IEICE TRANS. FUNDAMENTALS, VOL.E86–A, NO.5 MAY 2003

PAPER *Special Section on Discrete Mathematics and Its Applications*

Statistical Analysis of χ^2-Attacks**

Norihisa ISOGAI[†a)], ***Nonmember***, **Atsuko MIYAJI**[†*b)], ***Regular Member***, ***and*** **Masao NONAKA**[††c)], ***Nonmember***

SUMMARY The χ^2-attack was originally proposed by Knudsen and Meier. This attack is one of the most effective attacks for RC6. The χ^2-attack can be used for both distinguishing attacks and for key recovery attacks. Although, up to the present, theoretical analysis of χ^2-attacks, especially the relation between a distinguishing attack and a key recovery attack, has not been discussed, the security against key recovery attacks has been often discussed by the results of distinguishing attacks. In this paper, we investigate the theoretical relation between the distinguishing attack and the key recovery attack, and prove one theorem to evaluate the exact security against the key recovery attacks by using the results of the distinguishing attack. Furthermore we propose two key recovery attacks against RC5-64 and implement them. Our best key recovery attack can analyze RC5-64 with 16 rounds by using $2^{125.23}$ plaintexts with a success probability of 30%. This result works faster than exhaustive key search. As far as the authors know, this is the best result of known plaintext attacks to RC5-64. We also apply our theory on our key recovery attacks and demonstrate the validity.

key words: *block cipher, statistical analysis, RC5, χ^2-attacks*

1. Introduction

The χ^2-attack [10] was originally proposed by Knudsen and Meier as a chosen plaintext attack to RC6 [15]. They focused on the correlations between input (plaintext) bits and output (ciphertext) bits, measured by the χ^2-test: the specific rotation in RC6 is considered to cause the correlations between the corresponding two 10-bit integer values. The χ^2-attack can be used for both distinguishing attacks and for key recovery attacks. Distinguishing attacks have only to handle plaintexts in such a way that the χ^2-value of a part of ciphertexts becomes significantly a higher value. On the other hand, key recovery attacks have to rule out all wrong keys, and single out exactly a correct key by using the χ^2-value. Therefore, key recovery attacks often require more work and memory than distinguishing attacks. In [10], their key recovery attack was estimated by using only the results of the distinguishing attack. Note that their key recovery attack on RC6 with any round was not implemented because it required too much memory even in the case of small number of rounds. In [16], a key recovery attack on RC5-32 [14] by using a χ^2-attack was proposed. RC5-$w/r/b$ means that two w-bit-word plaintexts are encrypted with r rounds by b-byte keys. A χ^2-attack to RC5-32 was further improved [13]. Their attack can analyze RC5-32 with 10 rounds by a known plaintext attack with negligible memory. They also pointed out the significant difference between the distinguishing attack and the key recovery attack. For the distinguishing attack, the high χ^2-value is necessary and sufficient condition. On the other hand, the high χ^2-value is not sufficient for the key recovery attack. In fact, the high χ^2-value algorithm cannot recover a correct key with high probability [16], but the rather low χ^2-value algorithm can recover a correct key with high probability [13]. This reasons that the security against the key recovery attack cannot be estimated directly from that against the distinguishing attack. However, up to the present, any theoretical difference between the distinguishing attack and the key recovery attack has not been discussed.

In this paper, we investigate the theoretical relation between the distinguishing attack and the key recovery attack, and prove one theorem to evaluate the exact security against the key recovery attack by using the results of the distinguishing attack. We also propose two key recovery algorithms against RC5-64. By using these key recovery attacks, we also demonstrate our theory. As we will see in Sects. 3 and 4, the implemented results of the key recovery attacks are almost the same as the results evaluated under our theorem by using the results of the distinguishing attack.

Table 1 presents our experimental results of the key recovery attack to RC5-64. Our known plaintext attack, Algorithm 3, can be mounted on RC5-64 with 16 rounds with negligible memory. Our attack can analyze RC5-64 more efficiently than the previous best attack [2]. In concluding, we should note that RC5-64 with 16 rounds is not secure against the known plaintext attack.

This paper is organized as follows. Section 2 sum-

Manuscript received August 26, 2002.
Manuscript revised November 14, 2002.
Final manuscript received January 10, 2003.

[†]The authors are with Japan Advanced Institute of Science and Technology, Ishikawa-ken, 923-1292 Japan.
[††]The author is with Matsushita Electric Industrial Co., LTD, Osaka-shi, 571-8501 Japan. This work was conducted when he was with JAIST.
[*]Presently, with University of California, Davis.
a) E-mail: n-isogai@jaist.ac.jp
b) E-mail: miyaji@jaist.ac.jp
c) E-mail: nonaka@isl.mei.co.jp

[**]A preliminary version was presented at SCIS'2002.

Table 1 Attack on RC5-64 (Implemented).

	2 rounds		3 rounds		4 rounds	
	#texts	#keys	#texts	#keys	#texts	#keys
Linear attack [2]	2^{17}	39/100	2^{25}	28/50	2^{34}	9/10
	2^{19}	96/100	2^{27}	47/50		
Algorithm 3 [This paper]	2^{16}	71/100	2^{24}	54/100	2^{33}	76/100
	2^{17}	99/100	2^{25}	93/100	2^{34}	98/100

marizes the notation, RC5-64 algorithm, the χ^2-test, and statistical facts used in this paper. Section 3 proposes the chosen plaintext attack and the known plaintext attack, Algorithms 2 and 3, against RC5-64, and demonstrates our statistical method by using the results of the distinguishing attack. The experimental results of our key recovery attacks are described in Sect. 4. Section 5 investigates the difference between Algorithms 2 and 3 from the statistical point of view. Conclusion is given in Sect. 6.

2. Preliminary

2.1 Notation

- $+$ ($\boxplus$): addition (addition mod 2^{64});
- $-$: subtraction;
- $*$: multiplication;
- $\oplus$: bit-wise exclusive OR;
- r (h): number of full(half) rounds ($h = 2r$);
- $a \lll b$: cyclic rotation of a to the left by b-bit;
- $a \ggg b$: cyclic rotation of a to the right by b-bit;
- (L_i, R_i): input of the i-th half-round, (L_0, R_0), (L_{h+1}, R_{h+1}) is a plaintext, a ciphertext after h half-rounds encryption, respectively;
- S_i: i-th subkey (S_{h+1} is a subkey of the h-th half-round);
- $lsb_n(X)$: least significant n-bit of X;
- X^i: i-th bit of X.

We denote the least significant bit (lsb) to the 1st bit, and the most significant bit (msb) as the 64-th bit for any 64-bit element.

2.2 Block Cipher RC5-64

Here, we introduce the encryption algorithm of RC5-64: a plaintext (L_0, R_0) is added with (S_0, S_1) and set to (L_1, R_1). (L_1, R_1) is encrypted to (L_{h+1}, R_{h+1}) by h half-rounds iterations of a main loop, which is called one half-round. Two consecutive half-rounds correspond to one round of RC5.

Algorithm 1 (RC5-64 Encryption Algorithm):

1. $L_1 = L_0 \boxplus S_0$; $R_1 = R_0 \boxplus S_1$;
2. for $i = 1$ to h do:
   $L_{i+1} = R_i$; $R_{i+1} = ((L_i \oplus R_i) \lll R_i) \boxplus S_{i+1}$.

2.3 χ^2-Test

We make use of the χ^2-tests for distinguishing a non-random distribution from random distribution [8], [10], [11]. Let $X = X_0, ..., X_{n-1}$ be sets of $\{a_0, ..., a_{m-1}\}$, and $N_{a_j}(X)$ be the number of X which takes on the value a_j. The χ^2-statistic of X which estimates the difference between X and the uniform distribution is defined as follows:

$$\chi^2(X) = \frac{m}{n} \sum_{i=0}^{m-1} \left(N_{a_i}(X) - \frac{n}{m} \right)^2 .$$

2.4 Statistical Facts

Here, we summarize statistical facts and the notation.

Theorem 1 (Distribution of the Means [4]): Let μ and σ^2 be the mean and the variance of a population, respectively. Then the mean and the variance of the distribution of the mean of a random sample with the size n drawn from the population are μ, and σ^2/n, respectively.

Theorem 2 (Central Limit Theorem [4]): Choose a random sample from a population, if the sample size n is large, the sampling distribution of the mean is closely approximated by the normal distribution, regardless of the population.

Theorem 3 (Law of large numbers [4]): The lar-ger the sample size, the more probable it is that the sample mean comes arbitrarily close to the population mean.

The probability density function of the normal distribution with the mean x_0 and the variance σ^2 is described by the following equation,

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x - x_0)^2}{2\sigma^2} \right].$$

3. Proposed χ^2-Attacks on RC5-64

This section presents two attacks, Algorithms 2 and 3, by applying the correlation attack on RC5-32 [13]. We also investigate the security of RC5-64 from the statistical point of view.

3.1 Attacks

Our proposed two algorithms recover the least significant five bits of S_{h+1}, which can be applied to any consecutive bits of S_{h+1}. In our algorithms, set $(x, y) = (lsb_6(L_{h+1}), lsb_6(R_{h+1}))$, and $s = lsb_5(S_{h+1})$, where x corresponds to the rotation amount in the h-th half-round. Each algorithm is as follows:

Algorithm 2 (Proposed chosen plaintext attack):
1. Choose a plaintext (L_0, R_0) with $lsb_6(R_0) = 0$, and encrypt it.
2. For each $s (s = 0, 1, \cdots, 2^5 - 1)$, set $S^6_{h+1} = 0$, and decrypt y by 1 half-round. Note that, y decrypted with the rotation amount x in the h-th half-round by 1 half-round is set to z, then we exactly know it.
3. For each value s, x, and z, we update each array by incrementing $count[s][x][z]$.
4. For each s and x, compute $\chi^2[s][x]$.
5. Compute the average $ave[s]$ for $\{\chi^2[s][x]\}_x$ of each s, and output s with the highest $ave[s]$ as $lsb_5(S_{h+1})$.

Algorithm 3 (Proposed known plaintext attack):
1. Given known plaintexts (L_0, R_0), set $l = lsb_6(R_0)$, and encrypt them.
2. For each l, compute $\chi^2[l][s][x]$ according to Steps 2-4 in Algorithm 2.
3. Compute the average $ave[s]$ of $\{\chi^2[l][s][x]\}_{l,x}$ for each s, and output s with the highest $ave[s]$ as $lsb_5(S_{h+1})$.

Here, we discuss the distributions of χ^2-values in Algorithms 2 and 3 from the statistical point of view. In Algorithm 2, we classify $\chi^2[s][x]$ into 2^6 types by the h-th half-round rotation x. On the other hand, in Algorithm 3, we classify $\chi^2[l][s][x]$ into 2^{12} types by the 1st and h-th half-round rotations l and x. Using the facts in Theorem 1, the more number of classification is done, the smaller the variance becomes. We also note that the numbers of available plaintexts in Algorithms 2 and 3 are 2^{122} and 2^{128}, respectively. In consecutive sections, we will see how these differences in the number of classification and that of available plaintexts influence the applicability against RC5-64.

3.2 Notation

We give the following notation.
(1) e and SUC: recovered-key bit size and success probability of a key recovery attack, respectively. There are one correct key and $2^e - 1$ wrong keys.
(2) X_c and X_w: distributions of χ^2-values of a key recovery attack by using a correct key and a wrong key, respectively.
(3) C_m and C_v: mean and variance of distribution of mean of χ^2-values of a key recovery attack with a correct key, respectively.
(4) $P_c(x)$: $(1/\sqrt{2\pi C_v}) \exp\left[-(x - C_m)^2/(2C_v)\right]$ (probability density function of distribution of χ^2-values with a correct key).
(5) W_m and W_v: mean and variance of distribution of χ^2-values in a key recovery attack with a wrong key, respectively.
(6) $P_w(x)$: $(1/\sqrt{2\pi W_v}) \exp\left[-(x - W_m)^2/(2W_v)\right]$ (probability density function of distribution of χ^2-values with a wrong key).
(7) $E_{m[h,n]}$ and $E_{v[h,n]}$: mean and variance of distribution of χ^2-values on $lsb_6(R_{h+1})$ of RC5-64 with $lsb_6(R_0) = 0$ by using 2^n plaintexts, respectively.
(8) $C_{m1[h,n]}$ ($C_{m2[h,n]}$) and $C_{v1[h,n]}$ ($C_{v2[h,n]}$): mean and variance of distribution of χ^2-values recovered with a correct key in $lsb_6(R_{h+1})$ by using 2^n plaintexts in Algorithm 2 (Algorithm 3), respectively.
(9) $W_{m1[h,n]}$ ($W_{m2[h,n]}$) and $W_{v1[h,n]}$ ($W_{v2[h,n]}$): mean and variance of distribution of χ^2-values recovered with a wrong key in $lsb_6(R_{h+1})$ by using 2^n plaintexts in Algorithm 2 (Algortihm 3), respectively.

3.3 Statistical Analysis of χ^2-Attacks

We show one theorem on the success probability of the key recovery attack. First, we put forward three hypotheses and prove three lemmas on the distributions of χ^2-values of the key recovery attacks. Note that the key recovery attacks compute the χ^2-value on a part by using all candidates and outputs a key with the highest χ^2-value as a correct key.

Hypothesis 1: For $i = 1, 2, \cdots, 2^e - 1$, let $X_{w(i)}$ be the distribution of χ^2-values on the part recovered with the i-th wrong key. Then we assume that the distributions $X_{w(1)}, X_{w(2)}, \cdots, X_{w(2^e-1)}$ are independent and approximately equal.

Lemma 1: If the number of χ^2-values is enough large, then the distribution of the mean of χ^2-values is approximately normally distributed.

Proof: This follows easily from Theorem 2. □

Lemma 2: Let $n \geq 12$ and $h \geq 4$. In Algorithm 2 (Algorithm 3), the mean of χ^2-values recovered with a correct key by using 2^n plaintexts, $C_{m1[h,n]}$ ($C_{m2[h,n]}$), is estimated by the mean of χ^2-values on $lsb_6(R_{h-1})$ by using 2^{n-6} (2^{n-12}) plaintexts, $E_{m[h-2,n-6]}$ ($E_{m[h-2,n-12]}$), as follows,

$$C_{m1[h,n]} = E_{m[h-2,n-6]} (C_{m2[h,n]} = E_{m[h-2,n-12]}).$$

Proof: If we use a correct key in Algorithms 2 and 3, the correct six-bit-data decrypted by 1 half-round can be exactly obtained, which is the same as that decrypted by one more half-round. As a result, $lsb_6(R_{h+1})$ can be decrypted by 1 round (2 half-rounds) as long as we use a correct key. Since $lsb_6(R_0)$ and the

rotation amount in the final half-round are uniformly distributed, the χ^2-values in Algorithms 2 and 3 are estimated to be computed by using 2^{n-6} and 2^{n-12} plaintexts, respectively. Thus, by using Theorem 1, we get $C_{m1[h,n]} = E_{m[h-2,n-6]}$ and $C_{m2[h,n]} = E_{m[h-2,n-12]}$.
□

Lemma 3: Let $n \geq 12$ and $h \geq 4$. In Algorithm 2 (Algorithm 3), the variance of χ^2-values recovered with a correct key by using 2^n plaintexts, $C_{v1[h,n]}$ ($C_{v2[h,n]}$), is estimated by the variance of χ^2-values on $lsb_6(R_{h-1})$ by using 2^{n-6} (2^{n-12}) plaintexts, $E_{v[h-2,n-6]}$ ($E_{v[h-2,n-12]}$), as follows,

$$C_{v1[h,n]} = \frac{E_{v[h-2,n-6]}}{2^6}\left(C_{v2[h,n]} = \frac{E_{v[h-2,n-12]}}{2^{12}}\right).$$

Proof: In Algorithm 2, we compute the χ^2-values every 2^6 $lsb_6(L_{h+1})$ and compute the mean for a key. In Algorithm 3, we compute the χ^2-values every 2^{12} $(lsb_6(R_0), lsb_6(L_{h+1}))$ and compute the mean for a key. Therefore, by using Theorem 1 and Theorem 3, we get $C_{v1[h,n]} = E_{v[h-2,n-6]}/2^6$ and $C_{v2[h,n]} = E_{v[h-2,n-12]}/2^{12}$ in Algorithms 2 and 3, respectively.
□

Hypothesis 2: Let $n \geq 12$ and $h \geq 4$. We assume that the mean of χ^2-values with a wrong key by using 2^n plaintexts, $W_{m1[h,n]}$ ($W_{m2[h,n]}$), is approximately equal to that of χ^2-values on $lsb_6(R_{h+1})$ by using 2^{n-6} (2^{n-12}) plaintexts, $E_{m[h,n-6]}$ ($E_{m[h,n-12]}$) in Algorithm 2 (Algorithm 3), as follows,

$$W_{m1[h,n]} = E_{m[h,n-6]}\ (W_{m2[h,n]} = E_{m[h,n-12]}).$$

Hypothesis 3: Let $n \geq 12$ and $h \geq 4$. The variance of χ^2-values with a wrong key by using 2^n plaintexts, $W_{v1[h,n]}$ ($W_{v2[h,n]}$), is approximately equal to that of χ^2-values on $lsb_6(R_{h+1})$ by using 2^{n-6} (2^{n-12}) plaintexts, $E_{v[h,n-6]}$ ($E_{v[h,n-12]}$) in Algorithm 2 (Algorithm 3), as follows,

$$W_{v1[h,n]} = \frac{E_{v[h,n-6]}}{2^6}\ \left(W_{v2[h,n]} = \frac{E_{v[h,n-12]}}{2^{12}}\right).$$

Using the above preparations, the success probability of the key recovery attack is introduced as follows.

Theorem 4: The success probability SUC of e-bit key recovery can be evaluated by using $P_c(x)$ and $P_w(x)$ as follows,

$$SUC = \int_{-\infty}^{\infty} P_c(x) * \left(\int_{-\infty}^{x} P_w(u)du\right)^{2^e-1} dx.$$

Proof: The e-bit key can be recovered correctly if and only if the χ^2-value of a correct key is higher than that of all $2^e - 1$ wrong keys. Thus the conclusion follows.
□

Theorem 4 introduces the following two factors for high success probability.

- **(Factor 1)** Maximize the average of χ^2-values computed by a correct key;
- **(Factor 2)** Minimize the variances (the error) of each distribution of χ^2-values computed by each key.

The previous researches of χ^2-attacks have focused on only Factor 1. If there is no difference between the mean of χ^2-values recovered with a correct key and that of χ^2-values recovered with a wrong key, we cannot single out a correct key. As a result, we need the small variance to rule out all wrong keys. A correct key can be selected well by using the rather low χ^2-value as long as the variance is small. We will demonstrate our theorem by using Algorithms 2 and 3 in Sect. 3.5.

3.4 The Distribution of χ^2-Values on the Ciphertext

To evaluate the success probability, we conduct the following experiments on 4–8 half-rounds and get the distribution of χ^2-values on $lsb_6(R_{h+1})$. Note that the following tests use 100 kinds of plaintexts and 100 keys and thus conduct 10000 trials in total.

Distinguishing Test: The χ^2-test on $lsb_6(R_{h+1})$ with $lsb_6(R_0) = 0$.

By using the experimental results in Table 2, we compute the slope, that is, how many plaintexts are required to obtain the same χ^2-value. We set the χ^2-value to 82.53 (level = 0.95). By using the least squares method on the results of 4–7 half-rounds, the slope is computed to 4.11.

3.5 Theoretical Results on Proposed Attacks

We evaluate the security of RC5-64 with 6–8 half-rounds against Algorithms 2 and 3 from the following points of view: One is the estimation of the cost necessary to implement Algorithms 2 and 3. We estimate these algorithms by the number of plaintexts and the work amount. The other is the real cost necessary to obtain the estimation. The results are shown in Table 3.

First we investigate the estimation. By using the slope in Sect. 3.4, the numbers of plaintexts required for recovering a key in h half-rounds with a success probability of 90%, $\log_2(\#texts)$, are estimated to

$$\log_2(\#texts) = 4.11h - 1.77 \text{ (Algorithm 2) and}$$
$$\log_2(\#texts) = 4.11h + 1.23 \text{ (Algorithm 3).}$$

By substituting the numbers of available plaintexts, 2^{122} and 2^{128}, Algorithms 2 and 3 can analyze RC5-64 with 30 half-rounds by using $2^{121.53}$ and $2^{124.53}$ plaintexts with a success probability of 90%, respectively. Let us discuss the amount of work. We set one unit of work as one encryption. For each plaintext both Algorithms encrypt a plaintext, and decrypt a ciphertext by 1 half-round with each candidate key. Therefore, we set the amount of work to $\#texts \times (1 + 1/h \times 2^5)$. Thus

Table 2 The mean and the variance of χ^2-values on $lsb_6(R_{h+1})$ in 4–8 half-rounds (in 10000 trials).

Mean (Variance)					
#texts	4 half-rounds	5 half-rounds	6 half-rounds	7 half-rounds	8 half-rounds
2^{12}	63.41 (125.83)	–	63.02 (126.76)	–	–
2^{13}	63.40 (130.56)	–	62.91 (125.31)	–	–
2^{14}	64.03 (132.04)	–	62.96 (125.09)	–	–
2^{15}	64.80 (139.10)	–	62.91 (123.37)	–	–
2^{16}	66.63 (149.53)	63.20 (127.04)	62.97 (125.71)	63.01 (123.83)	–
2^{17}	70.02 (178.01)	63.51 (125.86)	62.83 (124.80)	63.17 (123.71)	–
2^{18}	–	64.08 (128.57)	–	63.00 (121.85)	–
2^{19}	–	65.33 (131.83)	–	62.85 (122.15)	–
2^{20}	–	67.68 (143.03)	63.09 (124.31)	63.07 (124.79)	62.84 (123.69)
2^{21}	–	71.99 (157.10)	63.41 (126.33)	63.12 (127.31)	62.99 (123.79)
2^{22}	–	80.79 (192.96)	63.68 (125.68)	63.26 (125.09)	63.01 (122.39)
2^{23}	–	–	64.39 (130.16)	–	63.18 (124.95)
2^{24}	–	–	66.11 (135.60)	–	63.06 (122.94)
2^{25}	–	–	69.22 (152.86)	–	63.05 (124.57)
2^{26}	–	–	75.43 (174.06)	–	63.23 (124.58)
$2^{18.1}$	80.70 (186.60)	–	–	–	–
$2^{18.2}$	82.85 (190.63)	–	–	–	–
$2^{22.1}$	–	82.12 (195.22)	–	–	–
$2^{22.2}$	–	83.42 (205.32)	–	–	–
$2^{26.5}$	–	–	82.39 (183.09)	–	–
$2^{26.6}$	–	–	83.45 (190.06)	–	–
$2^{30.3}$	–	–	–	81.80 (188.68)	–
$2^{30.4}$	–	–	–	83.28 (199.99)	–

Table 3 Theoretical results of our key recovery algorithms.

6 half-rounds: Mean (Variance)							
Algorithm 2				Algorithm 3			
#texts	correct	wrong	SUC	#texts	correct	wrong	SUC
2^{18}	63.41 (1.97)	63.02 (1.98)	0.053	2^{24}	63.41 (0.03)	63.02 (0.03)	0.559
2^{19}	63.40 (2.04)	62.91 (1.96)	0.065	2^{25}	63.40 (0.03)	62.91 (0.03)	0.742
2^{20}	64.03 (2.06)	62.96 (1.95)	0.128	2^{26}	64.03 (0.03)	62.96 (0.03)	0.999
2^{21}	64.80 (2.17)	62.91 (1.93)	0.277	2^{27}	64.80 (0.03)	62.91 (0.03)	1.000
2^{22}	66.63 (2.34)	62.97 (1.96)	0.683	2^{28}	66.63 (0.04)	62.97 (0.03)	1.000
2^{23}	70.02 (2.78)	62.83 (1.95)	0.991	2^{29}	70.02 (0.04)	62.83 (0.03)	1.000
7 half-rounds: Mean (Variance)							
Algorithm 2				Algorithm 3			
#texts	correct	wrong	SUC	#texts	correct	wrong	SUC
2^{22}	63.20 (1.98)	63.01 (1.93)	0.042	2^{28}	63.20 (0.03)	63.01 (0.03)	0.198
2^{23}	63.51 (1.96)	63.17 (1.93)	0.052	2^{29}	63.51 (0.03)	63.17 (0.03)	0.470
2^{24}	64.08 (2.00)	63.00 (1.90)	0.132	2^{30}	64.08 (0.03)	63.00 (0.03)	0.999
2^{25}	65.33 (2.06)	62.95 (1.91)	0.412	2^{31}	65.33 (0.03)	62.95 (0.03)	1.000
2^{26}	67.58 (2.23)	63.07 (1.94)	0.840	2^{32}	67.68 (0.04)	63.07 (0.03)	1.000
2^{27}	71.99 (2.45)	63.12 (1.98)	0.999	2^{33}	71.99 (0.04)	63.12 (0.03)	1.000
2^{28}	80.79 (3.01)	63.26 (1.95)	1.000	2^{34}	80.79 (0.05)	63.26 (0.03)	1.000
8 half-rounds: Mean (Variance)							
Algorithm 2				Algorithm 3			
#texts	correct	wrong	SUC	#texts	correct	wrong	SUC
2^{26}	63.09 (1.93)	62.84 (1.93)	0.045	2^{32}	63.09 (0.03)	62.84 (0.03)	0.293
2^{27}	63.41 (1.97)	62.99 (1.93)	0.058	2^{33}	63.41 (0.03)	62.99 (0.03)	0.631
2^{28}	64.68 (1.96)	63.01 (1.91)	0.080	2^{34}	64.68 (0.03)	63.01 (0.03)	0.944
2^{29}	64.39 (2.03)	63.18 (1.95)	0.146	2^{35}	64.39 (0.03)	63.18 (0.03)	1.000
2^{30}	66.11 (2.12)	63.06 (1.92)	0.552	2^{36}	66.11 (0.03)	63.06 (0.03)	1.000
2^{31}	69.22 (2.39)	63.05 (1.95)	0.973	2^{37}	69.22 (0.04)	63.05 (0.03)	1.000
2^{32}	75.43 (2.72)	63.23 (1.95)	1.000	2^{38}	75.43 (0.04)	63.23 (0.03)	1.000

Table 4 Implemented results of our proposed key recovery algorithms (in 100 trials).

	5 half-rounds		6 half-rounds		7 half-rounds		8 half-rounds	
	#texts	#keys	#texts	#keys	#texts	#keys	#texts	#keys
Algorithm 2	2^{16}	13	2^{20}	15	2^{24}	14	2^{28}	9
	2^{17}	39	2^{21}	36	2^{25}	23	2^{30}	53
	2^{18}	81	2^{22}	62	2^{27}	81	2^{31}	89
	2^{19}	99	2^{23}	92	2^{28}	100	2^{32}	100
Algorithm 3	2^{19}	12	2^{23}	22	2^{27}	21	2^{31}	23
	2^{20}	31	2^{24}	54	2^{28}	40	2^{32}	43
	2^{21}	89	2^{25}	93	2^{29}	78	2^{33}	76
	2^{22}	99	2^{26}	99	2^{30}	99	2^{34}	98

Table 5 Implemented results: #plaintexts required for recovering a key with the success probability of 90%, 50%, or 30% (in 100 trials).

	#half-rounds	5	6	7	8
Algorithm 2	#texts (90%)	$2^{18.4}$	$2^{23.0}$	$2^{27.5}$	$2^{31.4}$
	#texts (50%)	$2^{17.4}$	$2^{21.6}$	$2^{26.0}$	$2^{30.0}$
Algorithm 3	#texts (90%)	$2^{21.1}$	$2^{24.9}$	$2^{29.5}$	$2^{33.5}$
	#texts (50%)	$2^{20.3}$	$2^{23.9}$	$2^{28.5}$	$2^{32.4}$
	#texts (30%)	$2^{19.9}$	$2^{23.3}$	$2^{27.3}$	$2^{31.6}$

the amounts of work necessary to attack RC5-64 with 30 half-rounds in Algorithms 2 and 3 are $2^{122.58}$ and $2^{125.58}$, respectively.

Next we investigate the real cost necessary for the above estimation. By using Lemma 2 and Hypothesis 2, the numbers of plaintexts required for conjecturing the results of the key recovery attacks in h half-rounds with a success probability of 90%, $\log_2(\#texts)$, are estimated to

$$\log_2(\#texts) = 4.11h - 7.77 \text{ (Algorithm 2) and}$$
$$\log_2(\#texts) = 4.11h - 10.77 \text{ (Algorithm 3).}$$

Therefore, our theory can estimate the security of RC5-64 with 30 half-rounds against Algorithms 2 (Algorithm 3) by using $2^{115.53}$ ($2^{112.53}$) plaintexts with a success probability of 90%. Let us discuss the amount of work. We set one unit of work as one encryption in the same way as the above case. In our estimation, we need one h half-rounds encryption every text and trial, and thus we set the amount of work to $\#texts \times \#trials$. The above equation indicates that the amount of work also depends on the number of trials. In our distinguishing algorithms, we set $\#trials$ to 10^4 to obtain more precise results. However, in fact, 10^3 trails are enough to compute the theoretical results. Here, we estimate the amount of work by setting $\#trials$ to 10^3. Thus the amounts of work necessary to obtain the above estimation in Algorithms 2 and 3 are $2^{125.50}$ and $2^{122.50}$, respectively.

4. Comparison of Experimental Results and Theoretical Results

This section shows the experimental results of RC5-64 against Algorithms 2 and 3 and compares it to the estimation by Theorem 4.

4.1 Experimental Results of Algorithms 2 and 3

In our experiments, all plaintexts are generated by using M-sequence. For example, Algorithms 2 and 3 use 122-bit and 128-bit random numbers generated by M-sequence, whose primitive polynomials of M-sequence are $x^{122} + x^{108} + x^8 + x + 1$ and $x^{128} + x^7 + x^2 + x + 1$, respectively. The platform is IBM RS/6000 SP (PPC 604e/332 MHz × 256) with memory of 32 GB.

Table 4 shows the experimental results among 100 trials for RC5-64 with 5–8 half-rounds. More detailed experimental results are shown in Table 5. In Algorithm 2, the number of plaintexts required for recovering a key in RC5-64 with h half-rounds with a success probability of 90% or 50%, $\log_2(\#texts)$, is estimated to

$$\log_2(\#texts) = 4.35h - 3.20 \text{ (90\%) or}$$
$$\log_2(\#texts) = 4.22h - 3.68 \text{ (50\%),}$$

respectively by using the least squares method.

In Algorithm 3, the number of plaintexts required for recovering a key in RC5-64 with h half-rounds with a success probability of 90%, 50%, or 30%, $\log_2(\#texts)$, is estimated to

$$\log_2(\#texts) = 4.18h + 0.08 \text{ (90\%),}$$
$$\log_2(\#texts) = 4.09h - 0.31 \text{ (50\%), or}$$
$$\log_2(\#texts) = 3.91h + 0.11 \text{ (30\%),}$$

respectively by using the least squares method. By using above linear equations, we estimate the required numbers of plaintexts for the higher round of RC5-64 shown in Table 6.

Table 6 #plaintexts required for recovering a key with the success probability of 90%, 50%, or 30% (from implemented results).

	linear equation	#half-rounds							
		4	6	8	28	29	30	31	32
Algorithm 2	$4.35h - 3.20$ (90%)	14.20	22.90	31.60	118.60	-	-	-	-
	$4.22h - 3.68$ (50%)	13.20	21.64	30.08	114.48	118.70	-	-	-
Algorithm 3	$4.18h + 0.08$ (90%)	16.80	25.16	33.52	117.12	121.30	125.48	-	-
	$4.09h - 0.31$ (50%)	16.05	24.23	32.41	114.21	118.30	122.39	-	-
	$3.91h + 0.11$ (30%)	15.75	23.57	31.39	109.59	113.50	117.41	121.32	125.23

Table 7 Comparison of the implemented results and the theoretical results with the success probability of 90%.

		Algorithm 2			Algorithm 3		
		#texts	work	#half-rounds	#texts	work	#half-rounds
Implemented results		$2^{118.60}$	$2^{119.70}$	28	$2^{125.48}$	$2^{126.53}$	30
Theoretical results	Estimation	$2^{121.53}$	$2^{122.58}$	30	$2^{124.53}$	$2^{125.58}$	30
	Real cost	$2^{115.53}$	$2^{125.50}$	30	$2^{112.53}$	$2^{122.50}$	30

We investigate the amount of work. In the same way as Sect. 3.5, we set one unit of work as one encryption. By substituting the numbers of available plaintexts, 2^{122} and 2^{128}, Algorithms 2 and 3 can analyze RC5-64 with 28 and 30 half-rounds by using $2^{118.60}$ and $2^{125.48}$ plaintexts with a success probability of 90%, respectively. For each plaintext both Algorithms encrypt a plaintext, and decrypt a ciphertext by 1 half-round with each candidate key. Therefore, we set the amount of work to $\#texts \times (1 + 1/h \times 2^5)$. Thus the amounts of work necessary to attack RC5-64 with 28 and 30 half-rounds in Algorithms 2 and 3 are $2^{119.70}$ and $2^{126.53}$, respectively. Additionally, by using $2^{125.23}$ plaintexts with a success probability of 30%, RC5-64 with 32 half-rounds can be analyzed faster than exhaustive key search.

4.2 Comparison

We compare the above implemented results with the theoretical results in Sect. 3 from the points of view of the cost, the number of plaintexts and the amount of work. As for the implemented results, Table 7 shows each cost necessary to recover a key of RC5-64 with the success probability of 90% in Algorithms 2 and 3. These are obtained by using implemented results in Table 6. As for the theoretical results, Table 7 shows each cost from the following two points of view: One is the estimation of the cost necessary to implement the key recovery attack with the success probability of 90%. The other is the real cost necessary to obtain the estimation. Table 7 indicates that our theory can estimate the security of key recovery attack well. We see that our theory reduces the number of plaintexts necessary for estimation. Furthermore, in Algorithm 3, we can reduce the amount of work necessary to evaluate the security of RC5-64 by using the proposed statistical method.

5. Further Discussion

In this section, we investigate the difference between Algorithms 2 and 3 from the statistical point of view. In our case, Algorithm 3 can recover a key better than Algorithm 2. As for the mean of χ^2-values, Algorithm 3 requires 2^6 times as many texts as Algorithm 2 to get the same χ^2-value. On the other hand, the variance of Algorithm 3 is about 1/64 of that of Algorithm 2. This reflects the statistical facts: Algorithm 2 measures the χ^2-value for each of $lsb_6(L_{h+1})$. Algorithm 3 measures the χ^2-value for each $lsb_6(R_0)$ and $lsb_6(L_{h+1})$. By using Lemma 3 and Hypothesis 3, the variance of Algorithm 3 is about 1/64 of that of Algorithm 2. As we noted two factors in Sect. 3, the low variance is necessary to single out a correct key. Furthermore, the numbers of available plaintexts in Algorithms 2 and 3 are 2^{122} and 2^{128}, respectively, and thus Algorithm 3 can use 2^6 times as many texts as Algorithm 2. As a result, Algorithm 3 is more efficient than Algorithm 2. Note that it is coincident with the results introduced by our theorem in Sect. 3.

6. Conclusion

In this paper, we have proved a theory to evaluate the security against the key recovery attacks by using the results of the distinguishing attack. Here, we have also proposed two key recovery algorithms against RC5-64. Algorithm 3 can analyze RC5-64 with h half-rounds by using $2^{4.18h+0.08}$ and $2^{3.91h+0.11}$ plaintexts with the success probabilities of 90% and 30%, respectively. Therefore, Algorithm 3 can analyze RC5-64 with 30 and 32 half-rounds by using $2^{125.48}$ and $2^{125.23}$ plaintexts, respectively. By comparing the implemented results with

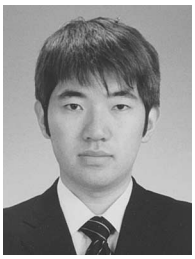
the theoretical results by our theory, we have demonstrated that our theory can estimate the security of the key recovery attack by using only the results of the distinguishing attack. Furthermore, our theory reduces surprisingly the number of plaintexts necessary for evaluation. In fact, in the case of RC5-64 with 30 half-rounds in the success probability of 90%, the best Algorithm 3 requires $2^{125.48}$ plaintexts, but our theory uses only $2^{112.53}$ plaintexts.

Acknowledgments

The authors wish to thank the anonymous referees for invaluable comments.

References

[1] A. Biryukov and E. Kushilevitz, "Improved cryptanalysis of RC5," Proc. EUROCRYPT'98, LNCS 1403, pp.85–99, 1998.
[2] J. Borst, B. Preneel, and J. Vandewalle, "Linear cryptanalysis of RC5 and RC6," Proc. Fast Software Encryption'99, LNCS 1636, pp.16–30, 1999.
[3] S. Contini, R. Rivest, M. Robshaw, and Y. Yin, "Improved analysis of some simplified variants of RC6," Proc. Fast Software Encryption'99, LNCS 1636, pp.1–15, 1999.
[4] R.J. Freund and W.J. Wilson, Statistical Method, Academic Press, San Diego, 1993.
[5] H. Gilbert, H. Handschuh, A. Joux, and S. Vaudenay, "A statistical attack on RC6," Proc. Fast Software Encryption'2000, LNCS 1978, pp.64–74, 2000.
[6] B. Kaliski and Y. Lin, "On differential and linear cryptanalysis of the RC5 encryption algorithm," Proc. CRYPTO'95, LNCS 963, pp.171–184, 1995.
[7] H. Kashiwagi, M-sequence and its applications, Shokodo, 1996.
[8] J. Kelsey, B. Schneier, and D. Wagner, "Mod n cryptanalysis, with applications against RC5P and M6," Proc. Fast Software Encryption'99, LNCS 1636, pp.139–155, 1999.
[9] L. Knudsen and W. Meier, "Improved differential attacks on RC5," Proc. CRYPTO'96, LNCS 1109, pp.216–228, 1996.
[10] L. Knudsen and W. Meier, "Correlations in RC6 with a reduced number of rounds," Proc. Fast Software Encryption'2000, LNCS 1978, pp.94–108, 2000.
[11] D. Knuth, The art of computer programming, vol.2, Seminumerical Algorithms, 2nd ed., Addison-Wesley, Reading, Mass., 1981.
[12] A. Menezes, P.C. van Oorschot, and S. Vanstone, Handbook of applied cryptography, CRC Press, Boca Raton, 1996.
[13] A. Miyaji, M. Nonaka, and Y. Takii, "Known plaintext correlation attack against RC5," Proc. RSA'2002 Conf., LNCS 2271, pp.131–148, 2002.
[14] R. Rivest, "The RC5 encryption algorithm," Proc. Fast Software Encryption'95, LNCS 1008, pp.86–96, 1995.
[15] R. Rivest, M. Robshaw, R. Sidney, and Y. Yin, "The RC6 Block Cipher. v1.1," 1998.
[16] T. Shimoyama, K. Takeuchi, and J. Hayakawa, "Correlation attack to the block cipher RC5 and the simplified variants of RC6," Third AES Candidate Conf., April 2000.
[17] S. Shirohata, An introduction of statistical analysis, Kyoritsu Shuppan, 1992.



Norihisa Isogai received the B.E. degree in Electrical and Computer Engineering from Kanazawa University, Japan, in 2001. He has joined Japan Advanced Institute of Science and Technology (JAIST) since 2001. He is currently a master student at School of Information Science. His main research topic is cryptology, especially cryptanalysis to symmetric key cryptosystem.

Atsuko Miyaji received her B.Sc., M.Sc., and Dr.Sci. degrees in mathematics from Osaka University, Osaka, Japan in 1988, 1990, and 1997, respectively. She was with Matsushita Electric Industrial Co., LTD from 1990 to 1998, where she engaged in research and development of secure communications. She has been an associate professor at JAIST (Japan Advanced Institute of Science and Technology) since 1998, and with the computer science department of University of California, Davis since 2002. Her research interests include the application of projective varieties theory into cryptography and information security. She received the IPSJ Sakai Special Researcher Award in 2002. She is a member of the International Association for Cryptologic Research, the Institute of Electronics, Information and Communication Engineers and the Information Processing Society of Japan.

Masao Nonaka received his B.Sc. degree in computer science and engineering from University of Aizu, and M. Info. Sci. degree from Japan Advanced Institute of Science and Technology in 2000 and 2002, respectively. He joined Matsushita Electric Industrial Co., LTD. in 2002 and is engaged in research and development in the field of information security systems. He is a member of the Information Processing Society of Japan.