PAPER

Key-Dependent Weak IVs and Weak Keys in WEP — How to Trace Conditions Back to Their Patterns —

Kazukuni KOBARA[†a)], *Member and* **Hideki IMAI**[†], *Fellow*

SUMMARY The WEP (Wired Equivalent Privacy) is a part of IEEE 802.11 standard designed for protecting over the air communication. While almost all of the WLAN (Wireless LAN) cards and the APs (Access Points) support WEP, a serious key recovery attack (aka FMS attack) was identified by Fluhrer et al. The attack was then extended and implemented as WEP cracking tools. The key recovery attacks can basically be prevented by skipping certain IVs (Initial Values) called weak IVs, but the problem is that there exist huge amount of key-dependent weak IVs and the patterns of them have not been fully identified yet. The difficult part is that a naive approach to identify the key-dependent weak IVs requires the exhaustive search of IVs and WEP keys, and hence is infeasible. On the other hand, it might be feasible to skip the key-dependent weak IVs for the currently set WEP key but this reveals information on the WEP key from the skipped patterns. To skip them safely, the patterns of the key-dependent weak IVs must be identified in the first place. In this paper, we analyze the famous condition for IVs and WEP keys to be weak in the FMS attack, i.e. $0 \leq S[1] \leq t' < t$ and $S[1] + S[S[1]] = t$ (cf. Sect. 2.3 for more details), and then trace it back to the patterns of IVs and WEP keys theoretically. Once such patterns are obtained, their safe skip patterns can be obtained by using them.

key words: RC4, WEP, IEEE802.11, WLAN, FMS attack

1. Introduction

The WEP (Wired Equivalent Privacy) is a part of IEEE 802.11 standard [12] designed for protecting over the air communication. While almost all of the WLAN (Wireless LAN) cards and the APs (Access Points) support WEP, a serious key recovery attack (aka FMS attack) was identified by Fluhrer et al. [7]. The attack was then extended and implemented in [1]–[5], [15]. The key recovery attacks can basically be prevented[*] by skipping certain IVs (Initial Values) called weak IVs, but the problem is that there exist huge amount of key-dependent weak IVs and the patterns of them have not been fully identified yet.

A naive approach to identify all the key-dependent weak IVs for "all the WEP keys" would be to try all the combinations of the IVs and the WEP keys and to see if they satisfy certain conditions employed in the attacks. This approach, however, requires exhaustive search of IVs and WEP keys and is computationally infeasible. Another approach would be to skip all the IVs meeting the condition but only for "the currently set WEP key." This approach may be feasible, but causes another vulnerability that the skipped IVs reveal information on the WEP key (since the skipped IVs depend on the WEP key). To skip IVs safely, the patterns of the key-dependent weak IVs must be identified in the first place. Once such patterns are obtained, the safe skip patterns can be obtained using them.

In this paper, we analyze the famous condition for IVs and WEP keys to be weak in the FMS attack, i.e. $0 \leq S[1] \leq t' < t$ and $S[1] + S[S[1]] = t$ (cf. Sect. 2.3 for more details), which is employed in the current versions of the publicly available WEP cracking tools [1]–[5] commonly. Then we theoretically trace it back to the patterns of IVs and WEP keys. Once such weak patterns are identified, they can be converted into the safe skip patterns that immunize WEP against the key-recovery attacks employing the condition without revealing the information on the WEP key.

The organization of this paper is as follows: In Sect. 2, we briefly review WEP, RC4 and the key recovery attacks on WEP. In Sect. 3, we explain how to trace the condition back to the patterns of IVs and WEP keys.

2. WEP and Its Key Recovery Attack

In this section, we briefly review WEP and its key recovery attacks. (For details, cf. the specification [12] and the papers [7], [8], [15].)

2.1 Data Encapsulation Format in WEP

Full description of WEP is available from [12]. What is needed here is its data encapsulation format, which is given as follows:

$$\mathrm{IV}||\{(m||\mathrm{CRC}(m)) \oplus \mathrm{RC4}(\mathrm{IV}||K')\}$$

where || denotes concatenation of the right and the left data, and ⊕ denotes exclusive-or. A data packet m is encapsulated as follows: first m is concatenated with its 32-bit cyclic redundancy check CRC(m). Then (m||CRC(m)) is exclusive-ored with a pseudo-random sequence denoted by RC4(IV||K') where RC4() is the RC4 key-stream generator, IV is a 24-bit initial value and K' is a symmetric-key. IVs should be unique to each other so that RC4(IV||K') can be

Manuscript received October 24, 2005.
Final manuscript received May 11, 2006.

[†]The authors are with the Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, Tokyo, 101-0021 Japan.
a) E-mail: k-kobara@aist.go.jp

DOI: 10.1093/ietfec/e89–a.8.2198

[*]Another option to prevent the attack is to employ the new algorithms such as, TKIP [14] and AES-CCM [14]. It will, however, take a little more time before all the home users replace their old WLAN cards and APs with new ones.

unique even if K' is fixed. K' is a symmetric-key (called WEP key), which is either shared in advance as a password among authorized members or given by another mechanism, such as 802.1x [13] or WPA-PSK [6]. Let k denote the size of $K = (\mathrm{IV}||K')$ in byte, which is either 8-bytes (64-bits) or 16-bytes (128-bits)[†]. In this paper, we assume $k = 16$.

2.2 Description of RC4

RC4 is a word oriented stream cipher. Its word size is defined by n and through out this paper we assume $n = 8$, i.e. one word is 8-bits, which is the most popular setting in RC4 including the case of WEP.

RC4 consists of two algorithms: KSA (Key-Scheduling Algorithm) and PRGA (Pseudo-Random Generation Algorithm). Their algorithms are shown in Figs. 1 and 2 respectively (where $S[i]$ and $K[i]$ denote the i-th bytes of S and K, respectively). The KSA accepts a key K of k bytes and then shuffles its inner buffer S of 2^8 bytes according to K. The PRGA accepts the shuffled S and then generates a pseudo-random sequence while shuffling S again.

The security of RC4 has been studied by a lot of researchers [9]. While some unwanted properties have been identified, such as the distinguishability from a real random stream, no critical vulnerability has been identified yet on the typical usage of RC4. RC4 in WEP is, however, out of the typical usage (since it opens a part of the RC4 key to the public as an IV) and this exposes the rest of the RC4 key (the WEP key K') to the risk of recovery [7].

```
Input: a key $K$ of $k$ bytes
Output: a buffer $S$ of $2^8$ bytes

$S := (0, 1, \cdots, 2^8 - 1)$
$j := 0$
For($i = 0; i < 2^8; i++$){
    $j := j + S[i]$
        $+K[i \mod k] \mod 2^8$
    Swap $S[i]$ and $S[j]$
}
Return $S$
```

Fig. 1 KSA (for $n = 8$).

```
Input: a buffer $S$ of $2^8$ bytes and an output size $s$
Output: a sequence $Z$

$j := 0$
For($v = 1; v \le s; v++$){
    $i := v \mod 2^8$
    $j := j + S[i] \mod 2^8$
    Swap $S[i]$ and $S[j]$
    $Z[v-1] := S[S[i] + S[j] \mod 2^8]$
    Return $Z[v-1]$
}
```

Fig. 2 PRGA (for $n = 8$).

2.3 Key Recovery Attack on WEP

In this subsection, we briefly review the key recovery attack identified by Fluhrer et al. (cf. [7], [8], [15] for details). It uses the correlation between certain positions of the WEP key and the first output byte of the PRGA. The correlation becomes higher than the average for certain IVs (called weak IVs). Thus by collecting a lot of pairs of such weak IVs and the corresponding first output bytes, the WEP key can be recovered. We explain why certain $(\mathrm{IV}[0], \mathrm{IV}[1], \mathrm{IV}[2])$ reveals the information on $K[3]$ (where $(\mathrm{IV}[0], \mathrm{IV}[1], \mathrm{IV}[2]) = (K[0], K[1], K[2])$).

Let $S_i[]$ and j_i denote $S[]$ and j at the round i (before the swap operation is applied in the round) in KSA. And then let $S^*_i[]$ and j^*_i denote $S_i[]$ and j_i in PRGA to distinguish them from those in KSA. Anyone who knows $(\mathrm{IV}[0], \mathrm{IV}[1], \mathrm{IV}[2])$ can know $S_i[]$ and j_i for $j \le 3$ since they depend only on $(\mathrm{IV}[0], \mathrm{IV}[1], \mathrm{IV}[2])$. Recall that in the round i, $S_i[i]$ and $S_i[j_i]$ are swapped with each other and $S_{i+1}[i] = S_i[j_i]$ holds where $j_i \equiv j_{i-1} + S_i[i] + K[i \mod k] \pmod{2^8}$. Accordingly in the round $i = 3$, $S_3[3]$ and $S_3[j_3]$ are swapped with each other and $S_4[3] = S_3[j_3]$ holds where $j_3 \equiv j_2 + S_3[3] + K[3] \pmod{2^8}$. Here $K[3]$ can be obtained from j_3 since j_2 and $S_3[]$ are known. And then j_3 can be obtained from $S_4[3]$ since S_3 is known in $S_4[3] = S_3[j_3]$. Interestingly, $S_4[3]$ is given as the first output byte of PRGA if the following two conditions hold:

1. In KSA, three bytes $S_3[1]$, $S_3[S_3[1]]$ and $S_4[3]$ stay in the same position respectively, i.e. the position of $1, S_3[1]$ and 3 in S, and then become $S^*_1[1]$, $S^*_1[S^*_1[1]]$ and $S^*_1[3]$ in PRGA.
2. In PRGA, both (1) and (2) hold:

$$0 \le S^*_1[1] \le 2 \tag{1}$$

$$3 \equiv S^*_1[1] + S^*_1[S^*_1[1]] \pmod{2^8}. \tag{2}$$

One can easily verify that $S^*_1[3]$ in PRGA, i.e. $S_4[3]$ in KSA, is given as the first output byte of PRGA if the above two conditions are satisfied. The first condition holds with probability around 0.05 [7]. Thus (1) and (2) can be rewritten using $S_3[1]$ and $S_3[S_3[1]]$ as follows:

$$0 \le S_3[1] \le 2 \tag{3}$$

$$3 \equiv S_3[1] + S_3[S_3[1]] \pmod{2^8}. \tag{4}$$

Then (4) is generalized to

$$t \equiv S_3[1] + S_3[S_3[1]] \pmod{2^8} \tag{5}$$

for guessing not only $K[3]$ but also $K[t]$ s.t. $3 \le t$.

In practice, some variants exist on the implementation of the key-recovery attack. We categorize them as follows:

Basic variant: uses the IVs that lead to both (3) and (5). We call such IVs **key-independent weak IVs** (since they are independent of the WEP key).

[†] Some chips accept 19-bytes (152-bits).

FMS: uses not only the key-independent weak IVs but also **key-dependent weak IVs**. When $K[0]$ to $K[t']$ s.t. $2 \leq t'$ are known or guessed the condition for guessing $K[t]$ s.t. $t' < t \leq 15$ (or simply $t = t' + 1$) is given by:

$$0 \leq S_{t'+1}[1] \leq t' \tag{6}$$

$$t \equiv S_{t'+1}[1] + S_{t'+1}[S_{t'+1}[1]] \pmod{2^8} \tag{7}$$

where the condition for the Basic corresponds to $t' = 2$. Skipping only the key-independent weak IVs is not enough to prevent this attack since even if $K[3]$ to $K[t']$ are unknown this attack still works by exhaustively searching $K[3]$ to $K[t']$. This is known as the guessing-early-key-bytes approach [15].

Korek: uses new conditions recently identified by Korek. (To the best of our knowledge, Korek's result has not been published in a paper but appeared in the source codes of [1], [2], [5]. We omit the details due to the limitation of pages, but some conditions are not stable and further analysis is needed.)

3. How to Trace Conditions Back to Weak IVs

In this section, we analyze the condition, (6) and (7), and then trace it back to the patterns of weak IVs and weak WEP keys theoretically.

3.1 Condition and States to Be Led

For simplicity, we abbreviate "$a \equiv b \mod 2^8$" to "$a = b$" and omit subscripts of j and S if obvious. We rewrite the condition to be weak in the FMS, i.e. (6) and (7), using a new variable u as follows:

Definition 1: (Weak IVs in FMS) Suppose $K[0]$ to $K[t']$ are known (or $K[3]$ to $K[t']$ are exhaustively searched), then weak IVs (and weak WEP keys) for guessing $K[t]$ in the FMS are those satisfying:

$$S_{t'+1}[1] = u \tag{8}$$

$$S_{t'+1}[u] = t - u \tag{9}$$

where $0 \leq u \leq t' < t$.

One can easily verify that satisfying (8) and (9) is equivalent to satisfying (6) and (7). We prove the following theorem on the new variable u:

Theorem 1: (8) and (9) do not hold if $u = 1$ or $t = 2u$.

Proof. Suppose $u = 1$ then $S[1] = 1$ from (8) and $S[1] = t - 1$ from (9). Thus $S[1] = t - 1 = 1$, i.e. $t = 2$, and this contradicts with $t \geq 3$. Note that $K[2]$ (or IV[2]) has already been known and cannot be a target byte to crack. The other condition $t = 2u$ is obtained from the fact that S is a permutation among distinct values, i.e. $S[i] \neq S[j]$ if $i \neq j$. Since $u \neq 1$, $S[u] \neq S[1]$ where $S[u] = t - u$ and $S[1] = u$. Thus $t - u \neq u$ and $t \neq 2u$.

□

Table 1 All the patterns satisfying (8) and (9) for $0 \leq u \leq 9$.

		First Cond.	Second Cond.	Exception
u	t'	$S_{t'+1}[1] = u$	$S_{t'+1}[u] = t - u$	$t \neq 2u$
0	$0 \leq$	$S_{t'+1}[1] = 0$	$S_{t'+1}[0] = t$	-
2	$2 \leq$	$S_{t'+1}[1] = 2$	$S_{t'+1}[2] = t - 2$	$t \neq 4$
3	$3 \leq$	$S_{t'+1}[1] = 3$	$S_{t'+1}[3] = t - 3$	$t \neq 6$
4	$4 \leq$	$S_{t'+1}[1] = 4$	$S_{t'+1}[4] = t - 4$	$t \neq 8$
5	$5 \leq$	$S_{t'+1}[1] = 5$	$S_{t'+1}[5] = t - 5$	$t \neq 10$
6	$6 \leq$	$S_{t'+1}[1] = 6$	$S_{t'+1}[6] = t - 6$	$t \neq 12$
7	$7 \leq$	$S_{t'+1}[1] = 7$	$S_{t'+1}[7] = t - 7$	$t \neq 14$
8	$8 \leq$	$S_{t'+1}[1] = 8$	$S_{t'+1}[8] = t - 8$	-
9	$9 \leq$	$S_{t'+1}[1] = 9$	$S_{t'+1}[9] = t - 9$	-

Taking Theorem 1 into account, we summarize the patterns satisfying both (8) and (9) for $0 \leq u \leq 9$ in Table 1. (The cases for $10 \leq u \leq 15$ can be obtained similarly.) If $t' = 2$, i.e. $K[0]$ to $K[2]$ (or IV[0] to IV[2]) are known, only the rows of $u = 0$ and 2 are available in Table 1 since $0 \leq u \leq t' = 2$ and $u \neq 1$. The row of $u = 2$, however, cannot be used to guess $K[t]$ for $t = 4$ due to $t \neq 2u$ as shown in the column of "Exception" in the table. If $K[3]$ becomes known or guessed further, $t' = 3$ holds and the row of $u = 3$ becomes available while it cannot be used to guess $K[t]$ for $t = 6$.

3.2 Operations Leading to The States

In this subsection, we consider what kinds of swap operations lead to the states, $S_{t'+1}[1] = u$ and $S_{t'+1}[u] = t - u$, as shown in Table 1. For this purpose, we categorize the swap operations into three types, Fish-up, Kick-in and Exchange (as explained below), and then express the swap operations for $S[1]$ and $S[u]$ with the combination of them using their abbreviations F, K and E, respectively. E.g. if both u and $t - u$ are placed into $S_{t'+1}[1]$ and $S_{t'+1}[u]$ respectively with the fish-up type swap operations, we call it FF type operation.

We start by considering the following simpler scenario first:

Definition 2: (One-target-value-one-target-box Scenario) Let x denote a target value and $S[y]$ denote a target box. The one-target-value-one-target-box scenario is a scenario where x should be placed into $S[y]$ with some operations.

We also define:

Definition 3: (Critical Swap Operation) A swap operation in KSA is called critical if it is the last swap operation that places the target value x into the target box $S[y]$ up to the round $i = t'$.

A critical swap operation (for the one-target-value-one-target-box scenario) is divided into two cases according to which refers to the target value x between j and i in the swap operation. Recall that a swap operation swaps the values in $S[i]$ and $S[j]$ with each other. If j refers to the target value (and i refers to the target box), we call it "fish-up" type since it looks as if a fisherman moving with i fishes up the target value x into $S[y]$ at the round $i = y$. (One can see that $S[y] = x$ is achieved after this swap operation.) And then we call the other case, i.e. i refers to the target value (and j refers to the target box), "kick-in" type since it looks as if a soccer moving with i kicks the target value x into $S[y]$ at the round $i = S^{-1}[x]$ where $S^{-1}[x]$ denotes the position of x in S, e.g. if $S[3] = x$, $S^{-1}[x] = 3$. Formally:

Definition 4: (Fish-up Type) A critical swap operation is called fish-up type if "i" refers to the target box $S[y]$ and "j" refers to the target value x, i.e. $S[i] = S[y]$ and $S[j] = x$. This type includes the case of $i = j$.

Definition 5: (Kick-in Type) A critical swap operation is called kick-in type if "i" refers to the target value x and "j" refers to the target box $S[y]$, i.e. $S[i] = x$ and $S[j] = S[y]$. This excludes the case of $i = j$.

Theorem 2: (Impossibility of Kick-in Type) The kick-in type is not available if $S^{-1}[x] \leq y$ at the round $i = S^{-1}[x]$ where $S^{-1}[x]$ denotes the position of x in S.

Proof. Suppose $S^{-1}[x] < y$ at the round $i = S^{-1}[x]$, then a swap operation between $S[i] = x$ and $S[j] = S[y]$ can somehow kick the value x into the box $S[y]$, but this contradicts with the definition of the critical swap operation since the swapped x, which is now in $S[y]$, must be fished again at $i = y$ to keep the x in $S[y]$. This is categorized as a fish-up type, but not a kick-in type.

□

Corollary 1: (Condition to be Kick-in Type) The kick-in type is available iff $y < S^{-1}[x] \leq t'$ at the round $i = S^{-1}[x]$.

Due to Theorem 2, a kick-in type is not available if $S^{-1}[x] \leq y$ at the round $i = S^{-1}[x]$. While a kick-in type is available if $y < S^{-1}[x]$, x must be kicked into $S[y]$ up to the round i s.t. $i \leq t'$ due to Definition 3 (since otherwise $j_{t'+1}$ cannot be calculated from $K[0]$ to $K[t']$). Thus $S^{-1}[x] = i \leq t'$ must hold.

Next, we consider the scenario we are interested in:

Definition 6: (Two-target-values-two-target-boxes Scenario) Let x_1 and x_2 denote two distinct target values, and then $S[y_1]$ and $S[y_2]$ denote two distinct target boxes. Without loss of generality, we assume $y_1 < y_2$. The two-target-values-two-target-boxes scenario is a scenario where x_1 and x_2 should be placed into $S[y_1]$ and $S[y_2]$ respectively with some operations.

A straight forward solution of this scenario is to put the two critical swap operations for $S[y_1]$ and $S[y_2]$ together. We abbreviate them as FF, FK, KF and KK respectively letting F and K denote a fish-in type and a kick-up type, respectively. Formally:

Definition 7: (FF, FK, KF and KK) A combination of the swap operations for $S[y_1]$ and $S[y_2]$ is said to be FF type if both $S[y_1]$ and $S[y_2]$ are filled with the fish-up type swap operations. If both are filled with the kick-in type swap operations, it is said to be KK type. If $S[y_1]$ is filled with a fish-up type and $S[y_2]$ is filled with a kick-in type, it is called FK type. KF type is the opposite of FK, i.e. $S[y_1]$ is filled with a kick-in type and $S[y_2]$ is filled with a fish-up type.

In addition to the combinations of F and K, the following new type swap operation is available in the two-target-values-two-target-boxes scenario:

Definition 8: (Exchange Type) A swap operation is called exchange type if a target value x_2 has already been in $S[y_1]$, the other one x_1 has already been in $S[y_2]$, the swap operation exchanges x_1 and x_2 at the round $i = y_2$ and it is the last swap operation involving either x_1 or x_2.

Note that we do not call it exchange type even if x_1 is in $S[y_2]$, x_2 is in $S[y_1]$ and they are exchanged at the round $i = y_1$ (instead of $i = y_2$). Since $y_1 < y_2$, the exchanged x_2, which is now in $S[y_2]$, must be swapped again at the round $i = y_2$ to keep the state of $S[y_2] = x_2$. This is categorized as an FF type operation since the last operation is a fish-up type for $S[y_2]$ and the previous critical operation is a fish-up type for $S[y_1]$ to place x_1 into it.

Finally, we consider more concrete cases where u and $t - u$ are placed into $S[1]$ and $S[u]$, respectively. The cases are divided into two according to whether $1 < u$ or $u < 1$. (Note that $u < 1$ means $u = 0$.) We prove:

Theorem 3: For $u < 1$, i.e. $u = 0$, only an FF type operation is available.

Proof. In the case of $u = 0$, both $t - 0$ and 0 must be placed into $S[0]$ and $S[1]$, respectively. Both "exchange" and "kick-in" type operations, however, cannot be applied to them due to the following reasons: To apply an exchange type to them, t must be placed into $S[1]$ before the round $i = 1$. The target value t in $S[t]$, however, cannot be swapped into $S[1]$ at $i = 0$. A kick-in type cannot be applied to them either for moving t in $S[t]$ into $S[0]$ since $t' < t$ and i s.t. $i \leq t'$ cannot refer to t. Thus t must be fished into $S[0]$ at the round $i = 0$ while giving 0 to $S[t]$. Then the moved 0, which is now in $S[t]$, must also be fished into $S[1]$ since 0 in $S[t]$ cannot be kicked.

□

Next, we consider the case for $1 < u$. In order to apply the exchange type in this case, $t-u$ must be placed into $S[1]$ before the round $i = u$. Note that u has already been placed in $S[u]$. Placing $t - u$ into $S[1]$ can be done with either a fish-up type or a kick-in type. Thus we define FE and KE types as follows:

Definition 9: (FE and KE) Suppose $1 < u$. We call it FE type if $t - u$ is placed into $S[1]$ with a fish-up type swap operation, and then $t - u$ in $S[1]$ and u in $S[u]$ are swapped with an exchange type at the round $i = u$. We call it KE type if $t - u$ is placed into $S[1]$ with a kick-up type swap operation, and then $t - u$ in $S[1]$ and u in $S[u]$ are swapped with an exchange type at the round $i = u$.

We summarize all the available operations for satisfying (8) and (9) in Table 2 where “Fish x” and “Kick x” in the column for $S[y]$, $y \in \{1, u\}$ denote the swap operations between $S[y]$ and x with the fish-up type and the kick-in type respectively, and “Exchange u with $t-u$” denotes the swap operation between u and $t-u$ with the exchange type.

Table 2 All the available operations to place u into $S_{t'+1}[1]$ and $t-u$ into $S_{t'+1}[u]$.

Operation Type	Operations for	
	$S[1]$	$S[u]$
$u = 0$		
FF	Fish 0	Fish t
$1 < u$		
FF	Fish u	Fish $t-u$
KF	Kick u	Fish $t-u$
FK	Fish u	Kick $t-u$
KK	Kick u	Kick $t-u$
FE	Fish $t-u$	Exchange u with $t-u$
KE	Kick $t-u$	Exchange u with $t-u$

3.3 IVs and WEP Keys Invoking Each Operation

In the previous section, we listed up all the available operations, FF, KF, FK, KK, FE and KE in Table 2. In this section, we show the concrete moves of j_i corresponding to the operations and then obtain the combinations of IVs and WEP keys that realize the moves.

We show the moves of j_i in the left columns of Tables 3 to 10 respectively where $j_{l_1<i\le l_2}$ denotes j_i for $l_1 < i \le l_2$ and $j_i \neq x_1, x_2$ denotes $j_i \neq x_1$ and $j_i \neq x_2$. One can check that each move of j_i follows the operation. The move of j_i can be converted into the combination of $K[]$ and $S[]$ by substituting

$$j_i = \sum_{l=0}^{i}(K[l \mod k] + S_l[l]) \mod 2^8 \qquad (10)$$

(that holds in KSA) in the equations in the left columns of

Table 3 FF operation for $u = 0$.

Move of j_i	Constraints on $K[]$	Exception
$j_0 = t$	**(Fish t up for $S[0]$)** $K[0] = t$ (and this makes $S_1[1] = 1$ and $S_1[t] = 0$)	
$j_1 = t$	**(Fish 0 up for $S[1]$)** $j_0 + K[1] + S_1[1] = t$ (and this makes $S_2[2] = 2$ and $S_2[t] = 1$)	
$j_{1<i\le t'} \neq 0, 1$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 0, 1$

Table 4 FF operation for $1 < u$ and $u < t-u$.

Move of j_i	Constraints on $K[]$	Exception
$j_0 \neq u, t-u$		$K[0] \neq u, t-u$
$j_1 = u$	**(Fish u up for $S[1]$)** $K[0] + K[1] + S_1[1] = u$ (and this makes $S_2[u] = S_1[1]$)	
$j_{1<i<u} \neq 1, t-u$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 1, t-u$
$j_u = t-u$	**(Fish $t-u$ up for $S[u]$)** $j_1 + \sum_{l=2}^{u}(K[l] + S_l[l]) = t-u$	
$j_{u<i\le t'} \neq 1, u$		$j_u + \sum_{l=u+1}^{i}(K[l] + S_l[l]) \neq 1, u$

Table 5 FF operation for $1 < u$ and $t-u < u$.

Move of j_i	Constraints on $K[]$	Exception
$j_0 \neq t-u, u$		$K[0] \neq t-u, u$
$j_1 = u$	**(Fish u up for $S[1]$)** $K[0] + K[1] + S_1[1] = u$ (and this makes $S_2[u] = S_1[1]$)	
$j_{1<i<t-u} \neq 1, t-u$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 1, t-u$
$j_{t-u} \neq t-u+1, \cdots, u-1$		$j_1 + \sum_{l=2}^{t-u}(K[l] + S_l[l])$ $\neq t-u+1, \cdots, u-1$
$j_{t-u<i<u} \neq 1, j_{t-u}$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 1,$ $j_{t-u} + \sum_{l=t-u+1}^{i}(K[l] + S_l[l]) \neq j_{t-u}$
$j_u = j_{t-u}$	**(Fish $t-u$ up for $S[u]$)** $j_{t-u} + \sum_{l=t-u+1}^{u}(K[l] + S_l[l]) = j_{t-u}$	
$j_{u<i\le t'} \neq 1, u$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 1, u$

Table 6 FK operation for $1 < u$ and $u < t - u \leq t'$.

Move	of	j_i	Constraints on $K[]$	Exception
j_0	$\neq$	$u, t-u$		$K[0] \neq u, t-u$
j_1	$=$	u	**(Fish u up for $S[1]$)** $K[0] + K[1] + S_1[1] = u$ (and this makes $S_2[u] = S_1[1]$)	
$j_{1<i<t-u}$	$\neq$	$1, t-u$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 1, t-u$
j_{t-u}	$=$	u	**(Kick $t-u$ into $S[u]$)** $j_1 + \sum_{l=2}^{t-u}(K[l] + S_l[l]) = u$	
$j_{t-u<i\leq t'}$	$\neq$	$1, u$		$j_{t-u} + \sum_{l=t-u+1}^{i}(K[l] + S_l[l]) \neq 1, u$

Table 7 KF operation for $1 < u$ and $u < t - u \leq t'$.

Move	of	j_i	Constraints on $K[]$	Exception
$j_{0\leq i<u}$	$\neq$	$u, t-u$		$\sum_{l=0}^{i}(K[l] + S_l[l]) \neq u, t-u$
j_u	$=$	$t-u$	**(Fish $t-u$ for $S[u]$)** $\sum_{l=0}^{u}(K[l] + S_l[l]) = t-u$ (and this makes $S_{u+1}[t-u] = S_u[u]$)	
$j_{u<i<t-u}$	$\neq$	$u, t-u$		$j_u + \sum_{l=u+1}^{i}(K[l] + S_l[l]) \neq u, t-u$
j_{t-u}	$=$	1	**(Kick u into $S[1]$)** $j_u + \sum_{l=u+1}^{t-u}(K[l] + S_l[l]) = 1$	
$j_{t-u<i\leq t'}$	$\neq$	$1, u$		$j_{t-u} + \sum_{l=t-u+1}^{i}(K[l] + S_l[l]) \neq 1, u$

Table 8 KK operation for $1 < u$ and $u < t - u \leq t'$.

Move	of	j_i	Constraints on $K[]$	Exception
$j_{0\leq i<u}$	$\neq$	$u, t-u$		$\sum_{l=0}^{i}(K[l] + S_l[l]) \neq u, t-u$
j_u	$=$	1	**(Kick u into $S[1]$)** $\sum_{l=0}^{u}(K[l] + S_l[l]) = 1$	
$j_{u<i<t-u}$	$\neq$	$1, t-u$		$j_u + \sum_{l=u+1}^{i}(K[l] + S_l[l]) \neq 1, t-u$
j_{t-u}	$=$	u	**(Kick $t-u$ into $S[u]$)** $j_{t-u} + \sum_{l=u+1}^{t-u}(K[l] + S_l[l]) = u$	
$j_{t-u<i\leq t'}$	$\neq$	$1, u$		$j_{t-u} + \sum_{l=t-u+1}^{i}(K[l] + S_l[l]) \neq 1, u$

Table 9 FE operation for $1 < u$.

Move	of	j_i	Constraints on $K[]$	Exception
j_0	$\neq$	$u, t-u$		$K[0] \neq u, t-u$
j_1	$=$	$t-u$	**(Fish $t-u$ up for $S[1]$)** $K[0] + K[1] + S_1[1] = t-u$ (and this makes $S_2[t-u] = S_1[1]$)	
$j_{1<i<u}$	$\neq$	$1, u$		$j_1 + \sum_{l=2}^{i}(K[l] + S_l[l]) \neq 1, u$
j_u	$=$	1	**(Exchange $t-u$ in $S[1]$ and u in $S[u]$)** $j_1 + \sum_{l=2}^{u}(K[l] + S_l[l]) = 1$	
$j_{u<i\leq t'}$	$\neq$	$1, u$		$j_u + \sum_{l=u+1}^{i}(K[l] + S_l[l]) \neq 1, u$

Tables 3 to 10. We summarize the results in the middle and right columns respectively in each table. E.g. in Table 3, the moves of j_i are $j_0 = t$, $j_1 = t$, $j_2 \neq 0, 1$ and so on. By substituting (10), i.e. $j_0 = K[0]$, $j_1 = j_0 + (K[1] + S_1[1])$ and $j_2 = j_1 + (K[2] + S_2[2])$, to them respectively, $K[0] = t$, $j_0 + (K[1] + S_1[1]) = t$ and $j_1 + (K[2] + S_2[2]) \neq 0, 1$ are obtained where the second and the third equations give $K[1] = -S_1[1]$, $K[2] \neq 0 - S_2[2] - t$ and $K[2] \neq 1 - S_2[2] - t$ respectively since $j_0 = j_1 = t$. IVs and WEP keys satisfying the constraints on $K[]$ become weak as long as they do not meet the exceptions. The exceptions, however, have little interest since they simply cannot be weak or require additional constraints to be weak. From now on, we focus on the constraints.

Next, we consider the value of $S_l[l]$. It is limited to $0 \leq S_l[l] \leq l$ since before the round $i = l$, $S_l[l]$ may be swapped with l_1, s.t. $l_1 < l$ but not with l_2 s.t. $l \leq l_2$. Moreover, in most cases $S_l[l] = l$ holds unless the constraints on

Table 10 KE operation for $1 < u$ and $1 < t - u < u$.

Move	of	j_i	Constraints on $K[]$	Exception
$j_{0\le i<t-u}$	$\neq$	$t-u, u$		$\sum_{l=0}^{i}(K[0] + S_l[l]) \neq t-u, u$
j_{t-u}	$=$	1	**(Kick $t-u$ into $S[1]$)** $\sum_{l=0}^{t-u}(K[l] + S_l[l]) = 1$	
$j_{t-u<i<u}$	$\neq$	$1, t-u$		$j_{t-u} + \sum_{l=t-u+1}^{i}(K[l] + S_l[l])$ $\neq 1, t-u$
j_u	$=$	1	**(Exchange $t-u$ in $S[1]$ and u in $S[u]$)** $j_{t-u} + \sum_{l=t-u+1}^{u}(K[l] + S_l[l]) = 1$	
$j_{u<i\le t'}$	$\neq$	$1, u$		$j_u + \sum_{l=t-u+1}^{i}(K[l] + S_l[l]) \neq 1, u$

Table 11 Patterns of IVs and WEP keys to be weak in FMS.

Type	Relationship among u, $t-u$ and t'	First constraint on $K[]$	Second constraint on $K[]$
$u = 0$			
FF		$K[0] = t$	$K[1] = 255$
$1 < u$			
FF	$t-u < u$	$K[0] + K[1]$ $= u - 1$	$K[t-u+1] + \cdots + K[u]$ $= -1 + t(t-2u+1)/2$
	$u < t-u$	$K[0] + K[1]$ $= u - 1$	$K[2] + \cdots + K[u]$ $= t - (u+3)u/2$
FK	$u < t-u \le t'$	$K[0] + K[1]$ $= u - 1$	$K[2] + \cdots + K[t-u]$ $= u - (t-u)(t-u+1)/2$
KF	$u < t-u \le t'$	$K[0] + \cdots + K[u]$ $= (2t - 3u - u^2)/2$	$K[u+1] + \cdots + K[t-u]$ $= (2 - t - t^2 + 2tu)/2$
KK	$u < t-u \le t'$	$K[0] + \cdots + K[u]$ $= 1 - u(u+1)/2$	$K[u+1] + \cdots + K[t-u]$ $= (-2 - t - t^2 + 4u + 2tu)/2$
FE	$t-u < u$	$K[0] + K[1]$ $= t - u - 1$	$K[2] + \cdots + K[u]$ $= 1 - u(u+1)/2$
	$u < t-u$	$K[0] + K[1]$ $= t - u - 1$	$K[2] + \cdots + K[u]$ $= 2 - t - u(u-1)/2$
KE	$1 < t-u < u$	$K[0] + \cdots + K[t-u]$ $= 1 - (t-u)(t-u+1)/2$	$K[t-u+1] + \cdots + K[u]$ $= (1+t)(t-2u)/2$

$K[]$ swap $S[l]$ with something at any round $i < l$. We note the cases where constraints on $K[]$ obviously swap $S[l]$ with something in the columns of "Constraints on $K[]$." For example, in Table 4, $j_1 = u$ swaps $S_1[1]$ with $S_1[u]$. Thus $S_2[u] = S_1[1]$ instead of $S_2[u] = u$.

Using the information on $S_l[l]$, the constraints on $K[]$ can be written in more simple forms. For example, $j_1 = u$ in Table 4 gives $K[0] + K[1] = u - S_1[1]$ and this is equivalent to $K[0] + K[1] = u - 1$ in most cases (precisely, as long as $S_1[1] = 1$, i.e. $K[0] \neq 1$). We call such cases (where $S_1[1] = 1$ holds in the above example) major cases and the other cases (where $S_1[1] \neq 1$) minor cases. The minor cases have little interest since they require additional constraints, e.g. $K[0] = 1$ in the above case. Note that one additional constraint reduces the chance of IVs to be weak by a factor of around 1/256. Similarly, $j_u = t - u$ in Table 4 gives $j_1 + \sum_{l=2}^{u}(K[l] + S_l[l]) = t - u$ and by substituting $j_1 = t$, $S_u[u] = S_1[1]$ and $S_l[l] = l$ for $1 \le l < u$, the constraint on $K[]$ is given by

$$\begin{aligned}\sum_{l=2}^{u} K[l] &= t - u - j_1 - \left\{\left(\sum_{l=2}^{u-1} S_l[l]\right) + S_u[u]\right\}\\ &= t - 2u - \left\{\left(\sum_{l=2}^{u-1} S_l[l]\right) + S_1[1]\right\}\\ &= t - 2u - \{(u-1)u/2\}\\ &= t - (u+3)u/2. \qquad (11)\end{aligned}$$

Repeating the similar procedures in each operation type, we obtained the constraints on IVs and WEP keys to be weak. We summarize them in Table 11. (We also show some examples derived from this table in App. Appendix.) The famous weak pattern $(K[0], K[1]) = (t, 255)$ identified in [7] corresponds to FF of $u = 0$ in our categorization, and then the pattern identified in [10] corresponds to FF of $u = t - 1$ in our categorization†. Other than them, there exist KK of $0 < u < t - 1$, KF, FK, KK, FE and KE. The IVs to skip can be obtained by analyzing the Table 11 and this is our further study.

4. Conclusion

We analyzed the condition for IVs and WEP keys to be weak

†The pattern identified in [11] is the incomplete version of FF and the ability of it is too low to recover the 104-bit WEP key in practice.

in the FMS attack and then theoretically traced it back to the patterns of IVs and WEP keys. The obtained patterns include the key-dependent weak IVs whose patterns have never been identified. Once weak patterns are identified, they can be used to obtain the safe skip patterns that immunize WEP against the key-recovery attacks. Obtaining such safe skip patterns and verifying how effectively they work (or not) are our further study.

References

[1] aircrack, http://www.cr0.net:8040/code/network/aircrack/

[2] AirSnort, http://airsnort.shmoo.com/

[3] dwepcrack, http://www.e.kth.se/~pvz/wifi/

[4] WEPCrack, http://wepcrack.sourceforge.net/

[5] WepLab, http://weplab.sourceforge.net/

[6] Wi-Fi Alliance, "Wi-Fi protected access," http://www.wi-fi.org/opensection/protected_access.asp, 2003.

[7] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," Proc. SAC'01, LNCS 2259, pp.1–24, 2001.

[8] S. Fluhrer, I. Mantin, and A. Shamir, "Attacks on RC4 and WEP," CryptoBytes (RSA Laboratories), vol.5, no.2, pp.26–34, Summer/Fall 2002.

[9] I. Mantin, "RC4 page," http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html

[10] T. Ohigashi, Y. Shiraishi, and M. Morii, "A chosen IV attack against FMS attack-resistant WEP implementation," IEICE Technical Report, ISEC04-113, March 2005.

[11] T. Ohigashi, Y. Shiraishi, and M. Morii, "Most IVs of FMS attack-resistant WEP implementation leak secret key information," 2005 Symposium on Cryptography and Information Security, vol.4, pp.1957–1962, Jan. 2005.

[12] IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11, 1999 Edition, http://standards.ieee.org/reading/ieee/std/lanman/, 1999.

[13] IEEE Computer Society, "IEEE standards for local and metropolitan area networks: Port-based network access control," IEEE 802.1X-2001, http://standards.ieee.org/reading/ieee/std/lanman/, 2001.

[14] IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11i/D3, http://standards.ieee.org/reading/ieee/std/lanman/, 2003.

[15] A. Stubblefield, J. Ioannidis, and A. Rubin, "A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP)," ACM Trans. Inform. Syst. Security, vol.7, no.2, pp.319–332, May 2004.

Appendix: Examples of Key-Dependent Weak IVs and Weak WEP Keys

We show the patterns of weak IVs and weak WEP keys after $K[0]$ to $K[t']$ are known (or after $K[3]$ to $K[t']$ are exhaustively searched) for $t' = 2, 3$ and 4 in Tables A· 1, A· 2 and A· 3, respectively. They are obtained from Table 11 by replacing u and t with the values s.t. $u \leq t' < t \leq 15$ and in accordance with the column of "Relationship among u, $t-u$ and t'" in the table. In Tables A· 1, A· 2 and A· 3, the column of t shows the t of $K[t]$ that can be guessed (after collecting the IVs meeting the corresponding constraints on $K[]$). The column of "#" shows the number of IVs meeting the constraints on $K[]$ (among all the $(256)^3$ IVs), which gives the approximate number of the weak IVs in the type.

The number in parentheses shows the case of weak WEP keys, i.e. the case where one constraint on $K[]$ is satisfied with the WEP key, $(K[3], \cdots, K[15])$. For example, in the row of "KE" in Table A· 2, the WEP keys s.t. $K[3] = -3$ satisfy the second constraint on $K[]$ and IVs satisfying only IV[0]+IV[1]+IV[2] $= -2$ reveal the information on $K[t]$ for $t = 5$. Since 2^{16} IVs satisfy IV[0]+IV[1]+IV[2] $= -2$, $K[5]$ can be guessed efficiently than usual. Under the assumption that WEP keys are chosen uniformly, $\lceil (t-4)/2 \rceil$ out of 64 WEP keys have this vulnerability on each $K[t]$ s.t. $t \geq 5$. For example, $K[5]$ and $K[6]$ can be guessed efficiently every 64 WEP connections and then $K[7]$ and $K[8]$ can be guessed efficiently every 32 WEP connections, and so on.

Table A· 1 Patterns of IVs and WEP keys to be weak in FMS after $K[0]$ to $K[2]$ are known.

Type	u	t	First constraint on $K[]$	Second constraint on $K[]$	#
FF	0	$3, \cdots, 15$	$K[0] = t$	$K[1] = 255$	256
	2	3	$K[0] + K[1] = 1$	$K[2] = 255$	256
	2	$5, \cdots, 15$	$K[0] + K[1] = 1$	$K[2] = t - 5$	256
FE	2	3	$K[0] + K[1] = t - 3$	$K[2] = 254$	256
	2	$5, \cdots, 15$	$K[0] + K[1] = t - 3$	$K[2] = 1 - t$	256

Table A· 2 Patterns of IVs and WEP keys to be weak in FMS after $K[0]$ to $K[3]$ are known (or after $K[3]$ is exhaustively searched).

Type	u	t	First constraint on $K[]$	Second constraint on $K[]$	#
FF	3	4	$K[0] + K[1] = 2$	$K[2] + K[3] = -3$	256
	3	5	$K[0] + K[1] = 2$	$K[3] = -1$	(65, 536)
	3	$7, \cdots, 15$	$K[0] + K[1] = 2$	$K[2] + K[3] = t - 9$	256
FK	2	5	$K[0] + K[1] = 1$	$K[2] + K[3] = -4$	256
KF	2	5	$K[0] + K[1] + K[2] = 0$	$K[3] = -4$	(65, 536)
KK	2	5	$K[0] + K[1] + K[2] = -2$	$K[3] = -2$	(65, 536)
FE	3	4, 5	$K[0] + K[1] = t - 4 = 0, 1$	$K[2] + K[3] = -5$	256
	3	$7, \cdots, 15$	$K[0] + K[1] = t - 4$	$K[2] + K[3] = -1 - t$	256
KE	3	5	$K[0] + K[1] + K[2] = -2$	$K[3] = -3$	(65, 536)

Table A·3 Patterns of IVs and WEP keys to be weak in FMS after $K[0]$ to $K[4]$ are known (or after $K[3]$ and $K[4]$ are exhaustively searched).

Type	u	t	First constraint on $K[]$	Second constraint on $K[]$	#
FF	4	5	$K[0] + K[1] = 3$	$K[2] + K[3] + K[4] = -6$	256
	4	6	$K[0] + K[1] = 3$	$K[3] + K[4] = -4$	$(65,536)$
	4	7	$K[0] + K[1] = 3$	$K[4] = -1$	$(65,536)$
	4	$9,\cdots,15$	$K[0] + K[1] = 3$	$K[2] + K[3] + K[4] = t - 14$	256
FK	2	6	$K[0] + K[1] = 1$	$K[2] + K[3] + K[4] = -8$	256
	3	7	$K[0] + K[1] = 2$	$K[2] + K[3] + K[4] = -7$	256
KF	2	6	$K[0] + K[1] + K[2] = 1$	$K[3] + K[4] = -8$	$(65,536)$
	3	7	$K[0] + K[1] + K[2] + K[3] = -2$	$K[4] = -6$	$(65,536)$
KK	2	6	$K[0] + K[1] + K[2] = -2$	$K[3] + K[4] = -6$	$(65,536)$
	3	7	$K[0] + K[1] + K[2] + K[3] = -5$	$K[4] = -2$	$(65,536)$
FE	4	$5,\cdots,7$	$K[0] + K[1] = t - 5$	$K[2] + K[3] + K[4] = -9$	256
	4	$9,\cdots,15$	$K[0] + K[1] = t - 5$	$K[2] + K[3] + K[4] = -t - 4$	256
KE	4	6	$K[0] + K[1] + K[2] = -2$	$K[3] + K[4] = -7$	$(65,536)$
	4	7	$K[0] + K[1] + K[2] + K[3] = -5$	$K[4] = -4$	$(65,536)$

Kazukuni Kobara received his B.E. degree in electrical engineering and M.E. degree in computer science and system engineering from the Yamaguchi University in 1992, 1994, respectively. He also received his Ph.D. degree in engineering from the University of Tokyo in 2003. From 1994 to 2000 and 2000 to 2006 he was a technical associate and a research associate respectively for the Institute of Industrial Science of the University of Tokyo. In 2006, he joined the National Institute of Advanced Industrial Science and Technology where he is the leader of the Research Team for Security Fundamentals at the Research Center for Information Security. He received the SCIS Paper Award and the Vigentennial Award from ISEC group of IEICE in 1996 and 2003, respectively. He also received the Best Paper Award of WISA, the ISITA Paper Award for Young Researchers, the IEICE Best Paper Award (Inose Award) and the WPMC Best Paper Award in 2001, 2002, 2003 and 2005, respectively. He is a member of IACR. He served as a member of CRYPTREC (2000-present) and the vice chairperson of WLAN security committee (2003).

Hideki Imai received the B.E., M.E., and Ph.D. degrees in electrical engineering from the University of Tokyo in 1966, 1968, and 1971, respectively. He also received Honor Doctor Degrees from Soonchunhyang University in 1999 and from University of Toulon Var in 2002. From 1971 to 1992 he was on the faculty of Yokohama National University. From 1992 to 2006 he was a Professor at the Institute of Industrial Science, the University of Tokyo. Currently, he is a Professor at the Chuo University and also the Director of Research Center for Information Security, National Institute of Advanced Industrial Science and Technology. He serves as the Chair of CRYPTREC (Cryptography Techniques Research and Evaluation Committee of Japan), the Junior Past President of IEEE Information Theory Society, and a Member of the Science Council of Japan. He is a Fellow of the IEEE. From IEICE he received Best Book Awards in 1976 and 1991, Best Paper Awards in 1992, 2003 and 2004, Yonezawa Memorial Paper Award in 1992, Achievement Award in 1995, Inose Award in 2003, and Distinguished Achievement and Contributions Award in 2004. He also received IEEE Information Theory Society Golden Jubilee Paper Award in 1998, Official Commendations from the Minister of Internal Affairs and Communications and from the Minister of Economy, Trade and Industry in 2002, and Ericsson Telecommunications Award in 2005.