

New Weakness in the Key-Scheduling Algorithm of RC4

Toshihiro OHIGASHI^{†a)}, Student Member, Yoshiaki SHIRAIISHI^{††}, and Masakatu MORII^{†††}, Members

SUMMARY In a key scheduling algorithm (KSA) of stream ciphers, a secret key is expanded into a large initial state. An internal state reconstruction method is known as a general attack against stream ciphers; it recovers the initial state from a given pair of plaintext and ciphertext more efficiently than exhaustive key search. If the method succeeds, then it is desirable that the inverse of KSA is infeasible in order to avoid the leakage of the secret key information. This paper shows that it is easy to compute a secret key from an initial state of RC4. We propose a method to recover an ℓ -bit secret key from only the first ℓ bits of the initial state of RC4 using linear equations with the time complexity less than that of one execution of KSA. It can recover the secret keys of which number is $2^{103.6}$ when the size of the secret key is 128 bits. That is, the 128-bit secret key can be recovered with a high probability when the first 128 bits of the initial state are determined using the internal state reconstruction method.

key words: cryptanalysis, stream cipher, RC4, key scheduling algorithm

1. Introduction

Stream ciphers [1], [2] are primitives for providing confidentiality and integrity, and they are used in both software and hardware. A typical stream cipher encrypts a plaintext by XORing it with a pseudo-random sequence (called a keystream) that is generated from the secret key. The core of the stream cipher generates the keystream from the secret key. It consists of a key scheduling algorithm (KSA) and a pseudo-random generation algorithm (PRGA). KSA initializes the internal state using a short secret key. PRGA generates a keystream from the initial state, which is the internal state when KSA was completed.

The goal of key recovery attacks against stream ciphers is to recover a secret key from a keystream. However, since a stream cipher consists of two algorithms, namely, KSA and PRGA, attacks can be split into two steps: an internal state reconstruction method and a key reconstruction method. The former is a method to recover the initial state from a given keystream, and the latter is a method to recover the secret key from the initial state. Internal state reconstruction methods against several stream ciphers (e.g., SOBER family [3], [4], SNOW 1.0 [5], AA5 [6], SOSE-

MANUK [7], [8], and Polar Bear [9]) have been shown.

We focus on the key reconstruction method. One may think that if an internal state reconstruction method succeeds, then it is unnecessary to recover a secret key because the keystream is completely reconstructed from the obtained initial state. Notice that stream ciphers usually encrypt a plaintext with not only a secret key but also an initialization vector (IV), and the internal state depends on the IV. In order to decipher a ciphertext that is generated with a distinct IV, it is necessary to know the secret key. However, the key reconstruction method is impossible if it is infeasible to compute the secret key from the initial state. Hence, it is desirable that the inverse of KSA is infeasible.

This paper shows that it is easy to compute a secret key from an initial state of RC4 [10]. First, we give linear equations that allow to recover one byte of a secret key from one or two bytes of the initial state with a non-negligible probability. Next, we propose a method to recover an ℓ -bit secret key from the first ℓ bits of an initial state with the time complexity less than that of one execution of KSA. In RC4, a typical value of ℓ is 128. We show that the number of 128-bit secret keys that are recoverable by the proposed method is $2^{103.6}$. This implies that a 128-bit secret key can be recovered with a high probability when the first 128 bits of the initial state of RC4 are determined using the internal state reconstruction method.

The security of Wired Equivalent Privacy (WEP) [11], which is based on RC4, has been extensively studied. The previous works [12]–[21] concluded that WEP is not secure. However, it does not mean that RC4 is insecure. Namely, the protocol vulnerability causes insecurity of WEP. In fact, no effective attack for threatening RC4 itself has been shown so far.

This paper is organized as follows: RC4 is described in Sect. 2. In Sect. 3, we define invertible keys and non-expanded invertible keys. These keys can be easily recovered from an initial state. In Sect. 4, we give the linear equations that present the relationship between one byte of a secret key and one or two bytes of the initial-state of RC4 with non-negligible probability. Additionally, we propose a key reconstruction method with the linear equations. Finally, we show the lower bound on the number of non-expanded invertible keys of RC4.

Manuscript received March 8, 2007.

Manuscript revised July 13, 2007.

[†]The author is with the Graduate School of Science and Technology, Kobe University, Kobe-shi, 657-8501 Japan.

^{††}The author is with the Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan.

^{†††}The author is with the Faculty of Engineering, Kobe University, Kobe-shi, 657-8501 Japan.

a) E-mail: ohigashi@stu.kobe-u.ac.jp

DOI: 10.1093/ietfec/e91-a.1.3

2. RC4

2.1 Description of RC4

RC4 is a stream cipher designed by R. Rivest in 1987, and it is used in many commercial products and standards; for example, Secure Sockets Layer (SSL) 3.0/Transport Layer Security (TLS) 1.0 [22], WEP, Wi-Fi Protected Access (WPA) [23]. In RC4, the size of a secret key varies from 8 bits to 2048 bits. In KSA, the internal state is shuffled using the secret key. The shuffled internal state of KSA is set to be the initial internal state of PRGA. This initial internal state is called an initial state, for simplicity. In PRGA, a keystream is generated from the initial state, and it is XOR-ed with a plaintext to obtain the ciphertext. Figure 1 shows the processes of RC4.

We describe KSA and PRGA as given in [10], [24]. In order to distinguish between KSA and PRGA, we use different symbols. The internal state and two pointers of KSA are denoted by (S^*, i^*, j^*) , and those of PRGA are denoted by (S, i, j) .

(1) Key Scheduling Algorithm

The internal state of KSA at time t consists of a permutation table $S_t^* = (S_t^*[0], S_t^*[1], \dots, S_t^*[N-1])$, where $N = 2^n$. The element $S_t^*[x]$ is an n -bit variable, where $x \in \{0, 1, \dots, N-1\}$ is the index of the element of an internal state. Usually, $n = 8$ and $N = 256$. Two pointers at time t are denoted by i_t^* and j_t^* ; these are n -bit variables. Those parameters are initialized as follows:

$$S_0^*[x] = x \text{ for } \forall x \in \{0, 1, \dots, N-1\}, \quad (1)$$

$$i_0^* = j_0^* = 0. \quad (2)$$

An ℓ -bit secret key is set into a table $K = (K[0], K[1], \dots, K[L-1])$, where $L = \lceil \ell/n \rceil$. The element $K[y]$ is an n -bit variable, where $y \in \{0, 1, \dots, L-1\}$ is the index of the element of a secret key. The recommended parameter is $\ell = 128$ (when $n = 8$, $L = 16$). An initial state S_0 ($= S_N^*$) is computed from Eqs. (3)–(5) for $t = 1, 2, \dots, N$.

$$j_t^* = (j_{t-1}^* + S_{t-1}^*[i_{t-1}^*] + K[i_{t-1}^* \bmod L]) \bmod N, \quad (3)$$

$$\begin{cases} S_t^*[i_{t-1}^*] = S_{t-1}^*[j_t^*], \\ S_t^*[j_t^*] = S_{t-1}^*[i_{t-1}^*], \\ S_t^*[i_t^*] = S_{t-1}^*[i_t^*] \end{cases} \text{ for } \forall i^* \in \{0, 1, \dots, N-1\} \setminus \{i_{t-1}^*, j_t^*\}, \quad (4)$$

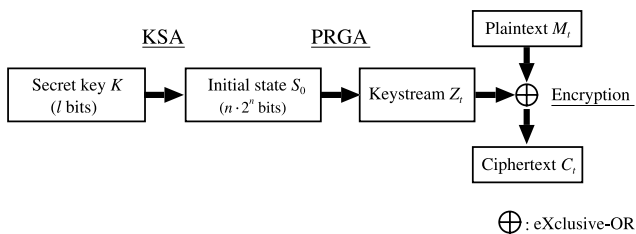


Fig. 1 Processes of RC4.

$$i_t^* = (i_{t-1}^* + 1) \bmod N. \quad (5)$$

Since $i_0^* = 0$ and KSA is processed at $t = 1, 2, \dots, N$, Eq. (5) is equivalent to

$$i_t^* = t. \quad (6)$$

Updating of j_t^* depends on $S_{t-1}^*[i_{t-1}^*]$ and $K[i_{t-1}^* \bmod L]$. When t is large, $S_{t-1}^*[i_{t-1}^*]$ is randomized by the swap operation of Eq. (4), and $K[i_{t-1}^* \bmod L]$ contributes to the randomization of j_t^* when t is small. Hence, we assume that j_t^* is uniformly distributed.

(2) Pseudo-Random Generation Algorithm

The internal state of PRGA at time t consists of a permutation table $S_t = (S_t[0], S_t[1], \dots, S_t[N-1])$. The element $S_t[x]$ is an n -bit variable. Two pointers at time t are denoted by i_t and j_t ; these are n -bit variables. Those parameters are initialized as follows:

$$S_0[x] = S_N^*[x] \text{ for } \forall x \in \{0, 1, \dots, N-1\}, \quad (7)$$

$$i_0 = j_0 = 0. \quad (8)$$

Let Z_t be an n -bit output word from PRGA at time t . The next-state and output functions for every $t \geq 1$ are defined as follows:

$$i_t = (i_{t-1} + 1) \bmod N, \quad (9)$$

$$j_t = (j_{t-1} + S_{t-1}[i_t]) \bmod N, \quad (10)$$

$$\begin{cases} S_t[i_t] = S_{t-1}[j_t], \\ S_t[j_t] = S_{t-1}[i_t], \\ S_t[i] = S_{t-1}[i] \text{ for } \forall i \in \{0, 1, \dots, N-1\} \setminus \{i_t, j_t\}, \end{cases} \quad (11)$$

$$Z_t = S_t[(S_t[i_t] + S_t[j_t]) \bmod N]. \quad (12)$$

2.2 Related Work

There are many security analyses for RC4. These are classified into (1) analysis of a keystream and (2) analysis of the initialization process (that is, KSA). Our study belongs to (2).

(1) Analysis of Keystream

A distinguishing attack is a method to distinguish a keystream from a truly random sequence. J. Dj. Golić proposed the distinguishing attack based on the correlation between the least significant bit of Z_t and that of Z_{t+2} [25]. The attack can distinguish a keystream from a truly random sequence using $2^{44.7}$ keystream bytes. Later, Golić's attack was improved by S. Fluhrer and D. McGrew, and this attack requires $2^{30.6}$ keystream bytes [26]. Their attack uses the correlations between Z_t and Z_{t+1} . I. Mantin and A. Shamir showed the probability that $Z_2 = 0$ is twice as large as expected [27]. S. Paul and B. Preneel also showed the bias caused by $Z_1 \neq Z_2$ [28]. When RC4 is used in broadcast applications, the distinguishing attacks using these biases work. The distinguishing attacks of [27] and [28] require 2^8 and 2^{25} keystream bytes, respectively. Recently,

I. Mantin proposed a distinguishing attack based on statistical biases of the digraphs distribution of a keystream [29]. Mantin's attack can distinguish the keystream from a truly random sequence using $2^{26.5}$ keystream bytes with a probability greater than $2/3$.

I. Mantin proposed a method for predicting a part of a keystream from another part of the keystream [29]. A prediction attack is based on the specific state called recyclable fortuitous states. The attack can predict 1 bit of a keystream with a probability of 0.85 using 2^{45} keystream bytes, and predict 8 bits of the keystream with a probability of 0.82 using 2^{50} keystream bytes. R. Jenkins discussed the probabilistic correlation between the secret information ($S_{t-1}[i_t]$, $S_{t-1}[j_t]$, j_t) and the public information (i_t , Z_t) [30]. He showed that the probability that $S_{t-1}[i_t] = (i_t - Z_t) \bmod N$ holds and the probability that $S_{t-1}[j_t] = (j_t - Z_t) \bmod N$ holds are twice as large as expected.

Analyses above are theoretically important, but they do not pose a direct threat. As attacks for threatening RC4, several analyses on the internal state reconstruction method [24], [29], [31]–[34] have been studied. If the size of a secret key is sufficiently large, then those methods can recover an initial state from its keystream more efficiently than exhaustive key search[†]. However, now, no practical internal state reconstruction method for a recommended-size secret key (that is, a 128-bit secret key) has been proposed. The internal state reconstruction method for a recommended-size secret key remains a challenging matter.

(2) Analysis of the Initialization Process

S. Fluhrer, I. Mantin, and A. Shamir proposed a distinguishing attack based on the weakness of KSA [12]. They identified a weak key class based on the invariance weakness of KSA. The property of the weak keys is as follows. $n + q(L - 1) + 1$ bits of the weak keys are fixed to satisfy a specific condition, where q is a parameter such that $q \leq n$ and $2^q | L$. The specific $O(qN)$ bits of the initial state initialized by the weak keys are determined with a probability of 0.5. The weak keys are used in the distinguishing attack, and the number of keystream bytes required by the attack depends on the length of a secret key. When the length is short (e.g., a 64-bit secret key), the attack works more efficiently than the attack of [26].

Well-known results for recovering a secret key are the chosen IV attack and the related-key attack^{††} proposed by S. Fluhrer, I. Mantin, and A. Shamir [12], [13]. Notice that these attacks work only under specific conditions, namely, the generation of the encryption key in each session has a specific rule. The most significant application of these attacks is the chosen IV attack against RC4 of the WEP implementation. The attack is called the Fluhrer, Mantin, and Shamir attack (the FMS attack). WEP uses a packet-variable key K (called a packet key). K is generated from a fixed secret information K' (called a WEP key) and a 3-byte IV as follows:

$$K = IV \parallel K', \quad (13)$$

where the IV of WEP is a packet-variable public value and \parallel is concatenation. $K[0], K[1], K[2]$ comprise the IV and $K[3], K[4], \dots, K[L - 1]$ comprise the WEP key. The FMS attack uses the IVs of specific pattern (called weak IVs). When a weak IV is used, a byte of the WEP key is recovered from the weak IV and Z_1 with a higher probability than random search. In the FMS attack, all the bytes of WEP key are recovered one by one using many pairs of weak IV and Z_1 . The time complexity of the FMS attack is proportional to the length of the WEP key. The FMS attack was verified by A. Stubblefield, J. Ioannidis, and A. D. Rubin [14], and the implementation of the FMS attack is available as open source software [15], [36]–[39]. In order to obtain a sufficient number of the pairs of weak IV and Z_1 for recovering a 104-bit WEP key, the FMS attack requires about 4,000,000 to 6,000,000 packets [14].

Later, the FMS attack was improved [15]–[19], and many weak IVs different from those of earlier FMS attacks have been identified. In order to protect against these key recovery attacks, WEPplus [14], [40] is used in current WEP products. WEPplus is a method for skipping weak IVs and the corresponding packet keys. If all the patterns of weak IVs are supported by WEPplus, such WEP implementation is secure against FMS-like key recovery attacks. Recently, the known IV attack against WEP has been proposed by A. Klein [20], and this attack has been optimized by E. Tews, R. Weinmann, and A. Pyshkin [21]. WEPplus cannot work against these attacks since all the IVs become weak IVs in these attacks. In addition, the attack proposed by Tews, Weinmann, and Pyshkin can recover a 104-bit WEP key with a probability of 0.5 when 40,000 packets are given and with a probability of 0.95 when 85,000 packets are given. Their results concluded that WEP is not secure. However, it does not mean that RC4 is insecure. Namely, the protocol vulnerability causes insecurity of WEP. We notice that no effective attack for threatening RC4 itself has been discovered.

As attacks for threatening RC4 itself, we focus on a key recovery attack that consists of the internal state reconstruction method and the key reconstruction method. The latter method, which is used for recovering a secret key from an initial state, has not been discussed yet. In this paper, we discuss this method against RC4. If the internal state reconstruction method succeeds in the future, our method will assist the key recovery attack from the recovered initial state.

3. Definition of Invertible Keys

When the internal state reconstruction method succeeds, it

[†]For example, a method proposed by L.R. Knudsen et al. [31] can recover an initial state from its keystream more efficiently than exhaustive key search when the key size is larger than 798 bits. In 2003, the method was improved by us [33], and the improved method can attack efficiently when the key size is larger than 588 bits.

^{††}The chosen IV attack is based on the weaknesses shown by D. Wagner [35]. The related-key attack is based on the weak keys that have an invariance weakness.

is desirable that no secret key is recovered from an initial state more efficiently than exhaustive key search in order to avoid the leakage of the secret key information. We discuss the secret key that is recovered from an initial state more efficiently than exhaustive key search. Such a secret key is recovered with the time complexity less than 2^ℓ times that of KSA. Specifically, we focus on the ℓ -bit secret key that is recovered very easily (with the time complexity at most ℓ times that of KSA), and we define it as follows:

Definition 1: Let us consider the time complexity that a secret key with ℓ bits can be recovered from an initial state. If the time complexity is at most ℓ times that of KSA, we call the secret key an *invertible-key*.

For simplification of the discussion, we also define a subclass of the invertible-keys as follows:

Definition 2: If an invertible-key with ℓ bits can be recovered from only ℓ bits of an initial state, we call the invertible-key a *non-expanded invertible-key*.

4. The Number of Non-expanded Invertible-Keys of RC4

In this section, we discuss the number of non-expanded invertible-keys of RC4. First, we give linear equations to recover one element of a secret key from one or two elements of an initial state. The linear equations can recover one element of the secret key with a higher probability than random search. Next, we give a method to recover all the L elements of the secret key from the first L elements of the initial state using the linear equations. Additionally, we evaluate a lower bound on the rate of non-expanded invertible-keys from the probability that the secret key is recovered by the method. From the lower bound and the size of secret keys, a lower bound on the number of non-expanded invertible-keys is determined.

4.1 Linear Equations

We present linear equations based on the vulnerability that the shuffle of an internal state in KSA is inadequate. Specifically, in KSA at time t , an internal state is updated by swapping only two elements; $S_{t-1}^*[i_{t-1}^*]$ and $S_{t-1}^*[j_t^*]$. In addition, $i_{t-1}^* = t - 1$ holds from Eq. (6). Then, for all the elements of the internal state, except for $S_t^*[t - 1]$, it probably holds that $S_t^*[x] = S_{t-1}^*[x]$. We generalize this property and calculate its probability as follows:

Lemma 1: Let $X = \{0, 1, \dots, N - 1\} \setminus \{t_s, t_s + 1, \dots, t_e - 1\}$ be a set of indices of the internal state elements that hold the condition, where t_s and t_e are any times such that $t_s < t_e$. The condition is that all the elements of X are different from i_{t-1}^* for $t = t_s + 1, t_s + 2, \dots, t_e$. Further, let B be a subset of any r elements of X . Suppose that j_t^* is a uniformly distributed variable for $t = t_s + 1, t_s + 2, \dots, t_e$. Then, $S_{t_e}^*[b] = S_{t_s}^*[b]$ for $\forall b \in B$ holds with the probability $((N - r)/N)^{t_e - t_s}$.

Proof. At time t , an internal state is updated by swapping $S_{t-1}^*[i_{t-1}^*]$ with $S_{t-1}^*[j_t^*]$. From Eq. (6), $i_{t-1}^* = t - 1$ holds. Hence, $i_{t-1}^* \neq b$ for $\forall b \in B (\subset X)$ holds for $t = t_s + 1, t_s + 2, \dots, t_e$. If $j_t^* \neq b$ for $\forall b \in B$ holds during the period, then $S_{t_e}^*[b] = S_{t_s}^*[b]$ for $\forall b \in B$ holds because $S_{t-1}^*[b]$ for $\forall b \in B$ is unchanged during the period. When j_t^* is assumed to be a uniformly distributed variable during the period, the probability that $S_{t_e}^*[b] = S_{t_s}^*[b]$ for $\forall b \in B$ holds is equal to $((N - r)/N)^{t_e - t_s}$. \square

Lemma 1 is used for obtaining the probability that r elements of the internal state are unchanged during $t = t_s + 1, t_s + 2, \dots, t_e$. It is necessary to obtain the probability that multiple elements of the internal state are unchanged during a certain period in order to calculate the probability that the linear equation holds. The same is true of the probability that the secret key is recovered by the method in Sect. 4.2. The derivations of those probabilities become simple by Lemma 1. Concretely, Lemma 1 is used in Lemma 3 and Lemma 5.

We obtain linear equations to guess one element of a secret key from one or two elements of an initial state using the property generalized as Lemma 1. The linear equations are given as follows:

$$K[x \bmod L] = \begin{cases} S_0[x] & \text{if } x = 0, \\ (S_0[x] - S_0[x - 1] - x) \bmod N & \text{if } x = 1, 2, \dots, N - 1. \end{cases} \quad (14)$$

We show conditions for Eq. (14).

Lemma 2: Let $t' \in \{1, 2, \dots, N\}$ be the time that satisfies $t' = x + 1$. When $t' = 1$, Eq. (14) holds if Eq. (21) holds. When $t' = 2, 3, \dots, N$, Eq. (14) holds if Eqs. (15)–(21) hold.

$$S_{t'-2}^*[i_{t'-1}^*] = i_{t'-1}^*, \quad (15)$$

$$S_{t'-2}^*[j_{t'-1}^*] = j_{t'-1}^*, \quad (16)$$

$$S_{t'-1}^*[j_{t'}^*] = j_{t'}^*, \quad (17)$$

$$j_{t'-1}^* \neq i_{t'-1}^*, \quad (18)$$

$$j_{t'}^* \neq i_{t'-2}^*, \quad (19)$$

$$S_N^*[t' - 2] = S_{t'}^*[t' - 2], \quad (20)$$

$$S_N^*[t' - 1] = S_{t'}^*[t' - 1]. \quad (21)$$

Proof. We show that Eq. (14) holds if Eqs. (15)–(21) hold when $t' = 2, 3, \dots, N$ (cf. Fig. 2). When Eqs. (15) and (18) hold, $S_{t'-2}^*[i_{t'-1}^*] = i_{t'-1}^*$ holds from Eq. (4). From Eq. (6), $i_{t'-1}^* = t' - 1$ holds. Then, Eq. (3) is rewritten as

$$K[(t' - 1) \bmod L] = (j_{t'}^* - j_{t'-1}^* - (t' - 1)) \bmod N. \quad (22)$$

From Eqs. (4), (6), (16), and (17), $S_{t'-1}^*[t' - 2] = j_{t'-1}^*$ and $S_{t'}^*[t' - 1] = j_{t'}^*$ hold. In addition, $S_{t'}^*[t' - 2] = S_{t'-1}^*[t' - 2] = j_{t'-1}^*$ holds from Eqs. (4), (6), and (19). Hence, the equalities $S_0[t' - 2] = S_N^*[t' - 2] = j_{t'-1}^*$ and $S_0[t' - 1] = S_N^*[t' - 1] = j_{t'}^*$ are derived from $S_0 = S_N^*$ and Eqs. (20) and (21). As a consequence, Eq. (14) holds using the equalities and Eq. (22), where $x = t' - 1$.

Suppose that $t' = 1$. It is noted that $S_{t'-1}^*[i_{t'-1}^*] =$

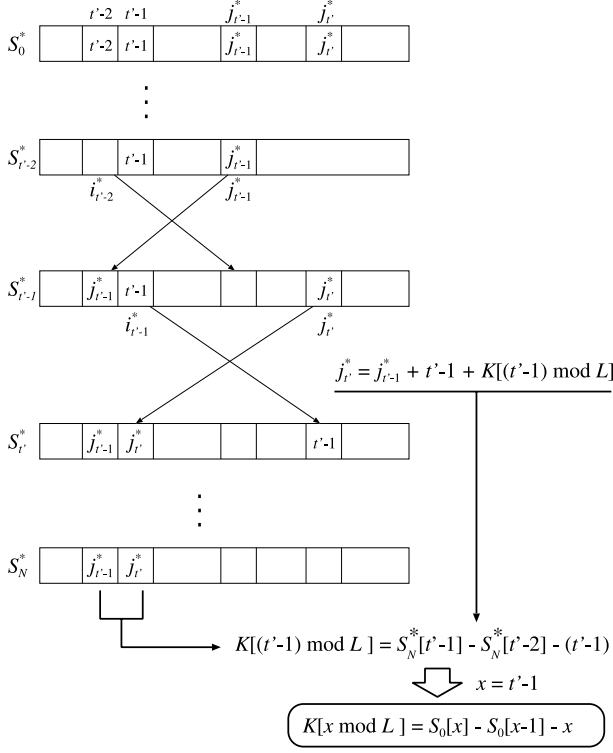


Fig. 2 Conditions for Eq. (14).

$S_0^*[i_{t'-1}^*] = i_{t'-1}^*$ implies Eq. (1) when $t' = 1$. Hence, Eqs. (15) and (18) are not required. In addition, Eqs. (16), (19), and (20) are not required, since $j_{t'-1}^* = j_0^* = 0$ holds from Eq. (2). $S_{t'-1}^*[j_{t'}^*] = S_0^*[j_{t'}^*] = j_{t'}^*$ implies Eq. (1) when $t' = 1$. Hence, Eq. (17) is not required. Therefore, Eq. (14) holds if Eq. (21) holds. \square

Lemma 3 gives the probability that the conditions of Lemma 2 hold.

Lemma 3: Let E_{S1} be the event where the conditions of Lemma 2 are satisfied. Suppose that j_t^* is a uniformly distributed variable for $t = 1, 2, \dots, N$. Then, the probability that E_{S1} occurs is

$$\Pr(E_{S1}) = P_1 \text{ if } x = 0, \quad (23)$$

$$\Pr(E_{S1}) = P_2 \cdot P_3 \cdot P_4 \cdot P_5 \text{ if } x = 1, 2, \dots, N-1, \quad (24)$$

where

$$\begin{aligned} P_1 &= \left(\frac{N-1}{N}\right)^{N-1}, \\ P_2 &= \left(\frac{N-1}{N}\right)^{x-1}, \\ P_3 &= \frac{x}{N} \cdot \frac{1}{N} + \frac{N-x-1}{N} \cdot \left(\frac{N-1}{N}\right)^{x-1}, \\ P_4 &= \frac{x-1}{N} \cdot \frac{1}{N} + \frac{N-x}{N} \cdot \left(\frac{N-1}{N}\right)^x, \\ P_5 &= \left(\frac{N-2}{N}\right)^{N-x-1}. \end{aligned}$$

Proof. When $t' = 2, 3, \dots, N$ ($x = 1, 2, \dots, N-1$), the probability $\Pr(E_{S1})$ is given by the product of the probabilities that Eqs. (15)–(21) hold.

First, we discuss the probability that Eq. (15) holds. Since $S_0^*[i_{t'-1}^*] = i_{t'-1}^*$ holds from Eq. (1), its probability is equal to the probability that $S_{t'-2}^*[i_{t'-1}^*] = S_0^*[i_{t'-1}^*]$ holds. Since $i_{t-1}^* \neq i_{t'-1}^* = t' - 1$ holds for $t = 1, 2, \dots, t' - 2$ from Eq. (6), Lemma 1 gives the probability that $S_{t'-2}^*[i_{t'-1}^*] = S_0^*[i_{t'-1}^*]$ holds as follows:

$$P_2 = \left(\frac{N-1}{N}\right)^{t'-2} = \left(\frac{N-1}{N}\right)^{x-1}. \quad (25)$$

Second, we discuss the probability that Eqs. (16) and (18) hold. Equation (18) is split into two cases; $j_{t'-1}^* \leq i_{t'-1}^* - 1$ and $j_{t'-1}^* \geq i_{t'-1}^* + 1$. We obtain the probability that Eqs. (16) and (18) hold by the sum of the probabilities that Eq. (16) holds in those cases. When $j_{t'-1}^*$ is a uniformly distributed variable, the probability that $j_{t'-1}^* \leq i_{t'-1}^* - 1$ occurs is equal to $(t' - 1)/N$. In the case of $j_{t'-1}^* \leq i_{t'-1}^* - 1$, $S_{t'-2}^*[j_{t'-1}^*]$ has already been swapped at time $t = j_{t'-1}^* + 1$. Then, the probability that $S_{t'-2}^*[j_{t'-1}^*] = j_{t'-1}^*$ holds is approximated to $1/N$. The probability that $j_{t'-1}^* \geq i_{t'-1}^* + 1$ occurs is equal to $(N - t')/N$. In the case of $j_{t'-1}^* \geq i_{t'-1}^* + 1$, $i_{t-1}^* \neq j_{t'-1}^*$ holds for $t = 1, 2, \dots, t' - 2$. Then, the probability that $S_{t'-2}^*[j_{t'-1}^*] = j_{t'-1}^* = S_0^*[j_{t'-1}^*]$ holds is equal to $((N - 1)/N)^{t'-2}$ from Lemma 1. Therefore, the total probability that Eqs. (16) and (18) hold is given as follows:

$$\begin{aligned} P_3 &= \frac{t'-1}{N} \cdot \frac{1}{N} + \frac{N-t'}{N} \cdot \left(\frac{N-1}{N}\right)^{t'-2}, \\ &= \frac{x}{N} \cdot \frac{1}{N} + \frac{N-x+1}{N} \cdot \left(\frac{N-1}{N}\right)^{x-1}. \end{aligned} \quad (26)$$

In the same way as Eq. (26), the probability that Eqs. (17) and (19) hold is obtained as follows:

$$\begin{aligned} P_4 &= \frac{t'-2}{N} \cdot \frac{1}{N} + \frac{N-t'+1}{N} \cdot \left(\frac{N-1}{N}\right)^{t'-1}, \\ &= \frac{x-1}{N} \cdot \frac{1}{N} + \frac{N-x}{N} \cdot \left(\frac{N-1}{N}\right)^x. \end{aligned} \quad (27)$$

Third, we discuss the probability that Eqs. (20) and (21) hold. Since $i_{t-1}^* \neq i_{t'-2}^*$ and $i_{t-1}^* \neq i_{t'-1}^*$ hold for $t = t' + 1, t' + 2, \dots, N$, Lemma 1 gives the probability that Eqs. (20) and (21) hold as follows:

$$P_5 = \left(\frac{N-2}{N}\right)^{N-t'} = \left(\frac{N-2}{N}\right)^{N-x+1}. \quad (28)$$

Therefore, when $x = 1, 2, \dots, N-1$, the probability $\Pr(E_{S1})$ is given as Eq. (24) by the product of Eqs. (25)–(28).

We discuss the probability that E_{S1} occurs when $t' = 1$ ($x = 0$); this probability is given by the probability that Eq. (21) holds. Since $i_{t-1}^* \neq i_{t'-1}^*$ holds for $t = t' + 1, t' + 2, \dots, N$ from Eq. (6), the probability that Eq. (21) holds is equal to P_1 from Lemma 1. Therefore, when $x = 0$, the probability $\Pr(E_{S1})$ is given by Eq. (23). \square

Theorem 1 gives the probability that each proposed linear equation holds.

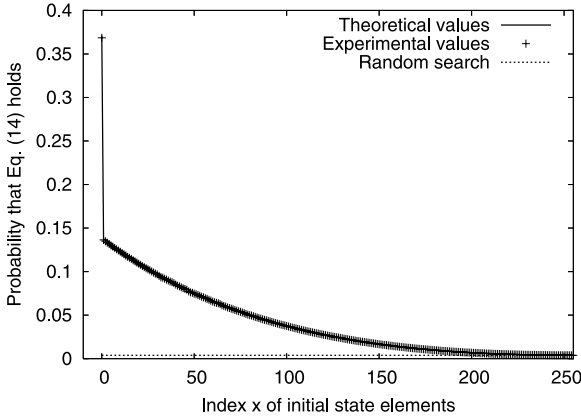


Fig. 3 Probability that Eq. (14) holds.

Theorem 1: Let E_{T1} be the event that satisfies Eq. (14). Suppose that the conditional probability of E_{T1} given $\overline{E_{S1}}$ is equal to $1/N$. Then, the probability that E_{T1} occurs is

$$\frac{1}{N} \cdot ((N-1) \cdot \Pr(E_{S1}) + 1). \quad (29)$$

Proof. From Lemma 2, the conditional probability of E_{T1} given E_{S1} is equal to one. Suppose that the conditional probability of E_{T1} given $\overline{E_{S1}}$ is equal to $1/N$, since the probability that Eq. (14) accidentally holds is equal to $1/2^n = 1/N$. Then, the probability that E_{T1} occurs is given as follows:

$$\begin{aligned} \Pr(E_{T1}) &= \Pr(E_{T1}|E_{S1}) \cdot \Pr(E_{S1}) \\ &\quad + \Pr(E_{T1}|\overline{E_{S1}}) \cdot \Pr(\overline{E_{S1}}) \\ &= 1 \cdot \Pr(E_{S1}) + \frac{1}{N} \cdot (1 - \Pr(E_{S1})) \\ &= \frac{1}{N} ((N-1) \cdot \Pr(E_{S1}) + 1). \end{aligned}$$

□

When Eq. (14) holds, one element of the secret key $K[x \bmod L]$ is recovered from one or two elements of the initial state $S_0[x]$ and $S_0[x-1]$ (or only $S_0[x]$). From Theorem 1, the probability that $K[x \bmod L]$ is recovered by Eq. (14) is $(N-1) \cdot \Pr(E_{S1}) + 1$ times that of the random search of $K[x \bmod L]$. Additionally, from Lemma 3, the probability is large when the index of the element of an internal state x is small. Figure 3 shows the probability for each x .

4.2 Key Recovering from the Initial State

In this subsection, we give a method to recover an ℓ -bit secret key from the first ℓ bits of an initial state using the linear equations with the time complexity less than that of one execution of KSA. It is certified that these secret keys are non-expanded invertible-keys. However, we cannot prove that non-expanded invertible-keys are obtained by only our method. Hence, the rate of non-expanded invertible-keys is not less than that of these secret keys.

We discuss the choice of the linear equations.

In order to recover all the elements of a secret key $K[0], K[1], \dots, K[L-1]$, we require these linear equations of which number is L . In most cases, a linear equation requires two elements of the initial state in order to recover one element of the secret key, thus, $2L$ elements of the initial state are required for recovering the secret key. However, in the following cases, the number of the required elements decreases. When we choose two linear equations that are used in $x = x'$ and $x = x' + 1$, the number of the required elements decreases by one. The reason is that one of the required elements overlaps in two linear equations. In addition, when $x = 0$, the linear equation requires one element of the initial state. Hence, in order to minimize the number of the required elements, the linear equations are used in every $x = 0, 1, \dots, L-1$. In such a case, L elements of the secret key can be recovered from the L elements of the initial state $S_0[0], S_0[1], \dots, S_0[L-1]$. Therefore, the ℓ -bit secret key can be recovered from $L \cdot n$ ($\sim \ell$) bits of the initial state with the time complexity for computing L linear equations.

We give the proposed method as follows:

Algorithm 1:

Input: $S_0[0], S_0[1], \dots, S_0[L-1]$.

Output: $K[0], K[1], \dots, K[L-1]$.

Step 1: Set $x = 0$.

Step 2: Calculate $K[x \bmod L]$ from Eq. (14).

Step 3: Update x by $x = x + 1$.

Step 4: If $x \neq L$ go to Step 2. Otherwise stop.

It requires the first L elements of the initial state to recover the secret key. Hence, it is executed after these elements are obtained from a keystream using the internal state reconstruction method. We can confirm whether the obtained secret key is correct or not by comparing a keystream generated from its secret key with the correct keystream.

Now, we discuss the time complexity of the method. Suppose that the computational cost of addition modulo N is equal to subtraction modulo N . Then the computational cost of Eq. (3) and that of a linear equation are equivalent. In KSA, Eqs. (3)–(5) are computed N times. Since $L \leq N$, the time complexity of our method is less than that of one execution of KSA.

We calculate the probability that the secret key is recovered by our method. However, we cannot obtain it by the product of the probabilities that Eq. (14) holds for $x = 0, 1, \dots, L-1$ since the conditions of Lemma 2 overlap for $x = 0, 1, \dots, L-1$. Therefore, in order to obtain the probability that the secret key is recovered by the method, we rearrange the conditions that Eq. (14) holds for $x = 0, 1, \dots, L-1$. Lemma 4 shows conditions that Eq. (14) holds for $x = 0, 1, \dots, L-1$.

Lemma 4: If Eqs. (30) and (31) hold for $t = 1, 2, \dots, L$ and Eq. (32) holds, then Eq. (14) holds for $x = 0, 1, \dots, L-1$.

$$j_t^* = c \text{ for } \exists c \in \{i_{t-1}^*, L, L+1, \dots, N-1\}, \quad (30)$$

$$S_{t-1}^*[j_t^*] = j_t^*, \quad (31)$$

$$S_N^*[t' - 1] = S_L^*[t' - 1] \text{ for } \forall t' \in \{1, 2, \dots, L\}. \quad (32)$$

Proof. We show that Eq. (14) holds for $x = 0, 1, \dots, L-1$ if Eqs. (30) and (31) hold for $t = 1, 2, \dots, L$ and Eq. (32) holds. $S_{t'-1}^*[i_{t'-1}^*] = S_0^*[i_{t'-1}^*] = i_{t'-1}^* = t' - 1$ holds for $\forall t' \in \{1, 2, \dots, L\}$ when Eq. (30) holds for $t = 1, 2, \dots, L$. Then, Eq. (22) holds at every $t' = 1, 2, \dots, L$. When Eq. (31) holds for $t = 1, 2, \dots, L$, $S_{t'}^*[i_{t'-1}^*] = S_{t'}^*[t' - 1] = j_{t'}^*$ holds for $\forall t' \in \{1, 2, \dots, L\}$. In addition, $S_L^*[t' - 1] = S_{t'}^*[t' - 1] = j_{t'}^*$ holds for $\forall t' \in \{1, 2, \dots, L\}$ when Eq. (30) holds for $t = 1, 2, \dots, L$. Hence, the equalities $S_0[t' - 1] = S_N^*[t' - 1] = j_{t'}^*$ for $\forall t' \in \{1, 2, \dots, L\}$ are derived from $S_0 = S_N^*$ and Eq. (32). Consequently, Eq. (14) holds for $x = 0, 1, \dots, L-1$ using the equalities and Eq. (22), where $x = t' - 1$. \square

Lemma 5 gives the probability that the conditions of Lemma 4 hold, namely, the probability that Eqs. (30) and (31) hold for $t = 1, 2, \dots, L$ and Eq. (32) holds.

Lemma 5: Let E_{S2} be the event where the conditions of Lemma 4 are satisfied. Suppose that j_t^* is a uniformly distributed variable for $t = 1, 2, \dots, N$. Then, the probability that E_{S2} occurs is

$$\Pr(E_{S2}) = P_6 \cdot P_7, \quad (33)$$

where

$$P_6 = \left(\frac{N-L+1}{N} \right)^L \cdot \left(\frac{N-1}{N} \right)^{\frac{L(L-1)}{2}},$$

$$P_7 = \left(\frac{N-L}{N} \right)^{N-L}.$$

Proof. The probability that E_{S2} occurs is given by the product of the probabilities that Eqs. (30) and (31) hold for $t = 1, 2, \dots, L$ and the probability that Eq. (32) holds.

First, we discuss the former. When j_{t-1}^* is a uniformly distributed variable, the probability that Eq. (30) holds for $t = 1, 2, \dots, L$ is equal to $((N-L+1)/N)^L$. In such a case, the probability that Eq. (31) holds at time t is equal to $((N-1)/N)^{t-1}$ from Lemma 1 since $i_{t'-1}^* \neq j_t^*$ holds at every $t' = 1, 2, \dots, t-1$ and $S_0^*[j_t^*] = j_t^*$ holds from Eq. (1). Then, the probability that Eq. (31) holds for $t = 1, 2, \dots, L$ is equal to $\prod_{t=1}^L ((N-1)/N)^{t-1}$. Therefore, the probability that Eqs. (30) and (31) hold for $t = 1, 2, \dots, L$ is given as follows:

$$P_6 = \prod_{t=1}^L \frac{N-L+1}{N} \cdot \prod_{t=1}^L \left(\frac{N-1}{N} \right)^{t-1},$$

$$= \left(\frac{N-L+1}{N} \right)^L \cdot \left(\frac{N-1}{N} \right)^{\frac{L(L-1)}{2}}. \quad (34)$$

Second, we discuss the latter. Since $i_{t-1}^* \neq 0, 1, 2, \dots, L-1$ holds for $t = L+1, L+2, \dots, N$, Lemma 1 gives the probability that Eq. (32) holds as follows:

$$P_7 = \left(\frac{N-L}{N} \right)^{N-L}. \quad (35)$$

Therefore, $\Pr(E_{S2})$ is given by Eq. (33) from the product of Eqs. (34) and (35). \square

Theorem 2 gives the probability that the secret key is recovered by the method.

Theorem 2: Let E_{T2} be the event that satisfies Eq. (14) for $x = 0, 1, \dots, L-1$. Suppose that the conditional probability of E_{T2} given $\overline{E_{S2}}$ is equal to $1/N^L$. Then, the probability that E_{T2} occurs is

$$\frac{1}{N^L} \cdot \left((N^L - 1) \cdot \Pr(E_{S2}) + 1 \right). \quad (36)$$

Proof. From Lemma 4, the conditional probability of E_{T2} given E_{S2} is equal to one. Suppose that the conditional probability of E_{T2} given $\overline{E_{S2}}$ is equal to $1/N^L$, since the probability that Eq. (14) accidentally holds for $x = 0, 1, \dots, L-1$ is equal to $(1/2^n)^L = 1/N^L$. Then, the probability that E_{T2} occurs is given as follows:

$$\begin{aligned} \Pr(E_{T2}) &= \Pr(E_{T2}|E_{S2}) \cdot \Pr(E_{S2}) \\ &\quad + \Pr(E_{T2}|\overline{E_{S2}}) \cdot \Pr(\overline{E_{S2}}) \\ &= 1 \cdot \Pr(E_{S2}) + \frac{1}{N^L} \cdot (1 - \Pr(E_{S2})) \\ &= \frac{1}{N^L} \cdot \left((N^L - 1) \cdot \Pr(E_{S2}) + 1 \right). \end{aligned}$$

\square

The lower bound on the rate of non-expanded invertible-keys is obtained from Eqs. (33) and (36) when the secret keys are selected randomly. When $\ell = 128$ and $n = 8$, the lower bound is equal to $1/2^{24.4}$, that is, at least $2^{103.6}$ secret keys are non-expanded invertible-keys.

4.3 Experimental Results

For simplification of the discussion, our analysis requires assumptions. In order to certify the validity of our analysis, two kinds of experiments are carried out. In the experiments, we use 2^{40} secret keys with $n = 8$ and $\ell = 128$, and recover the secret keys from the corresponding initial states.

In the first experiment, we check the validity of Theorem 1 by calculating the probability that the linear equation holds for each x . Figure 3 shows the experimental values and theoretical ones for each x . The theoretical values agree well with the experimental values.

In the second experiment, we check the validity of Theorem 2 by the rate of recoverable secret keys from the first ℓ bits of the initial state. From the experiment, we obtained 47,286 ($\approx 2^{15.5}$) non-expanded invertible-keys out of 2^{40} secret keys. Therefore, the rate of non-expanded invertible-keys is equal to $1/2^{24.5}$ ($= 2^{15.5}/2^{40}$), and it agrees well with the theoretical value ($1/2^{24.4}$). Although we examine a statistical relation of the non-expanded invertible-keys, we cannot find a sufficient condition for these keys. The reason for this is that the method does not require a specific initial state. Table A·1 presents an example of these keys.

5. Conclusion

We showed a method for recovering a secret key from an initial state of RC4. First, we gave the linear equations to recover one element of the secret key from one or two elements of the initial state with a higher probability than random search. Second, we presented a method to recover an

ℓ -bit secret key (L elements of the secret key) from the first L elements of the initial state using L linear equations with the time complexity less than that of one execution of KSA. For randomly selected 128-bit secret keys, it can recover the secret keys of which rate is $1/2^{24.4}$. Hence, at least $2^{103.6}$ secret keys of RC4 are non-expanded invertible-keys. Therefore, it is easy to compute a secret key from an initial state of RC4.

When the internal state reconstruction method is found, our result contributes to recover a secret key from the initial state. Now, no internal state reconstruction method for a recommended-size secret key has been proposed. However, it has not been proven that such a method does not exist. In addition, we expect to find an effective method for recovering the first L elements of an initial state from a keystream[†]. The future work is to find such a method.

Acknowledgements

We thank anonymous reviewers and Associate Editor for their helpful comments. We also thank Dr. Hidenori Kuwakado and Dr. Minoru Kuribayashi for useful discussions.

References

- [1] R.A. Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag, Berlin, 1986.
- [2] R.A. Rueppel, "Stream ciphers," in Contemporary Cryptology, ed. G.J. Simmons, pp.65–134, IEEE Press, New York, 1992.
- [3] D. Bleichenbacher and S. Patel, "SOBER cryptanalysis," Proc. FSE'99, LNCS 1636, pp.305–316, Springer-Verlag, 1999.
- [4] P. Hawkes and G. Rose, "Exploiting multiples of the connection polynomial in word-oriented stream cipher," Proc. ASIACRYPT2000, LNCS 1976, pp.302–316, Springer-Verlag, 2000.
- [5] P. Hawkes and G. Rose, "Guess-and-determine attacks on SNOW," Proc. SAC'02, LNCS 2595, pp.37–46, Springer-Verlag, 2003.
- [6] S. Kiyomoto, T. Tanaka, and K. Sakurai, "Experimental analysis of guess-and-determine attacks on clock-controlled stream ciphers," IEICE Trans. Fundamentals, vol.E88-A, no.10, pp.2778–2791, Oct. 2005.
- [7] H. Ahmadi, T. Eghlidos, and S. Khazaei, "Improved guess and determine attack on SOSEMANUK," ECRYPT Stream Cipher Project, Report 2005/085, 2005, available at <http://www.ecrypt.eu.org/stream/papersdir/085.pdf>
- [8] Y. Tsunoo, T. Saito, M. Shigeri, T. Suzaki, H. Ahmadi, T. Eghlidos, and S. Khazaei, "Evaluation of SOSEMANUK with regard to guess-and-determine attacks," ECRYPT Stream Cipher Project, Report 2006/009, 2006, available at <http://www.ecrypt.eu.org/stream/papersdir/2006/009.pdf>
- [9] J. Mattsson, "A guess-and-determine attack on the stream cipher Polar Bear," ECRYPT Stream Cipher Project, Report 2006/017, 2006, available at <http://www.ecrypt.eu.org/stream/papersdir/2006/017.pdf>
- [10] B. Schneier, Applied Cryptography, Wiley, New York, 1996.
- [11] IEEE Computer Society, "Wireless lan medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11, 1999.
- [12] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," Proc. SAC2001, LNCS 2259, pp.1–24, Springer-Verlag, 2001.
- [13] S. Fluhrer, I. Mantin, and A. Shamir, "Attacks on RC4 and WEP," CryptoBytes, vol.5, no.2, pp.26–34, RSA Laboratories, 2002.
- [14] A. Stubblefield, J. Ioannidis, and A.D. Rubin, "A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP)," ACM Trans. Information and System Security, vol.7, no.2, pp.319–332, May 2004.
- [15] Wepcrack, available at <http://sourceforge.net/projects/wepcrack/>
- [16] K. Kobara and H. Imai, "Yet other weak IVs for recovering wep keys," Proc. ISITA2004, pp.1130–1134, Parma, Italy, Oct. 2004.
- [17] T. Ohigashi, Y. Shiraishi, and M. Morii, "FMS attack-resistant WEP implementation is still broken — Most IVs leak a part of key information," Proc. CIS2005, Part II, Lecture Notes in Artificial Intelligence, vol.3802, pp.17–26, Springer-Verlag, 2005.
- [18] K. Kobara and H. Imai, "Key-dependent weak IVs and weak keys in WEP — How to trace conditions back to their patterns," IEICE Trans. Fundamentals, vol.E89-A, no.8, pp.2198–2206, Aug. 2006.
- [19] KoreK, "Next generation of WEP attacks?," 2004, available at <http://www.netstumbler.org/showpost.php?p=93942&postcount=35>
- [20] A. Klein, "Attacks on the RC4 stream cipher," 2006, available at <http://cage.ugent.be/~klein/RC4/RC4-en.ps>
- [21] E. Tews, R. Weinmann, and A. Pyshkin, "Breaking 104 bit WEP in less than 60 seconds," Cryptology ePrint, 2007, available at <http://eprint.iacr.org/2007/120.pdf>
- [22] A. Freier, P. Karlton, and P. Kocher, "The SSL 3.0 protocol," Netscape Communications, Nov. 1996.
- [23] Wi-Fi Alliance, "Wi-Fi protected access," available at <http://www.wi-fi.net/opensection/protectedAccess.asp>
- [24] J. Dj. Golić, "Iterative probabilistic cryptanalysis of RC4 keystream generator," Proc. ACISP2000, LNCS 1841, pp.220–233, Springer-Verlag, 2000.
- [25] J. Dj. Golić, "Linear statistical weakness of alleged RC4 keystream generator," Proc. EUROCRYPT'97, LNCS 1233, pp.226–238, Springer-Verlag, 1997.
- [26] S. Fluhrer and D. McGrew, "Statistical analysis of the alleged RC4 keystream generator," Proc. FSE2000, LNCS 1978, pp.19–30, 2001.
- [27] I. Mantin and A. Shamir, "A practical attack on broadcast RC4," Proc. FSE2001, LNCS 2355, pp.152–164, Springer-Verlag, 2001.
- [28] S. Paul and B. Preneel, "A new weakness in the RC4 keystream generator and an approach to improve the security of the cipher," Proc. FSE2004, LNCS 3017, pp.245–259, Springer-Verlag, 2004.
- [29] I. Mantin, "Predicting and distinguishing attacks on RC4 keystream generator," Proc. EUROCRYPT2005, LNCS 3494, pp.491–506, Springer-Verlag, 2005.
- [30] R. Jenkins, "Isaac and RC4," available at <http://burtleburtle.net/bob/rand/isaac.html>
- [31] L.R. Knudsen, W. Meier, B. Preneel, V. Rijmen, and S. Verdoelaege, "Analysis methods for (alleged) RC4," Proc. ASIACRYPT'98, LNCS 1514, pp.327–341, Springer-Verlag, 1998.
- [32] S. Mister and S.E. Tavares, "Cryptanalysis of RC4-like ciphers," Proc. SAC'98, LNCS 1556, pp.131–143, Springer-Verlag, 1999.
- [33] Y. Shiraishi, T. Ohigashi, and M. Morii, "Internal-state reconstruction of a stream cipher RC4," IEICE Trans. Fundamentals, vol.E86-A, no.10, pp.2636–2638, Oct. 2003.
- [34] V. Tomašević and S. Bojanić, "Reducing the state space of RC4 stream cipher," Proc. ICCS2004, LNCS 3036, pp.644–647, Springer-Verlag, 2004.
- [35] D. Wagner, "My RC4 weak keys," Post in sci.crypt, 1995.
- [36] AirSnort, available at <http://airsnort.shmoo.com/>
- [37] Aircrack, <http://www.cr0.net:8040/code/network/aircrack/>
- [38] Dwepcrack, <http://www.e.kth.se/~pvz/wifi/>
- [39] Weplab, <http://weplab.sourceforge.net/>
- [40] Orinoco, "WEPplus white paper," Oct. 2001.

[†]The methods for obtaining the information on the part of an initial state are discussed in [26], [27], [29]–[31].

Table A·1 Example of the non-expanded invertible-key K and the corresponding initial state S_0 ($= S_{256}^*$) and pointer j_t^* .

y	$K[y]$															
0–15	172	78	82	235	33	244	57	20	242	200	222	185	94	52	227	221

x	$S_0[x](= S_{256}^*[x])$															
0–15	172	251	79	61	98	91	154	181	175	128	104	44	150	215	200	180

t	j_t^*															
1–16	172	251	79	61	98	91	154	181	175	128	104	44	150	215	200	180
17–32	112	207	51	49	102	111	190	233	243	212	204	160	26	107	108	104
33–48	52	163	23	37	106	129	224	27	53	38	46	18	123	220	233	245
49–64	209	50	151	148	213	241	96	171	213	214	238	226	124	179	212	240
65–80	220	107	255	45	146	203	74	165	223	240	20	24	194	67	116	83
81–96	79	238	146	127	244	61	204	55	129	162	218	152	82	227	36	96
97–112	107	26	112	190	67	156	233	100	117	166	168	192	60	221	46	32
113–128	208	143	83	177	32	51	226	109	215	24	112	52	206	127	224	58
129–144	239	149	105	215	124	245	180	79	201	26	130	198	176	113	226	48
145–160	108	75	239	109	160	236	49	88	79	176	148	232	171	124	253	121
161–176	55	38	209	221	162	221	127	58	80	193	73	103	197	166	55	28
177–192	97	34	38	20	187	182	246	193	107	236	132	195	221	206	20	176
193–208	187	192	94	207	180	168	108	71	71	151	63	61	241	242	146	245
209–224	165	76	112	46	141	185	43	194	140	45	101	249	151	135	72	109
225–240	151	198	166	238	243	204	235	230	115	79	23	182	205	238	190	237
241–256	226	134	165	116	233	99	163	174	152	59	19	205	39	249	218	249

Appendix: An Example of Non-expanded Invertible-Keys

We give an example of a non-expanded invertible-key K and the corresponding initial state S_0 ($= S_{256}^*$) for the parameters $n = 8, \ell = 128$ ($N = 256, L = 16$). Table A·1 shows K and the first 16 elements of S_0 and j_t^* for $t = 1, 2, \dots, 256$.

We can confirm that $K[0], K[1], \dots, K[15]$ are recovered from $S_0[0], S_0[1], \dots, S_0[15]$ by the proposed method. In addition, we confirm that the conditions for E_{S_2} are satisfied in this example. From Table A·1, we can confirm that $j_t^* \geq 16$ holds for $t = 1, 2, \dots, 16$. Hence, Eq. (30) holds for $t = 1, 2, \dots, 16$. When $t = 1, 2, \dots, 16$, all the values of j_t^* differ from each other in Table A·1. Then, $S_{t-1}^*[j_t^*] = S_0^*[j_t^*] = j_t^*$ holds for $t = 1, 2, \dots, 16$. Hence, Eq. (31) holds for $t = 1, 2, \dots, 16$. From Table A·1, we can confirm that $j_t^* \geq 16$ holds for $t = 17, 18, \dots, 256$. This implies that $S_{t-1}^*[t' - 1]$ for $\forall t' \in \{1, 2, \dots, 16\}$ are unchanged for $t = 17, 18, \dots, 256$. Hence, Eq. (32) holds. In the example, all the conditions for E_{S_2} are satisfied.



Toshihiro Ohigashi received the B.E. and M.E. degrees from the University of Tokushima, Japan, in 2002 and 2004, respectively. He is currently a doctoral student at Kobe University. His current research interests include information security and cryptography. He received the SCIS 20th Anniversary Award from ISEC group of IEICE in 2003. He is a student member of the Information Processing Society of Japan.



Yoshiaki Shiraishi received the B.E. and M.E. degrees from the Ehime University, Japan, and the Ph.D. degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a Lecturer at the Department of Informatics, Kinki University, Japan. Since 2006, he has been an Associate Professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. His current research interests include information security, cryptography and computer network. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. Dr. Shiraishi is a member of the IEEE and the Information Processing Society of Japan.



Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor at the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering at Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering at the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronics Engineering, Faculty of Engineering at Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. Dr. Morii is a member of the IEEE, the Information Processing Society of Japan and the Society of Information Theory and Its Applications.