Passive–Only Key Recovery Attacks on RC4

Serge Vaudenay and Martin Vuagnoux

EPFL
CH–1015 Lausanne, Switzerland
http://lasecwww.epfl.ch

Abstract. We present several weaknesses in the key scheduling algorithm of RC4 when the secret key contains an initialization vector – a cryptographic scheme typically used by the WEP and WPA protocols to protect IEEE 802.11 wireless communications. First, we show how the previously discovered key recovery attacks can be improved by reducing the dependency between the secret key bytes. Then, we describe two new weaknesses related to the modulo operation of the key scheduling algorithm. Finally, we describe a passive-only attack able to significantly improve the key recovery process on WEP with a data complexity of 2^{15} eavesdropped packets.

Keywords: RC4, stream cipher, cryptanalysis, key related attack, WEP.

1 Introduction

RC4 is a stream cipher designed by Ronald Rivest in 1987. It had been initially a trade secret until the algorithm was anonymously posted to the Cypherpunks mailing list in September 1994. Nowadays, RC4 is still widely used: it is the default cipher of the SSL/TLS protocol and a cryptographic primitive of the WPA protocol. Its popularity probably comes from its simplicity and the cheap computational cost of the encryption and decryption. Due to its straightforwardness, RC4 has initiated extensive research, revealing weaknesses in case of misuse. The most famous example is the attack on the WEP (*Wired Equivalent Privacy*) protocol.

WEP is a part of the IEEE 802.11 wireless standard ratified in 1999 [1]. It was designed to provide confidentiality on wireless communications by using RC4. In order to simplify the key set up, WEP uses preinstalled fixed keys. However, RC4 is a stream cipher: the same secret key must never be used twice. To prevent any repetition, WEP concatenates to the key an initialization vector (IV), where the IV is a 24-bit value which is publicly disclosed in the header of the protocol.

The first analysis of the WEP standard has been done in 2001 by Borisov, Goldberg and Wagner in [2]. They demonstrated major security flaws revealing that WEP does not provide confidentiality, integrity and authentication. The same year, Fluhrer, Mantin and Shamir in [3] showed a noteworthy ciphertext-only attack on WEP based on the concatenated IV scheme on RC4. They proved that the secret part of the key can be recovered if a large amount of encrypted

C. Adams, A. Miri, and M. Wiener (Eds.): SAC 2007, LNCS 4876, pp. 344–359, 2007.

packets with some specific IV values are passively eavesdropped. In fact, these so called *weak keys* or *weak IV classes* were previously discovered by Andrew Roos [4] and David Wagner [5] four years before the publication of the IEEE 802.11b standard.

A practical issue of the key recovery process is to passively obtain a large amount of encrypted packets (about 4 millions of encrypted packets to recover the secret key with a success probability of 50%). To reduce this constraint, David Hulton [6] in 2002, Andrea Bittau [7] in 2003 and a hacker nicknamed Korek [8,9] in 2004 highlighted more weak IV classes. Thus, the amount of encrypted packets needed to recover the secret key with the same probability of success has been divided by four.

On the active side, WEP is not protected against active replay attacks: it is possible to replay some specific eavesdropped packets to generate wireless network traffic. Thus, the amount of encrypted packets with different IVs may be obtained faster. In 2004, tools merging all these attacks were publicly disclosed [10,11].

In 2005, Mantin presented additional attacks on truncated RC4 in [12], based on the Jenkins correlations [13]. In 2006, Klein applied the same correlations to WEP [14] to provide a remarkable known-plaintext attack which does not need weak IVs to recover the secret key. The same year, Bittau, Handley and Lackey presented in [15] new active attacks able to inject and decrypt data without recovering the secret key (these attacks are based on the fragmentation feature provided by the IEEE 802.11 standard). Finally, in 2007, a correlation related to the first three bytes of the secret key and the first byte of the keystream has been presented in [16].

In order to correct the weaknesses discovered before 2004, the Wi-Fi Alliance proposed in [17] a WEP improved protocol called WPA (*Wi-Fi Protected Access*). It has been established that WPA must be hardware compatible with existing WEP capable devices to be deployed as a software patch. Basically, WPA is a WEP wrapper which contains anti-replay protections and a key management scheme to avoid key reuse. However, the correlations discovered in this paper are still almost theoretically applicable to WPA despite that the RC4 secret key is completely different for each encrypted packet. In 2004, the Wi-Fi Alliance finally proposed a new standard called IEEE 802.11i or WPA2 [18], where RC4 can be replaced by AES.

Limitation of the Existing Attacks. Almost all key recovery attacks are related to the value of the IV: each recovered secret key byte is provided by a specific weak IV class. However, an attacker does not control the value of the IV. It means that the attacker cannot recover the secret byte $K[i]$ if he was not able to eavesdrop encrypted packets from its weak IV class.

In parallel, Klein's key recovery attack is related to the knowledge of the plaintext, which cannot be completely determined with passive-only attacks. Indeed, the secret key byte $K[i]$ cannot be recovered if the i^{th} byte of the plaintext is unknown.

Moreover, all existing key recovery attacks suffer from a relation between the secret key bytes. To recover the byte $K[i]$ of the secret key, we need to successfully rederive the previous bytes $K[0], K[1], \ldots, K[i-1]$. In practice, this constraint is a significant limitation because if the key recovery process does not work for only one key byte (because not enough encrypted packets were captured by the attacker from the concerning weak IV class or because a byte of the plaintext is unknown), all the following key bytes will be probably misrecovered. Furthermore, WPA and some implementations of WEP filter the weak IV classes discovered by Fluhrer, Mantin and Shamir in [3].

Our Contribution. In this paper, we propose an improvement, applicable to all the key recovery attacks to significantly reduce the key dependency. Therefore, it becomes possible to independently recover some parts of the secret key. It means that even if an attacker has passively eavesdropped a very limited number of encrypted packets, he is now able to recover a part of the secret key[1]. The missing key bytes may be recovered by an exhaustive search. Because we can do the assumption that the secret key byte $K[i]$ can be recovered even if the preceding key bytes are unknown, new weak IV classes have been discovered. These new attacks improve the global key recovery process.

By significantly reducing the secret key byte dependency, we have highlighted additional weaknesses. In RC4, the key is used modulo its size. It means that the secret key byte $K[i]$ is equal to $K[i + k\ell]$ (where ℓ is the size of the key, $k = 1, 2, \ldots$ and $i = 0, 1, 2, \ldots, \ell - 1$). This property was irrelevant for the existing key recovery attacks because the whole secret key had to be recovered to attack the repetition. Without the secret key byte dependency, we are able to provide new weak IV classes attacking $K[i + k\ell]$, where $k = 1, 2, \ldots, m$. A practical analysis of this improvement is given in order to prove the efficiency of these new key recovery attacks on WEP.

Structure of the Paper. Section 2 describes the foundation of the key recovery attacks on WEP, in particular the attack discovered by Fluhrer, Mantin and Shamir in [3] and the Klein attack, described in [14]. In Section 3, we study how to reduce the key bytes dependency. In section 4, we explain how to exploit the modulo operation in the KSA and how the repetition of the secret key provides new weak IV classes. Section 5 describes our practical attack. Finally we conclude with further improvements.

2 Foundation of the Key Recovery Attacks

2.1 Description of RC4

The stream cipher RC4 is divided into two parts: the Key Scheduling Algorithm (KSA) and the Pseudo Random Generator Algorithm (PRGA). The KSA

[1] This attack has been independently rediscovered later in April 2007 by Tews, Weinmann, and Pyshkin in [19]. It is based on [14] but applies to active attacks. In this paper, we decided to focus on passive ones since it is the gateway for the WPA analysis.

generates an initial state from a random key K of ℓ words of n bits as described in Algorithm 1. It starts with an array $\{0, 1, \ldots, N-1\}$, where $N = 2^n$ and swaps N pairs. At the end, we obtain the initial state S_{N-1}.

Algorithm 1. RC4 Key Scheduling Algorithm (KSA)

```
1: for i = 0 to N − 1 do
2:     S[i] ← i
3: end for
4: j ← 0
5: for i = 0 to N − 1 do
6:     j ← j + S[i] + K[i mod ℓ]
7:     swap(S[i],S[j])
8: end for
```

Once the initial state S_{N-1} created, it will be used by the second part of RC4, the PRGA. Its role is to generate a keystream of bytes which will be XORed with the plaintext to obtain the ciphertext. Thus, RC4 computes the loop of the PRGA each time a new keystream byte z_i is needed, according to Algorithm 2.

Algorithm 2. RC4 Pseudo Random Generator Algorithm (PRGA)

```
1: i ← 0
2: j ← 0
3: loop
4:     i ← i + 1
5:     j ← j + S[i]
6:     swap(S[i],S[j])
7:     keystream byte z_i = S[S[i] + S[j]]
8: end loop
```

Let $S_i[k]$ denotes the value of the array S at the index k, after the round i in the KSA. Let $S_i^{-1}[p]$ be the index of the value p in the array S after the round i in the KSA. For example $S_i^{-1}[S_i[k]] = k$ and $S_i[S_i^{-1}[p]] = p$. Let j_i be the value of j during the round i where the rounds are indexed in accordance with i. Thus, the KSA has rounds $0, 1, \ldots, N-1$ and the PRGA has rounds $1, 2, \ldots$ Let S'_1 denotes the array S after the first round of the PRGA (i.e. S'_1 is equal to S_{N-1} with $S_{N-1}[1]$ and $S_{N-1}[S_{N-1}[1]]$ swapped). We define z_1, the first byte of the keystream as:

$$z_1 = S'_1[S'_1[1] + S'_1[S_{N-1}[1]]] = S'_1[S_{N-1}[S_{N-1}[1]] + S_{N-1}[1]] \tag{1}$$

2.2 KSA Evolution

Definition 1 (p–protected). *During the KSA process, if $S_p^{-1}[m] \leq p$, we say that the value m is p–protected.*

To illustrate this definition, we present an example with the first three rounds of some KSA process. The values in bold are i–protected. We remark that if $m = S_i[k]$ is i–protected, then $m = S_{i+1}[k]$ if and only if $j_{i+1} \neq k$. In the KSA, this happens with probability of about $1 - 1/N$.

S_i							i	j_i	i–protected values
0	1	2	3	4	...	255	Init	Init	
3	1	2	0	4	...	255	0	3	{3}
1	**3**	2	0	4	...	255	1	0	{1, 3}
1	**3**	**42**	0	4	...	255	2	42	{1, 3, 42}

During the KSA, a permutation is done between two values at the end of each round. The indices of the two swapped values are given by i and j_i. Although the value of i is predictable, the evolution of j_i depends on the secret key and may be considered as random. To facilitate the analysis of the KSA we will approximate some rounds of the KSA by an idealized version in which step 6 assigns a random byte in register j.[2] However, even if j_i is considered as random, it is possible to guarantee with a relatively high probability that some values will not be modified during the process of the KSA. We propose to redefine the Evolution lemma given by Mantin in [12]:

Lemma 2 (Evolution Lemma). *Consider an idealized KSA where j is picked randomly for the last $(N - p)$ rounds. Let $\mathcal{I}$ be a set of p–protected values of cardinality x. The probability that no element of $\mathcal{I}$ is swapped during the last $(N - p)$ rounds of the KSA is*

$$P(x, p) = \left(\frac{N - x}{N}\right)^{N-p}$$

Furthermore, if $\mathcal{I}$ is a set of $(p-1)$–protected values and $\mathcal{J}$ is a non-intersecting set of p–protected values, the probability that no element of $\mathcal{I}$ is swapped during round p and no element of $\mathcal{I} \cup \mathcal{J}$ is swapped during the last rounds is

$$P(\#\mathcal{I}, \#\mathcal{J}, p) = \frac{N - \#\mathcal{I}}{N}\left(\frac{N - \#\mathcal{I} - \#\mathcal{J}}{N}\right)^{N-p}$$

2.3 Description of WEP

According to [1], WEP uses RC4 with $N = 256$ and $n = 8$ to provide confidentiality. The key contains a 24-bit long IV concatenated to a secret key of 40 or 104 bits. Thus, the complete key size is either 64 or 128 bits. Consider a 64-bit (8 bytes) key size:

$$K = K[0]||K[1]||K[2]||K[3]||\ldots||K[7] = IV_0||IV_1||IV_2||K[3]||\ldots||K[7]$$

[2] This approximation was also used by Mironov in [20].

where IV_i represents the i^{th} byte of the IV and $K[3]||\ldots||K[7]$ the secret part of the key. In theory, the value of the IV should be random but in practice, it is a counter mostly in *little-endian* and incremented by one each time a new 802.11b frame is encrypted. Thus, each packet uses a slightly different key. The key K is used by RC4 and the resulting keystream is XORed with the plaintext to obtain the ciphertext. Unfortunately, a portion of the plaintext is practically constant [21] and some of the following bytes can be derived. They correspond to the LLC header and the SNAP header and some bytes of the TCP/IP encapsulated frame. For example, by XORing the first byte of the ciphertext with the constant value `0xAA`, we obtain the first byte of the keystream.

2.4 Description of WPA

WPA has been designed for use with an IEEE 802.1X authentication server with the aim to distribute different keys to each user. However, it can also be used in a lightweighted mode called "pre-shared key" (WPA-PSK), where every user is given the same key. According to [17], each user must enter a pass-phrase to access the network. The pass-phrase may be from 8 to 63 printable ASCII characters or 64 hexadecimal digits. The major improvement in WPA over WEP is the Temporal Key Integrity Protocol (TKIP), a key management scheme to avoid key reuse. TKIP is a key scheduling in two phases used to generate a completely different RC4 key for each transmitted packet (called *Per Packet Key*). Thus, even if the attacks based on weak IVs and the Klein attack still exist, an attacker will have only one trial to recover a specific RC4 secret key. Moreover, a filter avoids the use of some weak IV classes (but only the weak IV class discovered by Fluhrer Mantin and Shamir in [3]). In addition, WPA also provides packet integrity which prevents replay attacks being executed. Thus, only passive key recovery attacks are theoretically applicable to WPA.

2.5 The Fluhrer Mantin and Shamir (FMS) Attack

To understand how key recovery attacks work, we briefly present the FMS attack. According to [3], this attack uses the property of some specific IV values called *weak keys*. Let $IV_0 = 3, IV_1 = 255$ and IV_2 equals some arbitrary value $x \notin \{251, 252\}$. We assume that j_3 is different from $\{0, 1\}$. Our goal is to obtain the value of the first secret byte of the key $K[3]$. Due to the assumption on x, $x + 5$ is different from $\{0, 1\}$. Together with the assumptions on j_3, we obtain that the first four rounds of the KSA are given by:

S_i							i	j
0	1	2	3	4	$\cdots$	255		
3	1	2	**0**	4	$\cdots$	255	0	$0 + 0 + IV_0 = 3$
3	**0**	2	**1**	4	$\cdots$	255	1	$3 + S_0[1] + IV_1 = 3$
3	0	$\boldsymbol{x+5}$	1	4	$\cdots$	255	2	$3 + S_1[2] + IV_2 = x + 5$
3	0	$\cdot$	$\boldsymbol{S_2[j_3]}$	$\cdot$	$\cdots$	255	3	$x + 5 + S_2[3] + K[3] = j_3$

Because $K[3]$ is unknown, we cannot predict the values of $S[i]$, $i \in \{3, \ldots, 255\}$ after the round 2, however they will eventually change, according to the KSA. Now we suppose that $S_2[0] = S_3[0] = S_{255}[0] = 3$, $S_2[1] = S_3[1] = S_{255}[1] = 0$ and $S_3[3] = S_{255}[3] = S_2[j_3]$. Following the Evolution lemma with $\mathcal{I} = \{0, 3\}$, $\mathcal{J} = \{S_2[3]\}$ and $p = 3$, the probability that $j_3 \neq \{0, 1\}$ and $S_2[j_3]$ remains at the same place is $P(2, 1, 3) \approx 5\%$. According to our assumptions and (1), the first byte of the keystream generated by the PRGA is:

S							i	j
3	0	·	$S_2[j_3]$	·	$\cdots$	255		
0	**3**	·	$S_2[j_3]$	·	$\cdots$	255	1	$j + S_{255}[i] = 0 + 0 = 0$

$$z_1 = S'_1[S'_1[1]+S'_1[S_{255}[1]] = S'_1[S_{255}[0]+S_{255}[1]] \stackrel{5\%}{=} S_p[S_{p-1}[0]+S_{p-1}[1]] = S_2[j_3]$$

Note that $z_1 \neq \{0, 3\}$ when this holds. We can now easily recover the secret key byte $K[3]$ because the first byte of the keystream can be recovered: $z_1 = S_2[j_3] = c_0 \oplus \texttt{0xAA}$ where c_0 is the first byte of the ciphertext and `0xAA` the first constant byte of the plaintext, the LLC header. Thus, $K[3] = S_2^{-1}[z_1] - S_2[3] - j_2 = S_2^{-1}[z_1] - x - 6$. We notice that the weak IV class given by $IV_0 = 3$, $IV_1 = 255$ and $IV_2 = x$ can be described as a specific S_2 table state class where $S_2[1] = 0$ and $S_2[0] = 3$.

We can generalize the FMS attack: we need a large amount of encrypted packets where the value of the IV gives the state S_{p-1} such that $S_{p-1}[1] = 0, S_{p-1}[0] = p$ and $z_1 \neq \{0, p\}$. This defines the weak IV class which recovers the secret key byte $K[p]$. The secret key byte is rederived with a probability of success $P_{\mathsf{FMS}}(p) = P(2, 1, p)$ according to the Evolution lemma.

$$\begin{aligned} K[p] &\stackrel{P_{\mathsf{FMS}}(p)}{=} S_{p-1}^{-1}[z_1] - S_{p-1}[p] - j_{p-1} \\ &= S_{p-1}^{-1}[z_1] - \sum_{j=1}^{p} S_{j-1}[j] - \sum_{i=0}^{p-1} K[i] \end{aligned} \tag{2}$$

The attacker will then collect the probed values for $K[p]$, according to (2) and finally select the one with the highest vote. Note that to rederive the secret key byte $K[p]$, the attack must successfully recover the previous bytes $K[p-1], \ldots, K[3]$ in order to compute $S_{p-1}[p]$ and j_{p-1}.

Nowadays, there are dozens of known key recovery attacks similar to the FMS attack. In order to have a relevant list of the known attacks, one has to read [22] or the source code of the tool Aircrack [11] or Weplab [10]. These attacks are divided into three categories. The first kind of attack uses only z_1 and the state of the array S_{p-1} of the KSA to recover the secret key (typically the FMS attack). The second one uses z_1 and z_2. Note that they can easily be extended to the combination of every known z_i to provide more weak IV classes.The last one highlights the unprobable secret key bytes, they are called *negative attacks*.

2.6 The Klein Attack

In 2006, Andreas Klein presented in [14] a practical application of the Jenkins correlation [13] to WEP.

Theorem 3. *Let S'_i be the i^{th} step of the PRGA where the internal state is a random permutation, and a random value j,*

$$\Pr\left(z_i + S'_i[j] \bmod N = c\right) = \begin{cases} \frac{2}{N} & \text{if } c = i \\ \frac{N-2}{N(N-1)} & \text{if } c \neq i \end{cases}$$

Klein demonstrated a strong correlation in the 7^{th} step of the PRGA which is not related to a specific weak IV class. It means that each eavesdropped packet may rederive the secret key.

$$S'_i[j] \overset{P_j}{=} i - z_i \qquad \text{(From Theorem 3 with } P_j = 2/N) \qquad (3)$$

$$S'_{i-1}[i] \overset{P'}{=} S_i[i] \qquad P' = ((N-1)/N)^{N-2} \qquad (4)$$

$$S'_i[j] = S'_{i-1}[i] \qquad \text{(step 6 of the PRGA)} \qquad (5)$$

$$S_i[i] = S_{i-1}[j_i] \qquad \text{(KSA)} \qquad (6)$$

$$j_i = S_{i-1}[i] + j_{i-1} + K[i] \qquad \text{(step 6 of the KSA)} \qquad (7)$$

From (3) with respectively (4), (5), (6) and (7) we have

$$K[p] \overset{P_{\mathsf{Klein}}}{=} S_{p-1}^{-1}\left[p - z_p \bmod N\right] - S_{p-1}[p] - j_{p-1} \bmod N \qquad (8)$$

which hold with a probability

$$P_{\mathsf{Klein}} = \frac{2}{N} \cdot \left(\frac{N-1}{N}\right)^{N-2} + \frac{N-2}{N(N-1)} \cdot \left(1 - \left(\frac{N-1}{N}\right)^{N-2}\right) \approx \frac{1.36}{N} \qquad (9)$$

A significant limitation of the Klein key recovery attack is that to recover the secret key byte $K[i]$, the i^{th} byte of the keystream has to be known.

3 The VX Attack: How to Reduce the Secret Key Bytes Dependency

A major issue related to all key recovery attacks is that if a secret key byte has not been correctly recovered, the whole key will be probably mis-recovered due to the key byte dependency (to rederive $K[i]$, the previous secret key bytes $K[i-1], K[i-2], \ldots, K[3]$ must be successfully recovered). In this section we present a new attack, called VX, able to recover more efficiently the secret key by significantly reducing the key bytes dependency.

3.1 The FMS Key Recovery Attack

The paradigm of this attack is to recover independently the sum of the secret key bytes by computing some predictable parts of the equation (2). Consider the FMS attack described above. According to (2), we obtain

$$\sum_{i=0}^{p} K[i] \overset{P_{\mathsf{FMS}}(p)}{=} S_{p-1}^{-1}[z_1] - \sum_{j=1}^{p} S_{j-1}[j] \tag{10}$$

Consider that we only know the state S_2 specified by the IV. We define $P_1(p)$, the probability that $S_{p-1}^{-1}[z_1] = S_2^{-1}[z_1]$ in the idealized version of the KSA by

$$P_1(p) = \Pr\left(S_{p-1}^{-1}[z_1] = S_2^{-1}[z_1]\right) = \left(\frac{N-1}{N}\right)^{p-2}, p \geq 2 \tag{11}$$

The array S_j with $j = 0, 1, \ldots, p-1$ is partially known if p is small, because it is close to the initialization state of S at the beginning of the KSA where $S[i] = i$. Thus the sum $\sum_{j=1}^{p} S_{j-1}[j]$ is equivalent to $S_0[1] + S_1[2] + \sum_{j=3}^{p} S_2[j]$ with a probability $P_2(p)$.[3]

$$\begin{aligned} P_2(p) &= \Pr\left(\sum_{j=1}^{p} S_{j-1}[j] = S_0[1] + S_1[2] + \sum_{j=3}^{p} S_2[j]\right) \\ &\approx \prod_{m=3}^{p} \left(\frac{N-p+m}{N}\right), p \geq 3 \end{aligned} \tag{12}$$

Thus, we can recover independently each sum of the key bytes $K[0 \ldots p]$, where $K[i \ldots j] = K[i] + K[i+1] + \ldots + K[j], j \geq i$ with a probability of success $P_{\mathsf{VX}_F}(p) = P_{\mathsf{FMS}}(3) \cdot P_1(p) \cdot P_2(p)$ and $p = 3, 4, \ldots, \ell - 1$. Indeed, using (11) and (12) in (10), we have

$$K[3 \ldots p] \overset{P_{\mathsf{VX}_F}(p)}{=} S_2^{-1}[z_1] - S_0[1] - S_1[2] - \sum_{j=3}^{p} S_2[j] - \sum_{v=0}^{2} IV_v \tag{13}$$

since the key bytes $K[0], K[1]$ and $K[2]$ are known and different for each packet because they correspond to the IV, we store the votes for the secret and fixed part of the key, the sum $K[3 \ldots p]$. Equation (13) is a correlation between a byte depending on the secret key only and a byte which can be computed from the 802.11b frame only. Finally, each secret key byte $K[i]$ can be recovered with

$$K[p] = K[3 \ldots p] - K[3 \ldots (p-1)]$$

3.2 The Klein Key Recovery Attack

The same technique can be applied to the Klein key recovery attack. Indeed, the dependency is based on the same values, only the probability P_{FMS} is different. According to (8) we have,

[3] This is an improvement of the correlation discovered by Roos in [4].

$$\sum_{i=0}^{p} K[i] \overset{P_{\mathsf{Klein}}(p)}{=} S_{p-1}^{-1}[p - z_p \bmod N] - \sum_{j=1}^{p} S_{j-1}[j]$$

Thus, we can apply the same technique described above and we obtain that

$$K[3 \dots p] \overset{P_{\mathsf{VX}_{\underline{K}}}(p)}{=} S_2^{-1}[p - z_p \bmod N] - S_0[1] - S_1[2] - \sum_{j=3}^{p} S_2[j] - \sum_{v=0}^{2} IV_v \tag{14}$$

where

$$P_{\mathsf{KleinTot}} = P_1(p) \cdot P_2(p) \cdot \left(\frac{N-1}{N}\right)^{N-2}$$

$$P_{\mathsf{VX}_K}(p) = \frac{2}{N} \cdot P_{\mathsf{KleinTot}} + \frac{N-2}{N(N-1)} \cdot (1 - P_{\mathsf{KleinTot}})$$

Note that for some values of the key bytes, the Klein attack may not work.

4 Weaknesses in the Modulo Operation of the KSA

During the KSA of RC4, the key is used modulo its size. It means that the secret key byte $K[i] = K[i + k\ell]$, where ℓ is the size of the key, $k = 1, 2, \dots, m$ and $i = 3, \dots, \ell - 1$. We remark that if an attacker is unable to recover the secret key byte $K[i]$ (because not enough frames were captured from its weak IV class or because the keystream byte needed to recover the secret key byte is unknown), he could be interested to recover the key byte $K[i + \ell]$ (through another weak IV class or another keystream byte) instead of $K[i]$. Due to the key bytes dependency, this property was irrelevant for the existing key recovery attacks. Indeed, the whole secret key had to be recovered to attack the modulo repetition.

4.1 Weakness in the Repetition of the Secret Key

According to the VX attack, it is possible to recover *independently* the value of the secret key bytes sum $K[3 \dots p]$ where $p = i + k\ell, k = 0, 1, \dots .m$ and $i = 3, \dots \ell - 1$. Consider the FMS attack described above and the equation (13). We define,

$$\begin{aligned} \overline{K}[p] &\triangleq K[3 \dots i] + k \cdot K[3 \dots (\ell - 1)] \\ &\overset{P_{\mathsf{VX}_{\underline{F}}}(p)}{=} S_2^{-1}[z_1] - S_0[1] - S_1[2] - \sum_{j=3}^{p} S_2[j] - (k+1) \cdot \sum_{v=0}^{2} IV_v \end{aligned} \tag{15}$$

If an attacker has not enough weak IV to recover the sum $K[3 \dots i]$ but he is able to rederive correctly $K[3 \dots (\ell - 1)]$, the targeted sum can be recovered when a vote for $\overline{K}[p]$ is collected, according to (15) with $k \geq 1$. The same technique can be used with the Klein attack when the keystream byte needed to recover the secret key byte is unknown.

4.2 Weakness in the Repetition of the IV

In the previous section, we have seen that the key repetition can be used to recover a part of the secret key if the sum $K[3\dots(\ell-1)]$ has been correctly rederived. An interesting feature of WEP is that the three first repeated bytes of the key are publicly disclosed, they correspond to the IV. Because these values are known, they can be used to recover more efficiently the critical secret key bytes sum $K[3\dots(\ell-1)]$. For $p = i + k\ell$ with $i = \{0, 1, 2\}$ we define $\overline{K}[p] = k \cdot K[3\dots(\ell-1)]$. Thus,

$$\begin{aligned}\overline{K}[p] &\triangleq k \cdot K[3\dots(\ell-1)] \\ &\overset{P_{\mathsf{Vx}_F}(p)}{=} S_2^{-1}[z_1] - S_0[1] - S_1[2] - \sum_{j=3}^{p} S_2[j] - k \cdot \sum_{j=0}^{2} IV_j - \sum_{j=0}^{i} IV_j \quad (16)\end{aligned}$$

Thus, four weak IV classes, instead of only one are dedicated to the recovery of the critical sum above. The same technique can be used with the Klein attack: four different keystream bytes may rederive the secret key sum. This finally leads us to many attack on byte $\overline{K}[p]$ where all bytes are linked by,

$$\overline{K}[p] = \begin{cases} k \cdot K[3\dots(\ell-1)] + K[3\dots(p \bmod \ell)] & \text{for } p \bmod \ell = 3\dots\ell-1 \\ k \cdot K[3\dots\ell-1] & \text{for } p \bmod \ell = 0, 1, 2 \end{cases} \quad (17)$$

5 Attack Principle

The principle of the attack is composed of three parts. The first one collects the IVs and the known keystream bytes of the passively eavesdropped 802.11 packets. Note that some keystream bytes are unknown (the Appendix A gives the probable plaintext bytes, for TCP and ARP packets, needed to recover the keystream bytes). For each known keystream byte z_p, the extended Klein attack described above will return a probed byte n for the sum of secret key bytes $\overline{K}[p]$ weighted by P_v the success probability of the vote. The key recovery attacks based on the IV are similarly used, by using the IV and the two first bytes of the keystream z_1 and z_2.

Once the vote process is accomplished, we use two techniques to rederive more efficiently the secret key sum $\overline{K}[\ell-1]$. Firstly we take profit of the modulo repetition of the IV according to (16). Secondly, we do an autocorrelation on the r discrete signals $\overline{K}[3] + k \cdot \overline{K}[\ell-1] || \overline{K}[4] + k \cdot \overline{K}[\ell-1] || \dots || \overline{K}[\ell-2] + k \cdot \overline{K}[\ell-1]$ $k = 0, 1, \dots, r$ where the time shifting value corresponds to $\overline{K}[\ell-1]$. When the autocorrelation is maximized for a given $\overline{K}[\ell-1]$, it is considered as the most probable value. We merge the results given by the autocorrelation for each potential value of $\overline{K}[\ell-1]$ with the votes given by (16) and we sort them according to their votes. Once $\overline{K}[\ell-1]$ is fixed we compute the votes for the repeated secret keys and we merge all the votes.

Finally, we successively test the first M secret keys, according to their distance to the most probable value (with the highest amount of vote). Note that each time a new value for $\overline{K}[\ell-1]$ is selected, we have to recompute the votes for all the repetition of the secret key bytes. See Algorithm 3 for more details.

Algorithm 3. VX Key Recovery Attack

VX(IV, Z)**:** IV, the set of known keystream bytes Z where z_i is the i^{th} byte of the keystream.
Data: V is a $(N \times (\ell \cdot m - 3))$ matrix
Data: V' is a $(N \times (\ell - 1))$ matrix
Output: The secret key K

for each passively eavesdropped packet $\{IV, Z\}$ **do**
  $(n, p, P_v) \leftarrow$ WeakAttack(IV, z_1, z_2)
  $V_{n,p} \leftarrow V_{n,p} + P_v$
  $(n, p, P_v) \leftarrow$ KleinAttack(IV, Z)
  $V_{n,p} \leftarrow V_{n,p} + P_v$
end for
for each r repetition of the secret key **do**
  for $n = 0$ to $N - 1$ **do**
    $V_{n,r\cdot\ell-1} \leftarrow V_{n.r\cdot\ell-1} + V_{n.r\cdot\ell} + V_{n.r\cdot\ell+1} + V_{n.r\cdot\ell+2}$
  end for
end for
$V \leftarrow$ Autocorrelation(V)
$V' \leftarrow$ MergeVotes(V)
for $i = 0$ to M **do**
  pick K the next most probable secret key in V'
  if K uses another value for $\overline{K}[\ell - 1]$ **then**
    $V \leftarrow$ Autocorrelation(V)
    $V' \leftarrow$ MergeVotes(V)
  end if
  if K is correct **return** K
end for

5.1 Practical Results on WEP

To demonstrate the improvement of the VX attack, we tried to recover randomly generated WEP 104-bit secret keys with a limited number of frames and randomly chosen IVs.

A first issue concerning the Klein attack, which is more efficient than the key recovery attacks based on weak IVs, is the ability to obtain the plaintext. Thus, we firstly concentrated our analysis on passively eavesdropped ARP frames because the plaintext of an ARP frame can be practically guessed until the 32^{nd} byte (see Appendix A). Then, we chose a more realistic scenario where the eavesdropped traffic is mainly based on TCP frames (we used real network traffic dumps for this scenario). According to Appendix A, when a TCP frame is passively eavesdropped, the first and the second byte of the keystream, used by the key recovery attacks based on weak IVs are practically always known. However, the following keystream bytes needed for the Klein attack cannot be completely recovered. By significantly reducing the key bytes dependency and according to the modulo repetition, the VX attack is able to recover the secret key, even if some keystream bytes are still unknown.

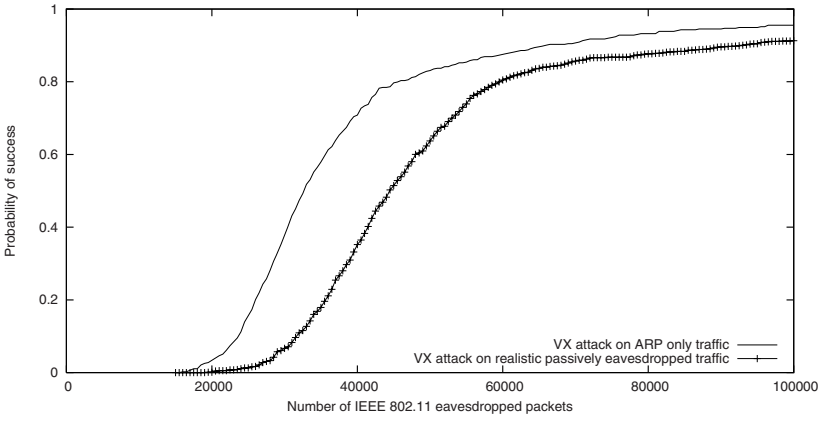

Fig. 1. The probability to recover the correct key after an exhaustive search of 2^{20} trials, according to the number of passively eavesdropped packets

The table (Figure 1) gives the average number of ARP frames needed to recover the complete secret key with an exhaustive search on a keyspace subset of $M = 2^{20}$ entries (with the highest probability of success, according to our votes). We notice that the average amount of ARP packets needed to recover the secret key with a probability $> 1/2$ is 32,700. The same table gives, according to the second scenario, the average number of frames needed to recover the complete secret key with an exhaustive search on a subset keyspace of $M = 2^{20}$ entries (with the highest probability of success, according to our votes). We notice that the average amount of packets needed to recover the secret key for the same probability is 44,500.

If we compare the VX attack with the previously published passive-only key recovery attacks on WEP [10,11], we reduce the data complexity from 2^{20} to 2^{15} for the same success probability to recover the secret key. We significantly reduce the computational complexity as well because the recomputation of the votes is not needed for each key trial.

Moreover, the VX attack can be transformed to an active one and needs about 25% less eavesdropped packets than the attack described in [19] thanks to the weaknesses in the modulo repetition and the use of the enhanced key recovery attacks based on weak IVs.

6 Conclusion

In this paper, we have seen that all the previously discovered key recovery attacks (the Klein attack as well as the key recovery attacks based on weak IVs) suffer from a relation between the secret key bytes. To rederive the i^{th} byte of the secret key, we have to successfully recover the $(i-1)$ previous key bytes. In practice, this constraint is a significant limitation because if the key recovery

process does not work for only one byte of the key, the complete key will be probably mis-recovered.

According to the VX attack presented in this paper, we are now able to significantly reduce the key bytes dependency and thus, to recover correctly each key byte with a stronger probability, even if some preceding secret key bytes are still unknown.

Because the i^{th} byte of the secret key can be recovered even if some previous bytes are missing, new weak IV classes appear.

Moreover, it becomes possible to take profit of the modulo repetition weaknesses of the secret key in the KSA of RC4 described in this paper, to improve the global key recovery process.

We showed that the Klein attack needs to know the $(i-1)^{th}$ byte of the keystream to recover the i^{th} byte of the secret key. However, this information cannot always be obtained with passive-only key recovery attacks. Associated to the enhanced attacks based on weak IVs and the modulo repetition weaknesses of the secret key (both presented in this paper), a part of the missing secret key bytes can be passively recovered.

Consequently, the VX attack is to the best of our knowledge, the most efficient passive-only key recovery attack on WEP. The previous ones needed about one million of passively eavesdropped packets to recover the secret key with a probability bigger than one half. The VX attack needs about 44,500 packets for the same success probability.

A question raised in this paper is the motivation to find new key recovery attacks on WEP: a still widely used protocol, but already broken since 2001. The weaknesses highlighted in this paper concern theoretically WPA as well. Indeed, only passive attacks are applicable on WPA because of anti-replay protections.In spite of the fact that the VX attack cannot be practically exploited on WPA, it represents a relevant first step for its analysis.

References

1. IEEE: ANSI/IEEE standard 802.11b: Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications (1999)
2. Borisov, N., Goldberg, I., Wagner, D.: Intercepting mobile communications: the insecurity of 802.11. In: MOBICOM, pp. 180–189 (2001)
3. Fluhrer, S., Mantin, I., Shamir, A.: Weaknesses in the key scheduling algorithm of RC4. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 1–24. Springer, Heidelberg (2001)
4. Roos, A.: A class of weak keys in RC4 stream cipher (sci.crypt) (1995)
5. Wagner, D.: Weak keys in RC4 (sci.crypt) (1995), `http://www.cs.berkeley.edu/~daw/my-posts/my-rc4-weak-keys`
6. Hulton, D.: Practical exploitation of RC4 weaknesses in WEP environments (2001), `http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt`
7. Bittau, A.: Additional weak IV classes for the FMS attack (2003), `http://www.cs.ucl.ac.uk/staff/a.bittau/sorwep.txt`
8. Korek: Need security pointers (2004), `http://www.netstumbler.org/showthread.php?postid=89036#post89036`

9. Korek: Next generation of WEP attacks? (2004), http://www.netstumbler.org/showpost.php?p=93942&postcount=35
10. Martin, J.I.S.: Weplab, http://weplab.sourceforge.net/
11. Devine, C., Otreppe, T.: Aircrack, http://www.aircrack-ng.org/
12. Mantin, I.: A practical attack on the fixed RC4 in the WEP mode. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 395–411. Springer, Heidelberg (2005)
13. Jenkins, R.: Isaac and RC4, http://burtleburtle.net/bob/rand/isaac.html
14. Klein, A.: Attacks on the RC4 stream cipher. Personal Andreas Klein website (2006), http://cage.ugent.be/~klein/RC4/RC4-en.ps
15. Bittau, A., Handley, M., Lackey, J.: The final nail in WEP's coffin. In: S&P, pp. 386–400. IEEE Computer Society Press, Los Alamitos (2006)
16. Paul, G., Rathi, S., Maitra, S.: On non-negligible bias of the first output bytes of RC4 towards the first three bytes of the secret key. In: WCC 2007. International Workshop on Coding and Cryptography, pp. 285–294 (2007)
17. IEEE: ANSI/IEEE standard 802.11i: Amendment 6 Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications, Draft 3 (2003)
18. IEEE: ANSI/IEEE standard 802.11i: Amendment 6: Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications (2004)
19. Tews, E., Weinmann, R.P., Pyshkin, A.: Breaking 104 bit WEP in less than 60 seconds. Cryptology ePrint Archive, Report 2007/120 (2007), http://eprint.iacr.org/
20. Mironov, I.: (Not so) random shuffles of RC4. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 304–319. Springer, Heidelberg (2002)
21. Postel, R.: Rfc1042 (1988) http://rfc.net/rfc1042.html
22. Chaabouni, R.: Breaking WEP Faster with Statistical Analysis. Ecole Polytechnique Fédérale de Lausanne, LASEC, Semester Project (2006)

A Appendix

ARP Packet	
0xAA	DSAP
0xAA	SSAP
0x03	CTRL
0x00 0x00 0x00	ORG Code
0x08 0x06	ARP
0x00 0x01	Ethernet
0x08 0x00	IP
0x06	Hardware size
0x04	Protocol
0x00	Opcode Request/Reply
0x??	
0x?? 0x?? 0x?? 0x?? 0x?? 0x??	MAC addr src
0x?? 0x?? 0x?? 0x??	IP src
0x?? 0x?? 0x?? 0x?? 0x?? 0x??	MAC addr dst
0x?? 0x?? 0x?? 0x??	IP dst

TCP Packet	
0xAA	DSAP
0xAA	SSAP
0x03	CTRL
0x00 0x00 0x00	ORG Code
0x08 0x00	IP
0x45	IP Version + Header length
0x?? 0x??	Packet length
0x?? 0x??	IP ID RFC815
0x??	Fragment type and offset
0x??	TTL
0x06	TCP type
0x?? 0x??	Header checksum
0x?? 0x?? 0x?? 0x??	IP src
0x?? 0x?? 0x?? 0x??	IP dst
0x?? 0x??	Port src
0x?? 0x??	Port dst

Fig. 2. The tables above represent the plaintext bytes of 802.11 data frames encapsulating resp. ARP and TCP protocols. The value in white are almost fixed or can be computed dynamically. The values in light grey can be guessed. The values in dark grey are not predictable.