL functions are cheap way to make hedges against cryptanalyses exploiting particular characteristics of a cipher, such as the sliding properties.

5 Concluding remarks

We described a novel way of combining slide attack and a known-plaintext cryptanalysis, and demonstrated some applications of the proposed enhancement. We also noted observations of key schedules properties relevant to slide attacks discussing the applicability of our attacks.

The proposed idea enhances slide attacks from two points of view: (1) the target round function can be more generalized; and (2) the required condition for key schedule is untightened. The first point is that in our enhanced attack, the cipher does not necessarily iterate a *weak* round function. The second point is that the iterating number of identical subkeys is not limited to one or two. Theoretically, there is a possibility that a cipher with more round subkey iteration is vulnerable against our enhancement.

References

1. E. Biham, A. Shamir, “Differential Cryptanalysis of DES-like Cryptosystems,” *Journal of Cryptology*, Vol.4, No.1, PP. 3-72, 1991. (The extended abstract was presented at CRYPTO'90.

2. E. Biham, A. Shamir, “Differential Cryptanalysis of the Data Encryption Standard,” *Springer-Verlag*, 1993.

3. A. Biryukov, D. Wagner, “Slide attacks,” *Preproceedings of FSE6, Fast Software Encryption Workshop 1999,* 1999.
4. A. Biryukov, D. Wagner, “Advanced Slide attacks,” *Advances in Cryptology, -EUROCRYPT2000, LNCS Vol. 1807, Springer- Verlag,* 2000.
5. D.W. Davies, “Some Regular Properties of the ‘Data Encryption Standard’ algorithm,” *Advances in Cryptology: Proceedings of CRYPTO82,* Plenum Press, 1983.
6. FIPS 46, “Data Encryption Standard,” Federal Information Processing Standards Publication 46, U.S. Department of Commerce/National Bureau of Standards, National Technical Information Service, Springfield, Virginia, 1977 (revised as FIPS46-1A988, FIPS46-2:1993, FIPS46-3:1999).
7. GOST, Gosudarstvennyi Standard 28147-89, “Cryptographic Protection for Data Processing Systems,” *Government Committee for the USSR for Standards,* 1989. (In Russian.)
8. L. R. Knudsen, “Cryptanalysis of LOKI91,” *Advances in Cryptology, -ASIACRYPT'91, LNCS Vol. 739, Springer-Verlag,* 1991.
9. B. S. Kaliski, M. J. B. Robshaw, “Linear Cryptanalysis Using Multiple Approximations,” *Advances in Cryptology, -CRYPTO'94, LNCS Vol. 839, Springer-Verlag,* 1994.
10. L. R. Knudsen, M. J. B. Robshaw, “Non-linear Approximations in Linear Crypt-analysis,” *Advances in Cryptology, -EUROCRYPT'96, LNCS Vol. 1070, Springer-Verlag,* 1996.
11. J. Kelsey, B. Schneier, D. Wagner, “Key-Schedule Cryptanalysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES,” *Advances in Cryptology, -CRYPTO'96, LNCS Vol. 1109, Springer-Verlag,* 1996.
12. S. K. Langford, M. E. Hellman, “Differential- Linear Cryptanalysis,” *Advances in Cryptology, -CRYPTO'94, LNCS Vol. 839, Springer-Verlag,* 1994.
13. M. Matsui, “Linear Cryptanalysis Method fur DES Cipher,” *Advances in Cryptology, -EUROCRYPT'93, LNCS Vol. 765, Springer-Verlag,* 1993.
14. M. Matsui, “The First Experimental Cryptanalysis of the Data Encryption Standard,” *Advances in Cryptology, - CRYPTO'94, LNCS Vol.839, Springer- Verlag,* 1994.
15. M. Matsui, “New Block Encryption Algorithm MISTY,” *Fast Software Encryption, 4th International Workshop*, *FSE'97, LNCS Vol. 1267, Springer-Verlag*, 1997.
16. M. J. Saarinen, “A chosen key attack against the secret *S*-boxes of GOST,” unpublished, available at http://www.jyu.fi/~mjos/gostcka.ps.
17. B. Schneier, “The GOST Encryption Algorithm,” *Dr. Dobb's Journal,* Vol. 20, No. 2, 1995.
18. B. Schneier, D. Whiting “Fast Software Encryption: Designing Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processor,” *Fast Software Encryption, 4th International Workshop, FSE'97, LNCS Vol. 1267,*

Springer-Verlag, 1997.
19. S. Vaudenay, "An experiment on DES statistical cryptanalysis," *Proc. of 3rd ACM CCCS,* 1996.

Table 1. Key scheduling of MISTY.

Round keys for FO function

Round key	KO_{i1}	KO_{i2}	KO_{i3}	KO_{i4}
Key data	K_i	K_{i+2}	K_{i+7}	K_{i+4}

Round keys for FI function

Round key	KI_{i1}	KI_{i2}	KI_{i3}
Key data	K'_{i+5}	K'_{i+1}	K'_{i+3}

Round keys for FL function

Round key	KL_{i1}	KL_{i2}
Key data	$K_{\frac{i+1}{2}}$ (odd i) (even i)	$K_{\frac{i+1}{2}+6}$ (odd i) $K_{\frac{i}{2}+4}$ (even i)

Au index, $i(\leq 8)$ is reduced to $i-8$.

Round	KO_{i1}	KO_{i2}	KO_{i3}	KO_{i4}	KI_{i1}	KI_{i2}	KI_{i3}	KL_{i1}	KL_{i2}
Actual	K_i	K_{i+2}	K_{i+7}	K_{i+4}	K'_{i+5}	K'_{i+1}	K'_{i+3}	$K_{\frac{j+1}{2}}$ (odd j) $K_{\frac{j}{2}+2}$ (even j)	$K_{\frac{j+1}{2}+6}$ (odd j) $K_{\frac{j}{2}+4}$ (even j)

Au index, $j(\leq 8)$ is reduced to $j-8$.