Improving Linear Cryptanalysis of LOKI91 by *Probabilistic Counting* Method

(Extended Abstract)

Kouichi SAKURAI Souichi FURUYA

Department of Computer Science and Communication Engineering,
Kyushu University, Hakozaki, Higashi-ku, Fukuoka 812-81, Japan
Email: sakurai@csce.kyushu-u.ac.jp

Abstract. We improve linear cryptanalysis by introducing a technique of probabilistic counting into the maximum likelihood stage.
In the original linear cryptanalysis based on maximum likelihood method with deterministic counting, the number of effective key and text bits is a multiple of the number of bit involved in the input to some S-box. Then, when larger S-boxes are used, 2R-method and even the 1R-methods can become impractical just because the number of effective text and key bits become excessive. Though 2R-method is practical for attacking DES, existing examples of ciphers where 2R-method is impractical include LOKI91.
We overcome this problem by selecting a part of the effective key bits and investigating the probabilistic behavior of the remained effective key bits. The previous attacks discusses deterministic evaluation of the given approximated formula only when all values of the effective text/key bits are known, while we compute the probability that the approximated formula with unknown inputs equals to zero.
This extension of linear cryptanalysis make useful for 2R-attack on LOKI91, then improves the performance of previous attacks. Furthermore, we implemented some experiments of attacks on 4-round LOKI91, and confirmed the effectiveness of our method.

1 Introduction

1R- and 2R-methods of Linear cryptanalysis: In a linear cryptanalysis developed by Matsui [Mat93, Mat94], the attacker identifies a (linear) relation between some bits of the plaintext, some bits of the ciphertext, and some bits of the user-provided key. Matsui showed that, if the relation does not hold exactly half the time, key information can be extracted by using maximum likelihood method with a large enough set of known plaintext and ciphertext pairs.

The fundamental method of linear cryptanalysis finds only one bit of the key, which is a parity of a subset of the key bits. Additional techniques of reducing the number of rounds of the approximations, by eliminating the first and/or last rounds, and counting on all the key bits affecting the data at the rounds not in the approximation can reduce the number of required plaintexts, and increase the number of key bits that the attack finds.

The block ciphers that we are now concerned with are iterative and repeatedly use a round transformation during encryption. In the 1R-method, the cryptanalyst guess the value of part of the use-provided key in either the first rounds or the last rounds. In the 2R-method, the guess is for the user-provided key from both the first and the last rounds simultaneously.

In practical implementation, we recover key bits using linear techniques by first counting the number of plaintext/ciphertext pairs that fall into a variety of classes. These classes are defined according to the text involved in the linear approximation (the effective text bits). We then process this data by guessing each possible value for the key bits involved in the linear approximation (the effective key bits) and combine this guess with the effective text, In this way scores can be kept for the number of times the bit identified by the linear approximation to the rest of the cipher is either zero or one. A guess can be made for the value of the effective key bits depending on these final scores.

Limitation of 2R-Method: We should note that the number of effective key and text bits is a multiple of the number of bit involved in the input to some S-box. Then, when larger S-boxes are used, 2R-method and even the 1R-methods can become impractical just because the number of effective text and key bits become excessive. Though 2R-method is practical for attacking DES, existing examples of ciphers where 2R-method is impractical include FEAL and LOKI91.

Our approach: To overcome the difficulty of applying 2R-method to ciphers with large S-boxes, instead of evaluating all effective text/key bits in the approximated equation in deterministic manner, we consider the probabilistic evaluation of the equation with unknown inputs. Namely, even though information on a part of inputs of the equation is not available, we can discuss the probability that the equation equal to zero, where the probability considers all possible patterns of unavailable input bits. Then, our probabilistic counting adds each probability, and finally judges the correct key among all key candidates according to maximum likelihood method. Thus, the probabilistic counting algorithm decreases the number of essentially evaluated effective (key) bits, which makes applicable the approximated formula for 2R-method into to ciphers with larger S-boxes,

Related works: A probabilistic approach in counting algorithm of linear cryptanalysis is initiated in Matsui's only-ciphertext attack of DES [Mat93]. Matsui eliminated the plaintext parts from the best linear expression for DES in a probabilistic manner under the assumption that plaintexts consists of natural English sentences represented by ASCII codes. Thus, Matsui's only-ciphertext attack considers the distribution of bits in plaintexts derived from the specific encoding, whereas we try to eliminate effective key bits by investigating the structure of S-boxes.

Aoki and Ohta [AO94] also considered the level in the role of effective bits in linear cryptanalysis of FEAL, in which even the direct 1R-method is impractical. Then, they applied a similar technique as our probabilistic method in reducing the required memories for implementing the attack on FEAL, though they counted not exact probabilities but only approximated probabilities by $\{0, 1\}$-values.

We should remark that Morris [Mor78] have considered how to reduce the

required registers for counting events by probabilistic arguments.

Another solution to discount excessive cost of 2R-method is proposed in [KR96]. They introduce newly discovered non-linear approximations into (multiple) linear cryptanalysis, which also achieve better performance than the previous 1R-attack on LOKI'91 [TSM94], while our probabilistic method uses the previously known (best) linear expressions. Then, this paper compares, via experimental performance, our method to the multiple linear cryptanalysis with non linear approximation [KR96].

Applying our idea into LOKI91: We investigates how the probabilistic counting method can be applied to linear cryptanalysis for LOKI91. Our theoretical estimation implies that

- For breaking 10-round LOKI91, our method needs 1.78×2^{54} known plaintexts with 2^{21} counter, while the multiple non-linear attack [KR96] requires 1.72×2^{56} known plaintexts with 2^{20} counters.
- For breaking 12-round LOKI91, our method with 2^{21} counters requires 1.88×2^{63} known plaintext, whereas the direct 1R-method [TSM94] with 2^{13} counters requires 1.97×2^{67} known plaintexts.

Experiments of our attacks: We implemented some experiments of attacks on 4 rounds LOKI91 for confirming our theoretical estimation on the number N of known-plaintexts required for successful attack. In particular, the success rate parameter $c = N/(p - 1/2)^2$, where p is the probability related to the used approximated formula, is an important for the practical performance on attacks

Then, we implemented similar experiments as one did in [KR96], to predict the success rate of our method. Namely, instead of direct implementing by using 2^{21} counters, we executed the counting algorithm by assuming that a part of target key (e.g. the effective key of the first round) is known, which decreases the number of implemented counter and makes feasible for implementing.

In our probabilistic method with 2^{21} counters, 4-round LOKI91 is breakable with 1.42×2^{18} known-plaintexts, which is theoretically estimated as $c = 8$. Under the condition that the 12 effective key bits of the first round is given to the cryptanalysis, we have implemented this attack with only $2^{21-12} = 2^9$ counters, and the experience results imply that 209,306 plaintexts ($c = 4.5$) is sufficient for achieving 96% success rate and $c = 6$ is sufficient for achieving 100% success rate, while the multiple non-linear attack [KR96] in the similar experiment with $2^{20-12} = 2^8$ counters is reported to require 1,442,632 plaintexts ($c = 8$) for achieving 96% success rate.

2 Preliminaries

2.1 Notation

Throughout this paper, we use the following notations.

P: The 64-bit data of the message
C: The 64-bit data of the encrypted message
$P_H(resp.P_L)$: The upper (resp. lower) 32-bit data of P, $P = (P_H, P_L)$
$C_H(resp.C_L)$: The upper (resp. lower) 32-bit data of C, $C = (C_H, C_L)$
$F_r(X_r, K_r)$: The r-th round F-function with input X_r and subkey K_r
Y^j: 6 input bits of the S_j box
$S_j(Y^j)$: the output of S_j box with the input Y^j
$A[i]$: The i-th bit of a binary vector A
Γ_X : the masked value of data X
$X[\Gamma_X]$: the even parity value of the bitwise AND between X and ΓX
$A[i, j, \cdots, k] := A[i] \oplus A[j] \oplus \cdots \oplus A[k]$

Note that we refer to the right most bit as the zero-th bit.

2.2 Description of LOKI91

LOKI91 is a 64-bit key/64-bit block cryptosystem similar to DES. This paper omits the details of the key-scheduling part because our analysis is independent of its algorithm. (See [BKPS91] for more precise description.) However, the F-function, which we describe below, plays an important role in our analysis.

The procedure of the LOKI91's i-round function F_i with the 32-bit input X_i and with 32-bit subkey K_i is defined as follows:

$$\begin{aligned} B' &= X_i \oplus K_i, & B &= E(B') \\ Y^0 &= B_{11}B_{10}\cdots B_0, & Y^1 &= B_{19}B_{18}\cdots B_8 \\ Y^2 &= B_{27}B_{26}\cdots B_{16}, & Y^3 &= B_3B_2\cdots B_0B_{31}\cdots B_{24} \\ O &= (S_3(Y^3)S_2(Y^2)S_1(Y^1)S_0(Y^0)), & F_i(X_i, K_i) &= P(O), \end{aligned} \tag{1}$$

where E is an expansion permutation with 32-bit input/48-bit output and P is a permutation over 32-bit. The computation $S_i(Y_i)$ in this procedure (1) is given in the following:

$$\begin{aligned} row = Y_{11}^i Y_{10}^i Y_1^i Y_0^i, col = Y_9^i Y_8^i \cdots Y_3^i Y_2^i \\ S_i(Y^i) = (col + ((row \times 17) \oplus ff_{16})\&ff_{16})^{31} (mod\ g_{row}) \end{aligned} \tag{2}$$

where g_{row}, which is selected from the set

$$\{375, 379, 391, 395, 397, 415, 419, 425, 433, 445, 451, 463, 471, 477, 487, 499\},$$

is an irreducible polynomial over $GF(2^8)$.

3 Our proposed method

3.1 Probabilistic 2R-method

We apply the proposed probabilistic counting into 2R-method in linear cryptanalysis.

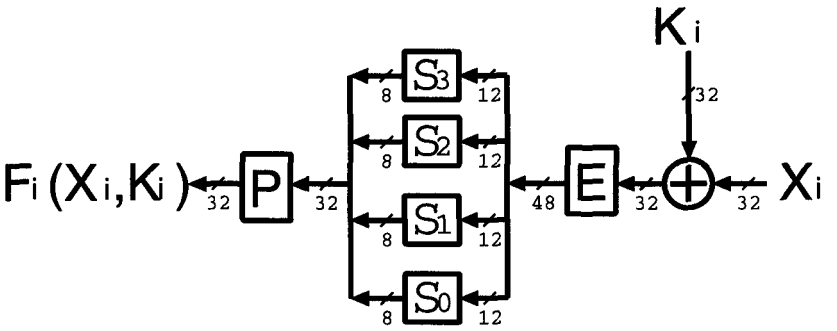

Fig. 1. F function of LOKI91

Consider the following approximated expression for n-round cipher, which is derived from the (best) linear expression for $(n-2)$-round version:

$$P[\Gamma_P] \oplus C[\Gamma_C] \oplus F_1(P_L, K_1)[\Gamma_{P_L}] \oplus F_n(C_L, K_n)[\Gamma_{C_L}] = K[\Gamma_K], \qquad (3)$$

which holds with probability $p \neq \frac{1}{2}$. Suppose that this formula includes t effective text bits and $k = k_1 + k_n$ effective key bits, where k_1 effective key bits are in F_1 and k_n effective key bits are F_n. So, the original counting algorithm requires $2^c + 2^t$ counters [Mat94].

Now we assume that only a part of the effective key bits is available, which we call *visible* effective key bits, and the other effective key bits, which is not available for attacker, are called *invisible*. In the evaluation of the approximated formula with invisible effective key bits, we consider the probability that the approximated formula equals to zero when the effective text bits and the visible effective key bits are deterministically given, where the probability takes the average over all possibilities of the *invisible* effective key bits.

We apply this probabilistic argument into $F_n(C_L, K_n)[\Gamma_{C_L}]$, which is assumed to include k_n^v visible effective key bits, while we use the deterministic evaluation for $F_1(P_L, K_1)[\Gamma_{P_L}]$ Then, the number of INvisible effective key bits is $k_n^{\overline{v}} = k_n - k_n^v$. Under this condition, the maximum likelihood method with probabilistic counting is implemented as follows.

[Probabilistic Counting Algorithm]

Step 1 Prepare 2^t counter $U_i (0 \leq i \leq 2^t)$, where i corresponds to each value on the t effective text bits of Equation (3).

Step 2 For each plaintext and the corresponding ciphertext, compute the value "i" of Step-1 and count up the counter U_i by one.

Step 3 Prepare $2^{k_1+k_n^v}$ counter $T_j (0 \leq j \leq 2^{k_v})$, where j corresponds to each value on the k_1 effective key bits of F_1 and the k_n^v VISIBLE effective key bits of F_n in Equation (3).

Step 4 For each "i" and "j", computes the probability p_{ij} that the left-side of Equation evaluated with "i" and "j" is zero, where the probability takes over all the "invisible" effective key bits. Then, set $T_j = \sum_i p_{ij} \times U_i$

Step 5 Let T_{exact} be the value which maximizes $|T_k - \frac{N}{2}|$, i.e., $|T_{exact} - \frac{N}{2}| = max_k(|T_k - \frac{N}{2}|)$. Then adopt the key candidate corresponding to T_{exact}.

Step 4 If $|T_{exact} - \frac{N}{2}| > \frac{N}{2}$, then guess that the right-side = 0. If $|T_{exact} - \frac{N}{2}| < \frac{N}{2}$, then guess that the right-side = 1.

Thus, the number of the counters required for implementing this algorithm is $2^t + 2^{k_1+k_n^v} = 2^t + 2^{k-k_n^{\overline{v}}}$, which corresponds to the number of counters, U_i and T_j, and the computational complexity for executing this algorithm is $O(N) + O(2^{t+k_1+k_n^v})$, where N is the number of the used known plaintexts.

Remark. Computing the probability p_{ij} for each (i, j) can be done by using a (common) precomputed table with $2^{k_n^v}$-entries, which is given in Appendix B. We describe the construction of this table in Subsection 3.2.

3.2 How to evaluate approximated formulas in a probabilistic manner

We discuss how to evaluate a given approximated formula with some unknown inputs in a probabilistic manner.

In the following, we consider an example of the parity $F_4(C_L, K_4)[18, 22, 26]$, which we use in our practical attacks on 4-round LOKI91 of Section 4 and is also useful for $N(> 4)$-round attack of Section 7. Note that $F_4(C_L, K_4)[18, 22, 26]$ is derived from the only single S-box S_2, namely, $S_2(Y^2)[70_{16}]$. Further, we assume that 4 bits $K_4[23], K_4[19], K_4[18], K_4[17]$ are invisible among 12 effective key bits of $K_4[27] \sim K_4[16]$, under which we attack 4-round LOKI91 in Section 4. Then, the input Y^2 of the corresponding S-box is $abcd?efg???h$, where $a, b, c, d, e, f, g, h \in \{0, 1\}$ and "?" denotes an unknown input bit for the cryptanalysis.

Consider 1001?100???1 as the input Y^2. Then, the possible inputs consist of the following 16 pattern:

100101000001 , 100101000011 , 100101000101 , 100101000111,
100101001001 , 100101001011 , 100101001101 , 100101001111,
100111000001 , 100111000011 , 100111000101 , 100111000111,
100111001001 , 100111001011 , 100111001101 , 100111001111

The each corresponding output of the S-box is the following:

01000110 , 01111111 , 00111000 , 10100010,
00100100 , 00101001 , 10100111 , 11101000,
10100001 , 00100001 , 01001111 , 00100111,
01001000 , 00101100 , 10011011 , 10101011

Then, the each corresponding parity of the output with 01110000 masked is the following:

$$\begin{array}{cccc} 1, & 1, & 0, & 1, \\ 1, & 1, & 1, & 0, \\ 1, & 1, & 1, & 1, \\ 1, & 1, & 1, & 1 \end{array}$$

This implies that only 2 times of the zero parity actually occur among 16 types of inputs. Thus, we observe that the parity of the output masked by 01110000 for input 1001?100???1 takes zero with probability 2/16 (under the assumption that all unknown input bits are selected in uniform and random), which is written as

$$p_s(1001?100???1, 01110000) = \frac{2}{16}.$$

The probability $p_s(abcd?efg???h, 01110000)$ for each $a, b, c, d, e, f, g, h \in \{0, 1\}$ is listed in Appendix B, which is easily tabulated by computer. Such a is obtained for a given pair of the pattern of the unknown input bits and the mask value of the output for a S-box. Therefor, in the probabilistic counting algorithm presented in Subsection 3.1, a common table is used for computing p_{ij} for any (i, j) at Step 4.

3.3 Analyzing our proposed probabilistic counting

The number of known-plaintexts required for successful linear cryptanalysis, if we use the approximated formula which holds probability p, is generally estimated as $c|p - 1/2|^{-2}$, where c is the success rate parameter that depends on the the approximated formula [Mat93, Mat94, TSM94]. We should note that, in our probabilistic counting method, the number of known-plaintexts required for successful attacking is depended not only upon the probability with which the approximated formula holds but also upon the probability of the bias of the parity of the output which we apply the probabilistic argument.

We investigate how to effect the probability p_s related to the probabilistic counting on the successful probability of the applied linear cryptanalysis. In the previous subsection, we consider the case of

$$p_s(1001?100???1, 01110000) = \frac{2}{16}.$$

as an example. Next we consider the average of the bias of such probabilities over all possible inputs $abcd?efg???h$, i.e.,

$$\tilde{p}(abcd?efg???h, 01110000) = \frac{1}{2^8} \sum_{i=0000?000???0}^{1111?111???1} |\frac{1}{2} - p_s(i, 01110000)|$$

Throughout the remained part of this paper, instead of $abcd?efg???h_2$, we use the notation 111101110001, where "1" denotes the positions of the available bits, then

# invisible bits	invisible pattern x	most unbalanced $\tilde{p}(x, 70_{16})$
11	001000000000_2	1.4375000×2^{-7}
10	000001100000_2	1.1250000×2^{-6}
9	010001000001_2	1.6875000×2^{-6}
8	011001000100_2	1.1562500×2^{-5}
7	011001000101_2	1.5468750×2^{-5}
6	011001000111_2	1.0468750×2^{-4}
5	011001100111_2	1.3437500×2^{-4}
4	111101110001_2	1.7265625×2^{-4}
3	011101011111_2	1.1718750×2^{-3}
2	010111111111_2	1.5664063×2^{-3}
1	011111111111_2	1.0175781×2^{-2}
0	111111111111_2	1.0000000×2^{-1}

Table 1. Most unbalanced values of $\tilde{p}(x, 70_{16})$

$\tilde{p}(abcd?efg???h, 01110000)$ is written as $\tilde{p}(111101110001, 01110000)$. The exact value of $\tilde{p}(111101110001, 01110000)$, which can be computed from the table in Appendix B, is 442/4096. This can be generalized for computing $\tilde{p}(x, \Gamma_y)$.

We have computed which value of x takes the maximum among the set with a common number of invisible inputs for the output mask 70_{16}, which is listed in Table 1. In the linear cryptanalysis with the probabilistic counting, once we decide the number of invisible effective key bits, we choose the position of these invisible bits according to Table 1. Remark that the value of the output mask 70_{16} is decided from the initially given approximated formula.

Thus, as the similar argument as Matsui's *Piling-up Lemma* [Mat93], under the assumption that the all keys are independent, the number of known-plaintexts required for successful probabilistic counting algorithm is theoretically estimated as:

$$N = c \times (2 \times |p_{linear} - 1/2| \times \tilde{p})^{-2}, \tag{4}$$

where p_{linear} is the probability which the given linear approximated formula holds with.

4 Attacking 4-round LOKI91

4.1 Previous deterministic 2R-method

The best linear expression of 2-round LOKI91 is computed in [TSM94] as the following formula (5):

$$P_H[\alpha] \oplus P_L[\alpha] \oplus C_L[\alpha] = K_1[\alpha] \quad (\alpha = 18, 22, 26), \tag{5}$$

which holds with probability $p_2 = \frac{1}{2} - 1.38 \times 2^{-6}$. Then, the 2R-method with this 2-round best linear expression induces the following expression for 4-round LOKI91:

$$P_H[\alpha]P_L[\alpha] \oplus C_H[\alpha] \oplus F_1(P_L, K_1)[\alpha] \oplus F_4(C_L, K_4)[\alpha] = K_2[\alpha] \qquad (6)$$

which holds as the same probability as p_2.

The expression 6 has 24-bits of the keys $K_1[27]K_1[26] \cdots K_1[16]\, K_4[27] \cdots K_4[16]$, which effect the evaluation of its left-side, and 24-bit effective text bits. Then, the original (deterministic) 2R-method requires 2^{25} counters and 2^{48} working complexity for implementing, which requires 1.06×2^{14} known plaintexts.

4.2 Our probabilistic 2R-method

To overcome the time/memory constraint of the previous 2R-method, instead of dealing with all effective key bits, we consider only a part of the effective key bits: we regard 4 bits $K_4[23], K_1[19], K_4[18], K_4[17]$ of the 4th round of F_4 as invisible.

In this case, though the attacker can deterministically decide the parity $F_1(P_L, K_1)[\alpha]$, he cannot evaluate $F_4(C_L, K_4)[\alpha]$ in deterministic manner because he does not get the complete information on inputs K_4. Instead of computing the deterministic parity, the we compute the probability that the parity of $F_4(C_L, K_4)[\alpha]$ takes 0 by using a precomputed table with 2^8 size, which tabulated in Appendix. By using this probability, we can obtain the probability that the left side of Equation (6) equals to zero.

This procedure is executed for all given plain/cipher texts and we get the information of the key by using the maximum likelihood primitive. Thus, the cryptanalysis regards 12 key-bits of F_1 and 8 key-bits of F_4 as the visible effective key bits, and extracts the user-key candidate by using the probabilistic counting algorithm presented in Subsection 3.1.

Now, we discuss the efficiency of our attack on 4-round LOKI91. The probability that the applied approximated formula (5) holds is p_2, and the expected bias of success probability of guessing the parity of the 4th-round function F_4 is $\tilde{p}(f71_{16}, 70_{16}) = 1.73 \times 2^{-4}$. So, *Piling-up Lemma* which we discuss at Subsection 3.3, implies that the success probability of the known plaintext attack is

$$p = \frac{1}{2} + 2 \times (1.38 \times 2^{-6}) \times (1.73 \times 2^{-4}) = \frac{1}{2} + 1.19 \times 2^{-8} \qquad (7)$$

Then, Formula (4) gives a theoretical estimation on the number of the known plaintext required for breaking the cipher that:

$$N_4 = 8.0 \times (1.19 \times 2^{-8})^{-2} = 1.42 \times 2^{18} \qquad (8)$$

Remark. In this theoretical estimation, we assume that the value of the success rate parameter is $c = 8.0$ as the previous attacks in [TSM94, KR96]. In Section 6, we discuss the experimental confirmation on this assumption.

This is 26% of the number 1.38×2^{20} of the required plaintexts in multiple non-linear cryptanalysis with 2^{20} counters and 2^{38} working effort [KR96]. Since this attack uses 20 visible effective key bits and 20 effective text bits, the probabilistic counting can be implemented over 2^{21} counters with 2^{40} time complexity.

Remark. If one counter costs 2 byte memory, then 2^{21} counters corresponds to 4 Mega byte memory, which is available over a recent PC. Furthermore, 2^{40} working complexity is less than the required time for Matsui's experimental attack on 16-round DES [Mat94].

5 More flexibility for attackers

In the original deterministic counting algorithm, the cryptanalyst is restricted to using a number of effective key and text bits which is a multiple of the number of bit involved in the input to some S-box. As we showed in the previous subsection, our new probabilistic method make possible 2R-attack on LOKI'91, which was impractical in the previous deterministic method.

However, the probabilistic method described in Subsection 4.2 requires 2^{21} counters, which is still impractical for certain attackers. This is improved by increasing the number of INVISIBLE effective key bits, which we apply the probabilistic argument. Suppose that the cryptanalyst is restricted to handle with 2^17 counters because of the memory-constraint on his PC. In this case, the attacker deals with 4 bits, $K_4[26]K_4[25]K_4[22]K_4[18]$ of 12 effective key bits $K_4[27]K_4[26]\cdots K_4[16]$ of the 4-round function in Formula (6) as VISIBLE. Then, he must compute the parity of S_2 with masked by 70_{16} from 4 bits information $Y^2[10]Y^2[9]Y^2[6]Y^2[2]$ of the 12 bits input Y^2. Note that bias $\tilde{p}(644_{16}, 70_{16}) = 1.16 \times 2^{-5}$, then we can theoretically estimate that the number of plaintexts required for breaking 4-round LOKI91 is

$$N_4' = 8 \times (2 \times 1.16 \times 2^{-5} \times 1.38 \times 2^{-6})^{-2} = 1.58 \times 2^{21} \tag{9}$$

Though this is 8.92 times of the required number of plaintexts N_4 in 8 of the previous attack with 2^{21} counters, the cryptanalysis can execute this attack with $\frac{1}{256}$ times working complexity as the previous case.

On the other hand, a cryptanalysis could be alive, who has much more memory for counters, though he cannot do complete implementation on the direct 2R-attack which requires 2^{25} counters. Suppose that he can use 2^{23} counters. In this case, the attacker deals with only 1 bits, $K_4[25]K_4[22]K_4[18]$ of 12 effective key bits $K_4[27]K_4[26]\cdots K_4[16]$ of the 4-round function in Formula (6) as *invisible*.

Then, he computes the parity of S_2 with masked by 70_{16} from 11 bits information expect $Y^2[9]$ of the 12 bits input Y^2. The bias $\tilde{p}(5ff_{16}, 70_{16}) = 1.57 \times 2^{-3}$, then we can theoretically estimate that the number of plaintexts required for breaking 4-round LOKI91 is

$$N_4'' = 8 \times (2 \times 1.57 \times 2^{-3} \times 1.38 \times 2^{-6})^{-2} = 1.72 \times 2^{16} \tag{10}$$

x	# counters	*#plaintexts*	work effort
	†2^{13}	1.49×2^{22}	2^{24}
060_{16}	2^{15}	1.67×2^{23}	2^{28}
441_{16}	2^{16}	1.49×2^{22}	2^{30}
644_{16}	2^{17}	1.58×2^{21}	2^{32}
645_{16}	2^{18}	1.77×2^{20}	2^{34}
647_{16}	2^{19}	1.93×2^{19}	2^{36}
667_{16}	2^{20}	1.17×2^{19}	2^{38}
$f71_{16}$	2^{21}	1.42×2^{18}	2^{40}
$75f_{16}$	2^{22}	1.54×2^{17}	2^{42}
$5ff_{16}$	2^{23}	1.72×2^{16}	2^{44}
$7ff_{16}$	2^{24}	1.02×2^{16}	2^{46}
fff_{16}	‡2^{25}	1.06×2^{14}	2^{48}
	*2^{20}	1.38×2^{20}	2^{38}

† The deterministic 1R attack [TSM94]
‡ The deterministic 2R attack (hypothetical)
* The multiple non-linear attack [KR96]

Table 2. Efficiency of each predict of 4th round F function

Though this requires 16 times working complexity as the attack with 2^{21} counters, the required number of plaintexts decrease into 0.30 times as N_4 of (8) in the previous attack.

Furthermore, the attacker can control the number of the VISIBLE part of the effective key bits in more flexible manner according to his computational resource. The performance of the variant attacks is listed in Table 2. Note that we assume that the success rate $c = 8$ for all cases, the correctness of which we discuss in our experiment of Section 6.

6 Experimental verification of our probabilistic method

We have carried out our experiment on 4-round version of LOKI91 for confirming our theoretical analysis.

The number of known-plaintexts required for successful linear cryptanalysis with the approximated formula, which holds probability p is generally estimated as $c|p-1/2|^{-2}$, where c is the success rate parameter depends on the the approximated formula. In the case of LOKI91, Tokita et al. [TSM94] estimated $c = 8$ for 1R-attack's achieving 100% success rate. However, if the number of the effective key bits in the used approximated formula increase, the parameter c would become bigger for achieving 100% success rate. So, deciding the exact value of the parameter c is an important for estimating the efficiency of the discussed attack.

To this end, we implemented some experiments, which is similar to one did in [KR96], to predict the success rate of our method. Namely, instead of dir-

# invisible bits	invisible pattern x	c					
		2.0	4.0	6.0	8.0	10.0	12.0
10	060_{16}	39%	48%	51%	56%	55%	65%
9	441_{16}	21%	34%	29%	39%	35%	32%
8	644_{16}	74%	95%	99%	99%	—	—
7	645_{16}	78%	95%	100%	100%	—	—
6	647_{16}	79%	95%	99%	100%	—	—
5	667_{16}	67%	92%	99%	100%	—	—
4	$f71_{16}$	65%	94%	100%	100%	—	—
3	$75f_{16}$	61%	99%	99%	100%	—	—
2	$5ff_{16}$	60%	96%	100%	100%	—	—
1	$7ff_{16}$	61%	98%	100%	100%	—	—

Table 3. Success rate of variant attacks

ect implementing by using 2^{21} counters, we executed the counting algorithm by assuming that a part of target key (e.g. the effective key of the first round) is known, which decreases the number of implemented counter and makes easy for implementing.

The obtained successful rate over 100 trials for the theoretical estimation on Table 2 is given in Table3, which we assume that 12 bits of effective key used in the first round remain fixed and known.

Remark. In Table 3, the (strange) degeneration of the success rate in the experimental result for the pattern $x = fff_{16}$, which corresponds to the deterministic 2R-method, suggests that the performance of the probabilistic counting in our theoretical analysis of Subsection 3.3 should be improved by more refined discussion.

On the other hand, the experimental results for two pattern $x = 060_{16}, 441_{16}$ show that recovery of only a few bits is not reliable, a similar fact is reported in [KR96]. This is because that there are two or more key candidates whose probabilistic behavior is quite similar. In Appendix A, we give a refined analysis with considering the behavior of incorrect-keys, and show other patterns, which achieves better performance than $x = 060_{16}, 441_{16}$ above.

Furthermore, we have implemented with variable counters in the case of the invisible pattern $x = f71_{16}$ with 4 invisible bits for observing how the success rate degrade when the number of recovered key bits increases. The experimented results over 100 trials are tabulated in Table 4, where the number of the known key bits of the first round changes while the number of the recovered key bits is fixed as 8: 4 bits are visible and 4 bits are invisible.

To the end of this section we estimate how success rates degrade. Following Table 5 is the summarized experimental results of variant numbers of the recovered bits for $x = 644_{16}$.

# key bits recoverd		c				
1st round F_1	4th round F_4	2.0	3.0	4.0	6.0	8.0
0	8	65%	84%	94%	100%	100%
1	8	64%	87%	96%	99%	100%
2	8	56%	78%	90%	99%	100%
3	8	48%	72%	94%	100%	100%
4	8	30%	78%	86%	100%	100%

Table 4. Success rate of variant numbers of the recovered bits for $x = f71_{16}$

#key bits recovered		c			
1st round F_1	4th round F_4	$c = 2.0$	$c = 4.0$	$c = 6.0$	$c = 8.0$
0	4	74%	95%	99%	99%
1	4	84%	100%	98%	100%
2	4	48%	94%	98%	100%
3	4	58%	96%	100%	96%
4	4	58%	84%	96%	100%
5	4	38%	88%	100%	98%
6	4	38%	80%	100%	100%
7	4	34%	80%	98%	100%
8	4	34%	72%	92%	100%
9	4	14%	68%	96%	96%
10	4	16%	72%	92%	100%
11	4	-%	-%	86%	97%
12	4	-%	-%	80%	96%

Table 5. Success rate of variant numbers of the recovered bits for $x = 664_{16}$

7 Attacking n-round LOKI91

We apply the probabilistic counting method into $n(> 4)$ round LOKI91.

As the 16-round LOKI91 [TSM94], iterative linear approximations is useful for the best linear approximations of $n = 4, 5, 7, 8, 10$. Then, we can apply the similar argument of the attack on 4-round LOKI91 into these reduced round LOKI91.

We discuss the efficiency of attacks under the assumption that we have only 2^{21} counters for 20 visible effective key bits:

$$K_1[27]K_1[26]K_1[25]K_1[24]K_1[23]K_1[22]K_1[21]K_1[20]K_1[19]K_1[18]K_1[17]K_1[16]$$
$$K_n[27]K_n[26]K_n[25]K_n[24]K_n[22]K_n[21]K_n[20]K_n[16]$$

Then, the remained effective bits $K_n[23], K_n[19]K_n[18]K_n[17]$ are regarded as invisible, over which we argue the probabilistic behavior.

Round	# plaintexts	# counters	work effort
4	1.42×2^{18}	2^{21}	1.00×2^{40}
6	1.50×2^{27}	2^{21}	1.00×2^{40}
7	1.59×2^{36}	2^{21}	1.00×2^{40}
9	1.68×2^{45}	2^{21}	1.68×2^{45}
10	1.78×2^{54}	2^{21}	1.78×2^{54}
12	1.88×2^{63}	2^{21}	1.88×2^{63}

Table 6. Attacks on each round LOKI91 with 2^{21} counters

In the case of 10-round LIKI91, the best linear expression of 8-round LOKI91 implies via 2R-method the following approximated formula (11) for 10-round LOKI91, which hold with probability $\frac{1}{2} - 1.23 \times 2^{-24}$.

$$P_H[\alpha] \oplus P_L[\alpha] \oplus C_H[\alpha] \oplus F_1(P_L, K_1)[\alpha] \oplus F_{10}(C_L, K_{10})[\alpha] = K_2[\alpha] \oplus K_4[\alpha] \oplus K_5[\alpha] \oplus K_7[\alpha] \oplus K_8[\alpha] \quad (11)$$

Then, the number of the known plaintext required for breaking 10-round LOKI91 is theoretically given as follows:

$$N_{10} = 8.0 \times (2 \times 1.23 \times 2^{-24} \times 1.73 \times 2^{-4})^{-2} = 1.78 \times 2^{54} \quad (12)$$

We should recall that the multiple non-linear attack requires 1.72×2^{56} known plaintexts with 2^{20} counters for breaking 10-round LOKI91 [KR96].

In the case of 12-round LIKI91, the best linear expression of 10-round LOKI91 implies via 2R-method the following approximated formula (13) for 12-round LOKI91, which hold with probability $\frac{1}{2} + 1.69 \times 2^{-29}$:

$$P_H[\alpha] \oplus C_H[\alpha] \oplus F_1(P_L, K_1)[\alpha] \oplus F_{12}(C_L, K_{12})[\alpha] = K_3[\alpha] \oplus K_4[\alpha] \oplus K_6[\alpha] \oplus K_7[\alpha] \oplus K_9[\alpha] \oplus K_{10}[\alpha] \quad (13)$$

Then, a theoretically estimation implies that the number of the known plaintext required for breaking 12-round LOKI91 is

$$N_{12} = 8.0 \times (2 \times 1.69 \times 2^{-29} \times 1.73 \times 2^{-4})^{-2} = 1.88 \times 2^{63} \quad (14)$$

The performance of these attacks is listed in Table 6. Thus, in our theoretical estimation, 12-round LOKI91 is breakable with 1.88×2^{63} known plaintexts faster than an exhaustive search for 64-bits keys, whereas the direct 1R-method [TSM94] with 2^{13} counters requires 1.97×2^{67} known plaintexts with 2^{67} working effort for breaking 12-round LOKI91.

8 Concluding remarks

Yet another extension of linear cryptanalysis has been presented by introducing probabilistic counting method. This new method improves the performance of the linear cryptanalysis of LOKI91, which correctness was confirmed via our implemented experiments. We note that, though we have examined 3R- and 4R-attack on DES, no significant advantage over existing attacks on DES are yet obtained.

A future research topic is to optimize our method. We have used the best linear expression for k-round LOKI91. However, alternative better expression could exist for our probabilistic method, which would improve the performance of our attack. Furthermore, a hybrid attack between our method and the others [LH94, KR94, KR96] shall be investigated for clarifying the limitation of the linear cryptanalysis, which is useful for designing provably secure block ciphers [NK95].

The final remark is that our probabilistic-counting method can be applicable to any statistical attack [Vau96], based on the maximum likelihood principle, which includes differential cryptanalysis [BS91, BS93], though advantage of applied cryptanalysis over existing attacks remains open.

Acknowledgments

The authors would like to thank the referees for helpful comments on the submitted version. Also thanks a referee's comment on the strange degeneration of the authors' experimental results and Robshaw's email-answer [Rob96] to the first author's query on the paper [KR96], which lead the authors to more detailed analysis of the proposed attack and improvements described in Appendix A. Final thanks Ronald Rivest for remarking the connection between an initiated research of Morris [Mor78] and our probabilistic counting method.

References

[AO94] K.Aoki, and K.Ohta, "Linear Cryptanalysis of the Fast Data Encipherment Algorithm," *Tech. Rept. of IEICE,* ISEC94-5 (1994).

[BKPS91] L. Brown, M. Kwan, J. Pieprzyk, J. Seberry, "Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI," *Advances in Cryptology, - ASIACRYPT'91, LNCS Vol. 739, Springer-Verlag*, 1991.

[BPS90] L. Brown, J. Pieprzyk, J. Seberry, "LOKI - A Cryptographic Primitive for Authentication and Secrecy Applications," *Advances in Cryptology, - AUSCRYPT'90, LNCS Vol. 453, Springer-Verlag*, 1990.

[BS91] E. Biham, A. Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer," *Advances in Cryptology, - CRYPTO'91, LNCS Vol.576, Springer-Verlag*, 1991.

[BS93] E. Biham, A. Shamir, "Differential Cryptanalysis of of the Data Encryption Standard," *Springer-Verlag*, 1993.

[Knu91] L. R. Knudsen, "Cryptanalysis of LOKI," *Advances in Cryptology, - ASIACRYPT'91, LNCS Vol. 739, Springer-Verlag*, 1991.

[Knu92] L. R. Knudsen, "Cryptanalysis of LOKI91," *Advances in Cryptology, - AUSCRYPT'92, LNCS Vol. 718, Springer-Verlag*, 1992.

[KR94] B. S. Kaliski, M. J. B. Robshaw, "Linear Cryptanalysis Using Multiple Approximations," *Advances in Cryptology, - CRYPTO'94, LNCS Vol.839, Springer-Verlag*, 1994.

[KR96] L. R. Knudsen, M. J. B. Robshaw, "Non-linear Approximations in Linear Cryptanalysis," *Advances in Cryptology, - EUROCRYPT'96, LNCS Vol. 1070, Springer-Verlag*, 1996.

[LH94] S. K. Langford, M. E. Hellman, "Differential-Linear Cryptanalysis," *Advances in Cryptology, - CRYPTO'94, LNCS Vol. 839, Springer-Verlag*, 1994.

[Mat93] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *Advances in Cryptology, - EUROCRYPT'93, LNCS Vol. 765, Springer-Verlag*, 1993.

[Mat94] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard," *Advances in Cryptology, - CRYPTO'94, LNCS Vol. 839, Springer-Verlag*, 1994.

[Mor78] R. Morris, "Counting large numbers of events in small registers," *Comm. of the ACM*, Vol.21, No.10 (1978).

[Nyb94] K.Nyberg, "Linear approximation of block ciphers," *Advances in Cryptology, - EUROCRYPT'94, LNCS Vol. 950, Springer-Verlag*, 1995.

[NK95] K.Nyberg and L.R.Knudsen, "Provable security against a differential attack." *J. Cryptology*, Vol.8, No.1, pp.27-37 (1995).

[OA94] K. Ohta, K. Aoki, "Linear Cryptanalysis of the Fast Data Encipherment Algorithm," *Advances in Cryptology, - CRYPTO'94, LNCS Vol. 839, Springer-Verlag*, 1994.

[OMA95] K. Ohta, S. Moriai, and K. Aoki, "Improving the search algorithm for best linear expression," *Advances in Cryptology, - CRYPTO'95, LNCS Vol. 963, Springer-Verlag*, 1995.

[Riv97] R. Rivest, "Oral remark on Morris's work [Mor78] at the second author's presentation of Fse4," Jan. 1997

[Rob96] J. B. Robshaw, "Email-answer to the first author's query on the experimental results on [KR96]," Oct. 1996.

[Vau96] S.Vaudenay, "An experiment on DES statistical cryptanalysis," *Proc. of 3rd ACM CCCS* , 1996.

[TSM94] T. Tokita, T. Sorimachi, M. Matsui, "Linear Cryptanalysis of LOKI and s^2DES," *Advances in Cryptology, - ASIACRYPT'94, LNCS Vol.917, Springer-Verlag*, 1994.

A A refined analysis of the probabilistic method

A.1 Theoretical formula of Success rate

Subsection 3.3 roughly discussed the performance of our probabilistic method by simply applying Matsui's *Piling-up Lemma* [Mat93]. This appendix considers an exact performance by dealing with the behavior of incorrect keys.

First, we consider the following linear approximation to use the cryptanalysis:

$$P_h[\Gamma P_h] \oplus P_l[\Gamma P_l] \oplus C_h[\Gamma C_h] \oplus C_l[\Gamma C_l] \quad (15)$$
$$\oplus F_1(P_l, K_1)[\Gamma P_h] \oplus F_n(C_l, K_n)[\Gamma C_h] = K[\Gamma K]$$

Now we assume that the parity of the 1st round F-function $F_1(P_l, K_1)[\Gamma P_h]$ and the parity of the nth round F-function $F_n(C_l, K_n)[\Gamma C_h]$ are derived from a single S-box. Then, by using the notation

$$PC[\Gamma PC] = P_h[\Gamma P_h] \oplus P_l[\Gamma P_l] \oplus C_h[\Gamma C_h] \oplus C_l[\Gamma C_l],$$

we can write the above linear expression as follows:

$$PC[\Gamma PC] \oplus S(Y_{1,x})[\Gamma S_1] \oplus S(Y_{n,y})[\Gamma S_n] = K[\Gamma K].$$

In the equation above, the notation $Y_{k,x}$ denotes inputs of the x-th S-box[1] in the k-round F-function, and ΓS_1,ΓS_n denote each mask obtained from ΓP_h,ΓC_h via expanding function E of each F-function.

Let $\frac{1}{2} + p'$ $(p' > 0)$ be the probability that this linear expression holds for the correct key $K_1^T K_n^T$. Then, the success rate of the original linear cryptanalysis with N random pairs of plain/cipher-texts is estimated [Mat93] in the following formula:

$$SR' = \int_{\frac{N}{2}}^{\infty} \prod_{\kappa=1}^{m} WE_\kappa(x) \sqrt{\frac{2}{\pi N}} \exp\left\{ -\frac{2(x-\mu_T)^2}{N} \right\} dx$$

Note that, in this formula,

$$\mu_T = N(\frac{1}{2} + p'),$$

m is the number of the key candidates which are not correct, and

$$WE_\kappa(x) = Prob\left(T^\kappa : \frac{N}{2} - x < T^\kappa < \frac{N}{2} + x \right),$$

where T^κ denotes the value of the counter corresponding to the (wrong) key candidates $K^W = K^T \oplus \kappa$.

Next we consider how to modify the formula above in the case with probabilistic counting.

In our attack, assume that there exists 12 effective textbits and 12 effective keybits for the first round S-box with the 12 input bits, and there exists $d(< 12)$ effective textbits and d effective keybits for the n round S-box with the 12 input bits. Then, the number of key candidates is 2^{12+d}, which the correct key exists in.

For simplifying the discussion, we consider the case of $d = 2$. In this case, the attacker guesses the value $S(Y_{n,y})[\Gamma S_n]$ is O with probability $\frac{1}{2} + \varepsilon_{ab}$, which is computed as discussed in Subsection 3.2, and corresponds to such a distribution table given in Appendix B.

[1] Note that, in the discussed attack of LOKI91, $x = 2$.

$a\backslash b$	0	1
0	$\frac{1}{2}+\varepsilon_{00}$	$\frac{1}{2}+\varepsilon_{01}$
1	$\frac{1}{2}+\varepsilon_{10}$	$\frac{1}{2}+\varepsilon_{11}$

Now we consider the following assumption.

> **Assumption A:** The distribution of the statistic T_n corresponding to the key candidate $K_n = K^T \oplus \kappa_n$ can be modeled by using a bimonomial distribution.

Theorem: Under Assumption A, the success rate of our probabilistic-counting linear-cryptanalysis with N random pairs of plain/cipher-texts is

$$\int_{\frac{N}{2}}^{\infty}\left\{\prod_{\kappa_n=01_2}^{11_2}\int_{\frac{N}{2}-x}^{\frac{N}{2}+x}\sqrt{\frac{2}{\pi N}}\exp\left(-\frac{2}{N}(y-\frac{N}{2}+(\sum_{i=00_2}^{11_2}\varepsilon_{i\oplus\kappa_n}\frac{\varepsilon_i N}{2}))^2\right)dy\right\}$$
$$\times\sqrt{\frac{2}{\pi N}}\exp\left\{-\frac{2}{N}\left(x-N(\frac{1}{2}+p')\right)^2\right\}dx$$

The proof of this theorem is omitted from this extended abstract.

We should remark that this formula can be easily generalized into the case $d(2 \le d < 12)$ as follows.

$$\int_{\frac{N}{2}}^{\infty}\left\{\prod_{\kappa_n}\int_{\frac{N}{2}-x}^{\frac{N}{2}+x}\sqrt{\frac{2}{\pi N}}\exp\left(-\frac{2}{N}\times(y-\frac{N}{2}+(\sum_{i=0}^{2^d-1}\varepsilon_{i\oplus\kappa_n}\frac{\varepsilon_i N}{2}))^2\right)dy\right\}$$
$$\times\sqrt{\frac{2}{\pi N}}\exp\left\{-\frac{2}{N}\left(x-N(\frac{1}{2}+p')\right)^2\right\}dx.$$

We calculated the formula for the corresponding ε_i to the parameters of Table 3 of experimental results.

A.2 Numerical versus Experimental

We compare success rates numerically computed from the theoretical formula to experimentally obtained success rates. Note that, in both case of numerical and experimental, we assume that the attacker knows the key of the first round in advance, and discuss success rate to find the n-th round key.

First, we consider the case when the invisible pattern is $x = 060_{16}$. Though this is the most unbalanced among the patterns with 10 invisible bits (see Table 1), the experimental result in Table 3 shows that this pattern is not so good for attacking. The distribution table $p_s(x = ?????ab?????_2, 01110000)$ is the following:

$$\varepsilon_i = \begin{pmatrix} \frac{22}{1024} & \frac{-20}{1024} \\ \frac{-16}{1024} & \frac{14}{1024} \end{pmatrix}$$

Then, via the theoretical formula with the distribution table above implies the following relation between the parameter c and the success rate.

# *invisible bits*	*invisible pattern* x	success rate parameter c					
		2.0	4.0	6.0	8.0	10.0	12.0
10	060_{16}	27%	28%	29%	30%	31%	31%
9	441_{16}	19%	21%	22%	24%	25%	25%
8	644_{16}	93%	99%	100%	100%	—	—
7	645_{16}	99%	100%	100%	100%	—	—
6	647_{16}	99%	100%	100%	100%	—	—
5	667_{16}	95%	99%	100%	100%	—	—
4	$f71_{16}$	87%	97%	99%	100%	—	—
3	$75f_{16}$	74%	89%	95%	98%	—	—
2	$5ff_{16}$	60%	77%	85%	91%	—	—
1	$7ff_{16}$	48%	64%	74%	80%	—	—

Table 7. Calculation of success rate of varianlt attacks

c	2.0	4.0	6.0	8.0	10.0	12.0
Numerical	27%	28%	29%	30%	30%	31%
Experiment	39%	48%	51%	56%	55%	65%

We give some results on other patterns with 10 invisible bits for $c = 8$ in the following table.

x	014_{16}	044_{16}	084_{16}
N	31,551,642	45,434,364	24,929,692
$\begin{pmatrix} \varepsilon_{00_2} & \varepsilon_{01_2} \\ \varepsilon_{10_2} & \varepsilon_{11_2} \end{pmatrix}$	$\begin{pmatrix} \frac{-8}{1024} & \frac{24}{1024} \\ \frac{-12}{1024} & \frac{-4}{1024} \end{pmatrix}$	$\begin{pmatrix} \frac{-2}{1024} & \frac{4}{1024} \\ \frac{-18}{1024} & \frac{16}{1024} \end{pmatrix}$	$\begin{pmatrix} \frac{-27}{1024} & \frac{18}{1024} \\ \frac{7}{1024} & \frac{2}{1024} \end{pmatrix}$
Numerical	85%	51%	71%
Experiment	100%	72%	100%

The comparision on the pattern $x = 441_{16}$, which is the most unbalanced among 9 invisible bits, is the following.

c	2.0	4.0	6.0	8.0	10.0	12.0
Numerical	19%	21%	23%	24%	25%	25%
Experiment	21%	34%	29%	39%	35%	32%

The comparison in the case of other patterns with 9 invisible bits for $c = 8$ is the following.

x	007_{16}	064_{16}	260_{16}	441_{16}	484_{16}	602_{16}	604_{16}
N	6,721,059	7,887,910	8,588,726	6,721,059	8,588,726	6,987,214	6,721,059
Numerical	50%	85%	95%	24%	100%	50%	100%
Experiment	57%	97%	100%	38%	100%	52%	100%

Thus, the refined formula well simulates the experimental results on patterns with a few visible bits, and suggests how to choose the invisible pattern for more strong attacks on LOKI91.

B The distribution table of p_s

$abcd?e\backslash fg???h$	$00???0_2$	$00???1_2$	$01???0_2$	$01???1_2$	$10???0_2$	$10???1_2$	$11???0_2$	$11???1_2$
$0000?0_2$	9	6	7	3	7	4	6	9
$0000?1_2$	8	9	5	11	7	8	8	8
$0001?0_2$	10	13	7	11	12	7	9	7
$0001?1_2$	3	5	5	6	9	8	11	10
$0010?0_2$	9	9	9	11	9	6	4	8
$0010?1_2$	13	10	8	7	11	8	9	9
$0011?0_2$	8	11	5	5	7	9	5	8
$0011?1_2$	6	9	10	6	12	6	8	9
$0100?0_2$	10	5	8	7	9	8	8	11
$0100?1_2$	5	11	10	8	11	9	10	6
$0101?0_2$	9	8	9	7	8	6	6	5
$0101?1_2$	7	8	6	11	7	11	7	10
$0110?0_2$	6	6	10	7	9	5	10	9
$0110?1_2$	7	8	4	9	5	12	7	8
$0111?0_2$	13	6	11	7	8	6	7	7
$0111?1_2$	7	12	8	8	7	7	7	8
$1000?0_2$	5	6	9	10	6	11	10	7
$1000?1_2$	9	3	5	9	12	6	3	6
$1001?0_2$	7	8	7	11	6	8	9	8
$1001?1_2$	10	2	7	8	8	5	9	9
$1010?0_2$	6	8	11	8	6	12	5	13
$1010?1_2$	11	4	11	9	7	7	6	11
$1011?0_2$	9	7	8	7	10	6	9	8
$1011?1_2$	6	12	9	8	10	11	10	8
$1100?0_2$	9	13	7	8	8	7	11	8
$1100?1_2$	6	7	7	6	8	7	5	9
$1101?0_2$	11	7	9	7	6	10	4	6
$1101?1_2$	6	9	9	9	11	10	5	8
$1110?0_2$	13	10	10	6	11	6	5	9
$1110?1_2$	10	9	9	7	7	5	9	8
$1111?0_2$	8	10	6	11	6	9	10	3
$1111?1_2$	7	7	7	8	9	10	7	7

Table 8. $16 \times p_s(abcd?efg???h, 01110000)$ in S box of LOKI91