Distinguishing WPA

Sourav Sen Gupta[1], Subhamoy Maitra[1], and Willi Meier[2]

[1] Indian Statistical Institute, Kolkata, India
[2] FHNW, Windisch, Switzerland

Abstract. We present an efficient algorithm that can distinguish the keystream of WPA from that of a generic instance of RC4 with a packet complexity of $O(N^2)$, where N denotes the size of the internal permutation of RC4. In practice, our distinguisher requires approximately 2^{19} packets; thus making it the best known distinguisher of WPA to date. This is a significantly improved distinguisher than the previous WPA distinguisher identified by Sepehrdad, Vaudenay and Vuagnoux in Eurocrypt 2011, which requires more than 2^{40} packets in practice. The motivation of our distinguisher arises from the recent observations on WPA by AlFardan, Bernstein, Paterson, Poettering and Schuldt[3], and this work puts forward an example how an experimental bias may lead to an efficient theoretical distinguisher.

Keywords: RC4, WPA, TKIP, Bias, Distinguisher, First byte, Initial bytes.

1 Introduction

RC4, also known as Alleged RC4 or ARC4, is the most widely deployed commercial stream cipher, having applications in network protocols such as SSL, WEP, WPA and in Microsoft Windows, Apple OCE, Secure SQL, etc. The cipher consists of a Key Scheduling Algorithm (KSA) and a Pseudorandom Generation Algorithm (PRGA). The internal state of RC4 is obtained as a permutation of all 8-bit words, i.e., a permutation of $N = 2^8 = 256$ bytes, and the KSA produces the initial pseudorandom permutation of RC4 by scrambling an identity permutation using the secret key k. The secret key k of RC4 is of length typically between 5 to 32 bytes, which generates the expanded key K of length $N = 256$ bytes by simple repetition. If the length of the secret key k is l bytes (typically $5 \leq l \leq 32$), then the expanded key K is constructed as $K[i] = k[i \bmod l]$ for $0 \leq i \leq N-1$. The initial permutation produced by the KSA acts as an input to the next procedure PRGA that generates the keystream. The RC4 algorithms KSA and PRGA are depicted in Figure 1.

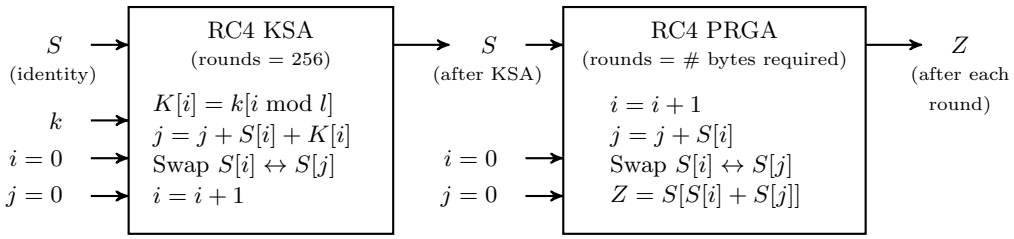

Fig. 1. Description of RC4 stream cipher.

For round $r = 1, 2, \ldots$ of RC4 PRGA, we denote the indices by i_r, j_r, the keystream output byte by Z_r, the output byte-extraction index as $t_r = S_r[i_r] + S_r[j_r]$, and the permutations before and after the swap by S_{r-1} and S_r respectively. After r rounds of KSA, we denote the state variables by adding a superscript K to each variable. All additions (subtractions) in context of RC4 are to be considered as 'addition (subtraction) modulo N', and all equalities in context of RC4 are to be considered as 'congruent modulo N'.

[3] Note that the results by AlFardan, Bernstein, Paterson, Poettering and Schuldt [1] have identified through experiments several biases in the initial keystream bytes of RC4 in TLS mode of operation (with $N = 256$ and 16-byte keys). Almost all of these TLS-related unexplored biases, except for the one at $Z_1 = 129$, have been proved in the Ph.D thesis of Sen Gupta [10] (part of a joint work [9] with Sarkar, Paul and Maitra), submitted on 12 July 2013.

1.1 Description of WPA

IEEE 802.11 standard protocol for WiFi security used to be Wired Equivalent Privacy (WEP), which has now been replaced by Wi-Fi Protected Access (WPA). Both WEP and WPA use RC4 as their core module. In case of WEP, the protocol uses RC4 with a pre-shared key appended to a public initialization vector (nonce) for self-synchronization. Using the technique of related key attacks on RC4, this scheme has been broken through passive full-key recovery attacks, and thus WEP is considered insecure in practice.

To mitigate this problem, WEP has been replaced by WPA. The goal of WPA was to resolve all security threats of WEP. However, the original WEP protocol was extensively adopted by the industry, and it was already implemented in several commercial products, both in software and hardware. This rendered a design of WPA from scratch quite impractical and costly. The work-around was to fix the full-key recovery problems of WEP using a patch, as minimal as possible, on top of the original protocol.

The WPA protocol can be thought of as a wrapper on top of WEP to provide good key mixing features. WPA introduces a key hashing module in the original WEP design to defend against the Fluhrer, Mantin and Shamir attack [3]. It also includes a message integrity feature and a key management scheme to avoid key reuse in the protocol.

TKIP key schedule. WPA uses a 16-byte secret key for RC4 PRNG, the core encryption module of the system. This RC4 secret key is generated through a key schedule procedure known as TKIP [4], which takes as input a 128-bit *temporal key* TK (shared between the parties), transmitter's 48-bit MAC address TA and a 48-bit *initialization vector* IV, and passes those through two phases to obtain the final RC4 secret key.

In Phase 1, a 80-bit key P1K is generated from TK, TA and IV32, the upper 32 bits of the IV, using an unbalanced Feistel cipher with 80-bit block and 128-bit key structure. In Phase 2, the 128-bit RC4KEY is generated from TK, P1K (from Phase 1) and IV16, the lower 16 bits of the IV. In this phase, TK and P1K are mixed (using a temporary key PPK) to construct the last 104 bits (13 bytes) of the RC4KEY, and the first 24 bits (3 bytes) of the RC4KEY are constructed directly from the IV16, as follows [4, Annex H.1].

```
RC4KEY[0] = Hi8(IV16);                    /* RC4KEY[0..2] is the WEP IV */
RC4KEY[1] = (Hi8(IV16) | 0x20) & 0x7F;    /* Help avoid FMS weak keys */
RC4KEY[2] = Lo8(IV16);
```

In the above expression, Hi8(IV16) and Lo8(IV16) indicate the top and lower bytes of IV16, respectively. RC4KEY[0] and RC4KEY[2] are simply two parts of the counter IV16, while RC4KEY[1] is purposefully constructed to avoid the known WEP attack by Fluhrer, Mantin and Shamir [3]. Once the 128-byte (16-byte) RC4KEY is prepared, it is directly used for encryption in the RC4 PRNG core of the protocol.

1.2 Distinguishing attack on WPA

Note that the best distinguisher of RC4 to date is the one based on the second-byte bias (for event $Z_2 = 0$), identified and proved by Mantin and Shamir [7] in 2001, which requires roughly $N = 2^8$ bytes to distinguish the keystream generated by RC4 from a truly random sequence of bytes. This bias is effective even if the secret key of RC4 is truly random, and thus it prevails in WPA as well. This produces a natural $O(N)$ distinguisher of WPA keystream from truly random sequence of bytes. However, the second-byte distinguisher of [7] fails to distinguish between WPA and a generic RC4-based protocol, if the bias is prominent in both cases.

> *We describe a 'distinguisher of WPA' as an algorithm that can effectively distinguish the keystream of WPA from the keystream of a generic RC4-based protocol (definition similar to [12, Section 7.3]).*

Although various security analyses of WPA are available in the literature, mostly targeted towards key-recovery of WPA using vulnerabilities of TKIP key schedule, the first distinguisher of WPA was proposed quite recently (during 2011-12) by Sepehrdad, Vaudenay and Vuagnoux [12, 13]. The distinguisher of [13], first presented in Eurocrypt 2011, achieves a 0.5 probability of success in distinguishing WPA with time complexity 2^{43} and packet complexity 2^{40}. Later in [12], the distinguisher was improved to achieve 0.5 probability of success in distinguishing WPA with time complexity 2^{42} and packet complexity 2^{42}.

1.3 Contribution of this paper

We present an efficient algorithm that can distinguish the keystream of WPA from that of a generic instance of RC4 with a packet complexity of $O(N^2)$. In practice, our distinguisher requires approximately 2^{19} packets; thus making it the best known distinguisher of WPA to date. We clearly improve the previous WPA distinguisher identified by Sepehrdad, Vaudenay and Vuagnoux in Eurocrypt 2011 [13], which requires approximately 2^{40} packets in practice, and its revised version in [12] that requires 2^{42} packets to reduce the time complexity. The motivation of our distinguisher arises from the recent experimental observations on WPA by AlFardan, Bernstein, Paterson, Poettering and Schuldt [1], and this work puts forward an example how an experimental bias may lead to an efficient theoretical distinguisher in case of RC4-based protocols.

2 Biases in WPA resulting from TKIP

Equation (1) summarizes the construction of the first three bytes of the RC4 secret key in WPA/TKIP.

$$K[0] = (\mathsf{IV16} >> 8)\ \&\ \mathsf{0xFF} \qquad K[1] = ((\mathsf{IV16} >> 8)\ |\ \mathsf{0x20})\ \&\ \mathsf{0x7F} \qquad K[2] = \mathsf{IV16}\ \&\ \mathsf{0xFF} \qquad (1)$$

Note that only a 16-bit (2-byte) IV16 is expanded to the initial 3 bytes of the key, and the first two bytes $K[0]$ and $K[1]$ have quite a few bits in common. Specifically, the expansion of IV16 is as shown in Fig. 2, and one may note that 6 bits are shared by $K[0]$ and $K[1]$, apart from the two fixed bits in $K[1]$. The third key-byte, $K[2]$ however, is independent of the first two bytes of the key. Thus, TKIP can generate only 2^{16}, and not 2^{24}, distinct values of the first 3 bytes of the RC4 secret key – a loss in entropy that we believe may result into some non-random behavior in the initial phases of the cipher.

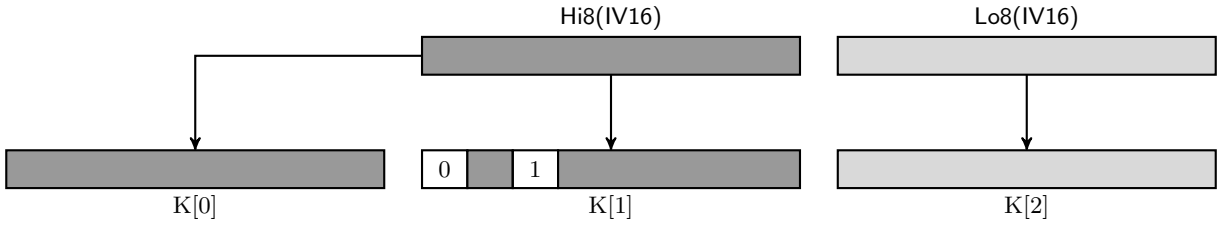

Fig. 2. Expansion of WPA IV16 into the first three bytes of the RC4 secret key.

2.1 Bias in $K[0] + K[1]$ for WPA/TKIP

As $K[0]$ and $K[1]$ share 6 bits from the common source Hi8(IV16), we first take a look at their sum, $K[0]+K[1]$, for potential non-randomness. We notice the following pattern in this direction.

1. The value of $K[0] + K[1]$ must always be *even*, as $K[0]$ and $K[1]$ have the same LSB.
2. The value of $K[1]$ can never exceed 127 as the MSB is 0. The value can not attain all possible numbers below 127 either, as the 6-th bit (from LSB side) is fixed at 1.
3. Value of $K[1]$ and hence $K[0] + K[1]$ strictly depend on the value and range of $K[0]$.

The above restrictions result in corresponding conditions on the range of $K[1]$ and $K[0]+K[1]$, depending on the range of $K[0]$. The complete set of conditions on the respective ranges is shown in Table 1, which results in a consolidated probability distribution of $K[0] + K[1]$, as follows.

Theorem 1. *The probability distribution of the sum of first two bytes of the RC4 key generated by TKIP key schedule in WPA, i.e., the distribution of* $\Pr(K[0] + K[1] = v)$ *for* $v = 0, 1, \ldots, 255$*, is as in Table 1:*

$$\begin{aligned}
&\Pr(K[0] + K[1] = v) = 0 && \text{if } v \text{ is odd};\\
&\Pr(K[0] + K[1] = v) = 0 && \text{if } v \text{ is even and } v \in [0, 31] \cup [128, 159];\\
&\Pr(K[0] + K[1] = v) = 2/256 && \text{if } v \text{ is even and } v \in [32, 63] \cup [96, 127] \cup [160, 191] \cup [224, 255];\\
&\Pr(K[0] + K[1] = v) = 4/256 && \text{if } v \text{ is even and } v \in [64, 95] \cup [192, 223].
\end{aligned}$$

Proof. The value of $K[0] + K[1]$ is always even, as discussed earlier. The value and range of $K[1]$, and hence that of $K[0] + K[1]$, depends on the range of $K[0]$; shown in Table 1. The probability distribution of $K[0] + K[1]$ may be calculated directly from this dependence pattern; also shown in Table 1. One may check

$$\underbrace{(128 \times 0)}_{\text{odd values}} + \left(16 \times 0 + 16 \times \frac{2}{256} + 16 \times \frac{4}{256} + 16 \times \frac{2}{256} + 16 \times 0 + 16 \times \frac{2}{256} + 16 \times \frac{4}{256} + 16 \times \frac{2}{256}\right) = 1,$$

to validate the consistency of the probability distribution of $K[0] + K[1]$, as depicted in Table 1. □

Table 1. Probability distribution of $K[0] + K[1]$ resulting due to TKIP key scheduling in WPA.

$K[0]$	$K[1]$ (depends on $K[0]$)		$K[0] + K[1]$ (only even)	
Range	Value	Range	Value	Range
0 – 31	$K[0] + 32$	32 – 63	$2K[0] + 32$	32 – 95
32 – 63	$K[0]$	32 – 63	$2K[0]$	64 – 127
64 – 95	$K[0] + 32$	96 – 127	$2K[0] + 32$	160 – 223
96 – 127	$K[0]$	96 – 127	$2K[0]$	192 – 255
128 – 159	$K[0] - 96$	32 – 63	$2K[0] - 96$	160 – 233
160 – 191	$K[0] - 128$	32 – 63	$2K[0] - 128$	192 – 255
192 – 223	$K[0] - 96$	96 – 127	$2K[0] - 96$	32 – 95
224 – 255	$K[0] - 128$	96 – 127	$2K[0] - 128$	64 – 127

$K[0] + K[1]$ (only even)	Probability (0 for odd)
0 – 31	0
32 – 63	2/256
64 – 95	4/256
96 – 127	2/256
128 – 159	0
160 – 191	2/256
192 – 223	4/256
224 – 255	2/256

2.2 Bias in RC4 PRGA initial permutation S_0 for WPA/TKIP

In 2008, Maitra and Paul [5] proved the famous Roos' biases [8], which states that the initial bytes of the permutation S_0, right after the completion of RC4 KSA, are biased towards certain combination of secret key bytes. We get $S_0[0]$ biased towards $K[0]$, which is uniformly distributed, identical to the lower half of the counter IV16. For $S_0[1]$ however, we get the following result.

Theorem 2. *In WPA/TKIP, the probability distribution of the second location of the RC4 permutation S_0 generated after KSA, i.e., the distribution of $\Pr(S_0[1] = v)$ for $v = 0, 1, \ldots, 255$, is given as*

$$\Pr(S_0[1] = v) = \alpha \cdot \Pr(K[0] + K[1] = v - 1) + (1 - \alpha) \cdot (1/N),$$

where $\alpha = \frac{1}{N} + \left(1 - \frac{1}{N}\right)^{N+2}$, and the term $\Pr(K[0] + K[1] = v - 1)$ can be computed using Theorem 1.

Proof. From the proof of Roos' biases in [5], the initial permutation byte $S_0[y]$ of RC4 is biased towards $f_y = \sum_{x=0}^{y} K[x] + y(y+1)/2$. In particular, for $y = 1$, we get $S_0[1]$ biased towards $K[0] + K[1] + 1$, as follows:

$$\Pr(S_0[1] = K[0] + K[1] + 1) \;\approx\; \frac{1}{N} + \left(1 - \frac{1}{N}\right)^{N+2} \;=\; \alpha, \quad \text{say.}$$

Thus, the probability distribution of $S_0[1]$ in case of WPA/TKIP is given as

$$\begin{aligned}
\Pr(S_0[1] = v) \;&=\; \Pr(S_0[1] = v \;\wedge\; S_0[1] = K[0] + K[1] + 1) + \Pr(S_0[1] = v \;\wedge\; S_0[1] \neq K[0] + K[1] + 1) \\
&=\; \Pr(S_0[1] = K[0] + K[1] + 1) \cdot \Pr(K[0] + K[1] + 1 = v) \\
&\qquad + \Pr(S_0[1] \neq K[0] + K[1] + 1) \cdot \Pr(S_0[1] = v) \\
&\approx\; \alpha \cdot \Pr(K[0] + K[1] = v - 1) + (1 - \alpha) \cdot (1/N),
\end{aligned}$$

where we have assumed that $S_0[1] = K[0] + K[1] + 1$ and $K[0] + K[1] + 1 = v$ are mutually independent, and that $S_0[1] = v$ occurs with random probability of association $1/N$ in case $S_0[1] \neq K[0] + K[1] + 1$. □

While computing for $N = 256$, as in practical WPA and RC4, $\alpha \approx 0.368$ in Theorem 2, and we have:

$$\Pr(S_0[1] = v) = 0.368 \times \Pr(K[0] + K[1] = v - 1) + 0.00246875.$$

The values of $\Pr(K[0] + K[1] = v - 1)$ in case of WPA/TKIP may be taken from Theorem 1, while in case of generic RC4, the distribution $K[0] + K[1] = v - 1$ may be assumed to be uniform as the key bytes $K[0]$ and $K[1]$ are chosen at random. This produces two different distributions of $S_0[1]$, one for generic RC4, and another for WPA/TKIP, as shown in Fig. 3.

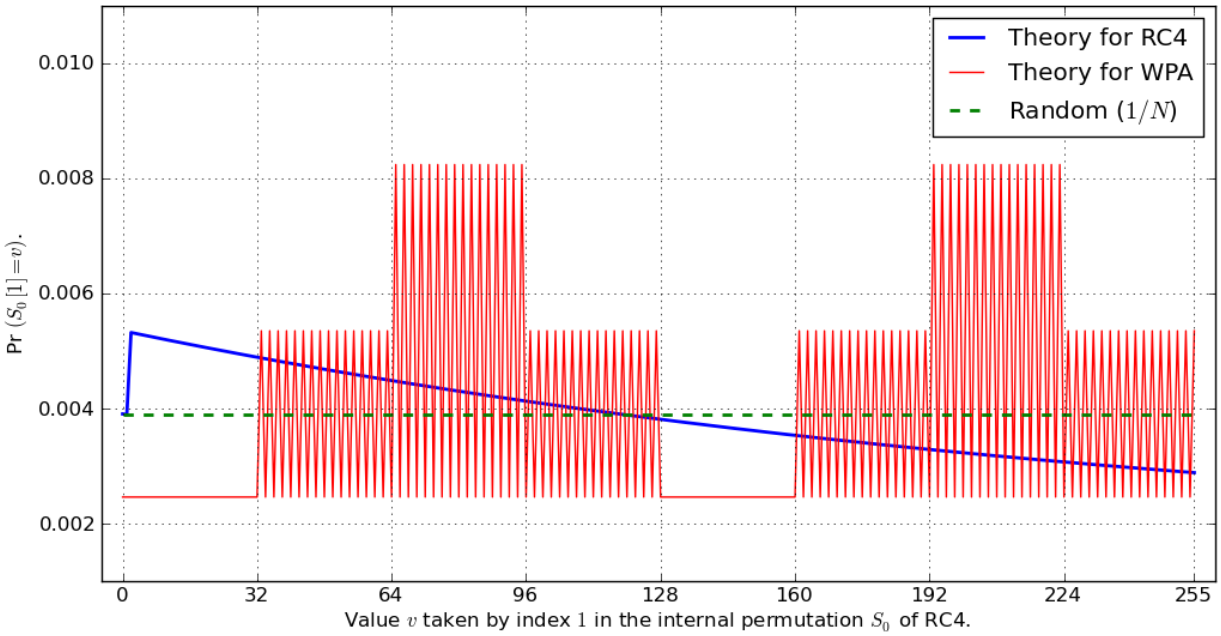

Fig. 3. Theoretical plot for $\Pr(S_0[1] = v)$ for RC4 and WPA, where $v = 0, \ldots, 255$.

2.3 Bias in the first keystream byte Z_1 of WPA/TKIP

Recall that in the first round of RC4 PRGA, the initial permutation entry $S_0[1]$ is crucial; it serves as $j_1 = S_0[i_1] = S_0[1]$, and plays an important role in determining the first keystream byte

$$Z_1 = S_1[S_1[i_1] + S_1[j_1]] = S_1[S_0[j_1] + S_0[i_1]] = S_1[S_0[S_0[1]] + S_0[1]].$$

In fact, we know that $S_0[1]$ is a vital component in the closed-form expression for Z_1, as proved by Sen Gupta, Maitra, Paul and Sarkar [11]. We reproduce the expression for Z_1 from [11, Theorem 13] as follows.

Proposition 1 (from [11]). *The probability distribution of the first output byte of RC4 keystream is as follows, where* $v \in \{0, \ldots, N-1\}$, $\mathcal{L}_v = \{0, 1, \ldots, N-1\} \setminus \{1, v\}$ *and* $\mathcal{T}_{v,X} = \{0, 1, \ldots, N-1\} \setminus \{0, X, 1-X, v\}$.

$$\Pr(Z_1 = v) = Q_v + \sum_{X \in \mathcal{L}_v} \sum_{Y \in \mathcal{T}_{v,X}} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v),$$

$$\textit{with} \quad Q_v = \begin{cases} \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0), & \textit{if } v = 0; \\ \Pr(S_0[1] = 0 \ \wedge \ S_0[0] = 1), & \textit{if } v = 1; \\ \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) & \\ \quad + \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1 - v] = v), & \textit{otherwise.} \end{cases}$$

We consider two cases while computing the numeric values of $\Pr(Z_1 = v)$. If the initial permutation S_0 of RC4 PRGA is constructed from the regular KSA with random key, the probabilities $\Pr(S_0[u] = v)$ closely follow the distribution proved by Mantin in [6, Theorem 6.2.1]. However, if the initial permutation S_0 originates from RC4 KSA using TKIP-generated keys, as in the case with WPA, then $\Pr(S_0[1] = v)$ must be computed using Theorem 2, including its idiosyncratic biases for WPA/TKIP shown in Fig. 3.

We compute the exact probabilities $\Pr(Z_1 = v)$ for generic RC4 and WPA/TKIP using the estimation strategy of joint probabilities proposed in [11], where the distribution of $S_0[1] = v$ is considered independently in each case. This results in two different distributions of Z_1; one for generic RC4 and the other for RC4 used with TKIP, as in WPA. Figure 4 displays the two distributions, clearly pointing out the bias resulting in the PRGA as a result of TKIP key schedule.

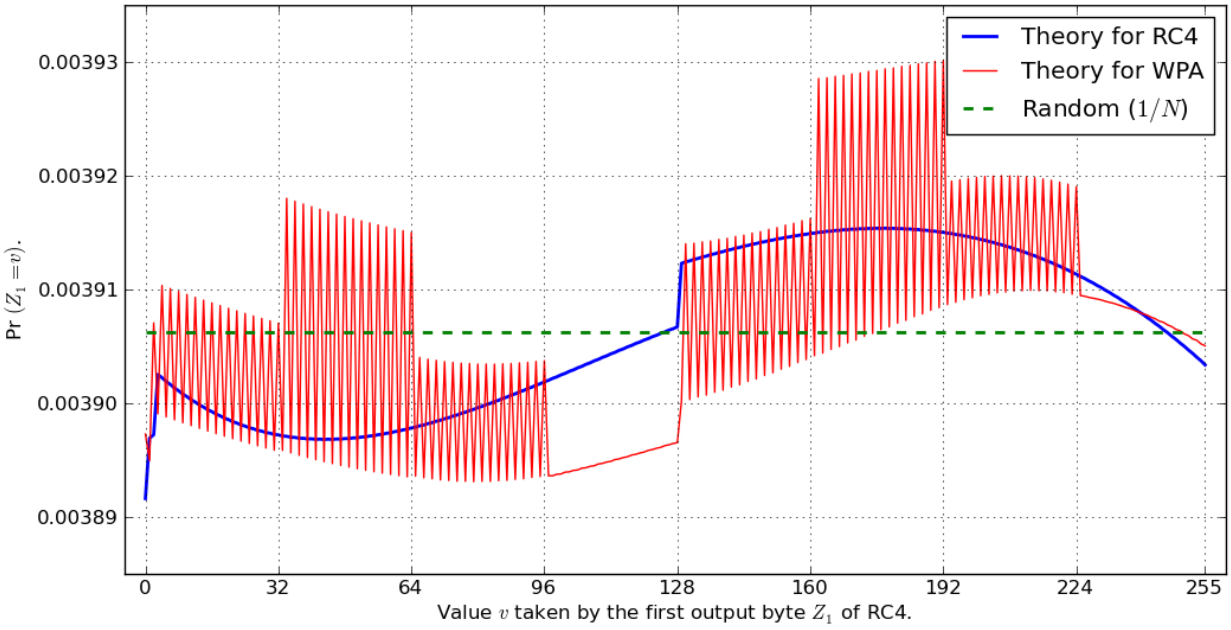

Fig. 4. Theoretical plot for $\Pr(Z_1 = v)$ for RC4 and WPA, where $v = 0, \ldots, 255$.

Note that the patterns of these two theoretical distributions closely match the recent experimental observations of AlFardan, Bernstein, Paterson, Poettering and Schuldt [1] (Fig. 10(a) in the full version of the paper, available online). The only difference is that there exist keylength dependent spikes at $Z_1 = 129$ for the observations in [1], as the experiments were done using 16-byte keys, whereas in our theoretical analysis, we disregard the keylength dependence altogether, and prove a general distribution of Z_1.

In fact, if WPA had employed RC4 with full-length 256-byte secret keys, where the first three bytes of the key $K[0], K[1], K[2]$ were constructed from the IV using TKIP key schedule principle (as in Equation (1)), the pattern of the bias in Z_1 for WPA/TKIP would have been the same. We have independently verified our theoretical results through experiments involving secret keys of various lengths.

2.4 Bias towards zero in keystream bytes $Z_3, \ldots, Z_{255}$ of WPA/TKIP

We extend the effect of the bias in S_0 to the biases in the initial keystream bytes towards zero. Sen Gupta, Maitra, Paul and Sarkar [11] proved the biases of the initial keystream bytes $Z_3, \ldots, Z_{255}$ towards zero, and we reproduce their result from [11, Theorem 14] in Proposition 2, as follows.

Proposition 2 (from [11]). *For PRGA rounds $3 \le r \le N-1$, the probability that $Z_r = 0$ is given by:*

$$\Pr(Z_r = 0) \approx \frac{1}{N} + \frac{c_r}{N^2}, \qquad \textit{where} \qquad c_r = \begin{cases} \frac{N}{N-1}\left(N \cdot \Pr(S_{r-1}[r] = r) - 1\right) - \frac{N-2}{N-1}, & \textit{for } r = 3; \\ \frac{N}{N-1}\left(N \cdot \Pr(S_{r-1}[r] = r) - 1\right), & \textit{otherwise.} \end{cases}$$

In [11], the computation of $\Pr(Z_r = 0)$ depended on the computation of $\Pr(S_{r-1}[r] = r)$, which in turn required the distribution of the initial permutations S_0 and S_1 of RC4 PRGA (refer to [11, Corollary 2] and [11, Lemma 1] for details). We consider two cases – one in which the initial permutation S_0 is generated by generic RC4 KSA using random keys, and the other where S_0 is biased (as discussed earlier) for using RC4 with keys originating from TKIP. These two cases produce two different distributions of $\Pr(Z_r = 0)$ for $r = 3, \ldots, 255$, as depicted in Fig. 5. The patterns closely match the experimental observations of AlFardan, Bernstein, Paterson, Poettering and Schuldt [1] (Fig. 11 in the full version of the paper, available online).

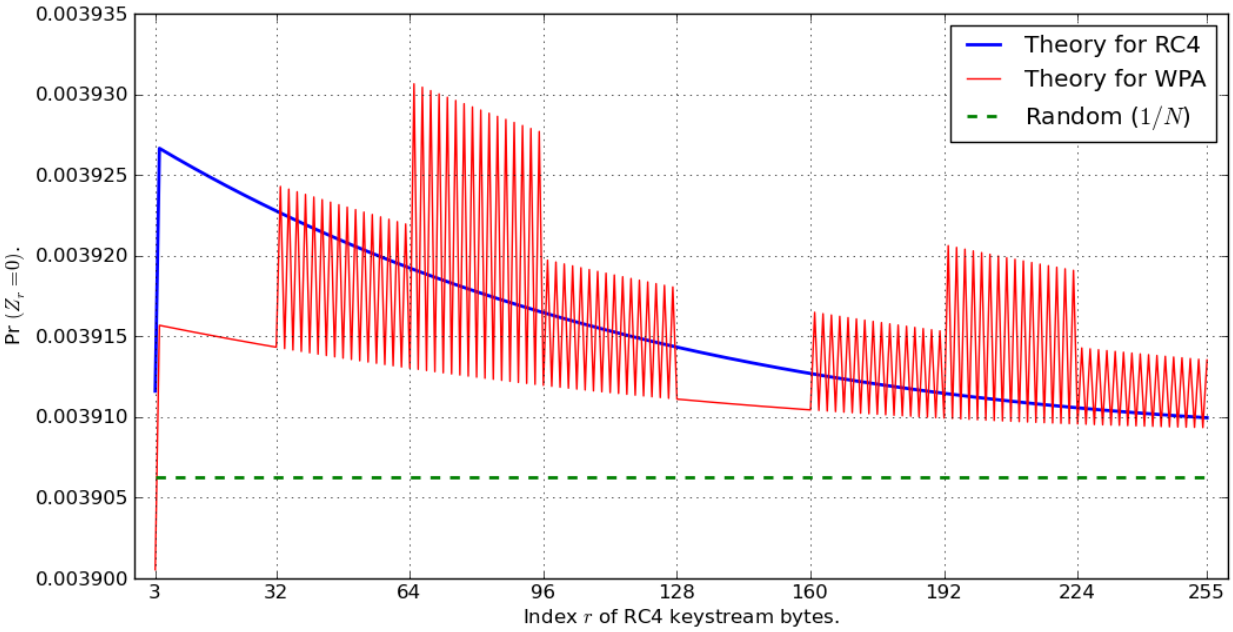

Fig. 5. Theoretical plot for $\Pr(Z_r = 0)$ for RC4 and WPA, where $r = 3, \ldots, 255$.

3 Distinguishers of WPA

Our target is to use the aforesaid biases of WPA to build a distinguisher that can efficiently distinguish WPA from generic RC4. The striking difference between WPA and RC4 have already been displayed in Figures 4 and 5, in terms of the distributions of $(Z_1 = v)$ and $(Z_r = 0)$, respectively. We may exploit either case towards the target distinguisher.

3.1 Distinguishers based on individual values of Z_1

From Fig. 4, it is evident that the probabilities $\Pr(Z_1 = v)$ for WPA and RC4 differ for almost all v. Thus, any event of type $(Z_1 = v)$, for a fixed v, will produce a distinguisher of WPA. For such a distinguisher, the complexity is estimated by [7, Theorem 2], restated as follows.

Proposition 3 (from [7]). *Let X, Y be distributions, and suppose that the event e happens in X with probability p and in Y with probability $p(1+q)$. Then for small p and q, $O(\frac{1}{pq^2})$ samples suffice to distinguish X from Y with a constant probability of success.*

We assume the distribution of $Z_1 = v$ in RC4 as our base distribution X, and the distribution of $Z_1 = v$ in WPA as the distribution Y. From our theoretical results on the distribution of Z_1 in WPA and RC4, as proved in Section 2.3, we estimate the distinguisher complexity depending on each event $(Z_1 = v)$ for $v = 0, 1, \ldots, 255$, and find the following.

Best complexity for a distinguisher based on the event $(Z_1 = v)$ that can distinguish between WPA and RC4 with more than 70% probability of success is approximately 2^{23}, applicable for $v = 34$.

In addition, we obtain the following comprehensive estimate on the distinguisher complexities:

- Complexity for a distinguisher based on $(Z_1 = v)$ is less than 2^{24} (i.e., less than N^3) for precisely 16 values of v; when $v \in \{34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64\}$.
- Complexity for a distinguisher based on $(Z_1 = v)$ is between 2^{24} and 2^{32} (i.e., between N^3 and N^4) for precisely 222 values of v.
- Complexity for a distinguisher based on $(Z_1 = v)$ is between 2^{32} and 2^{40} (i.e., between N^4 and N^5) for precisely 16 values of v; when $v \in \{148, 232, 233, 234, 235, 236, 237, 238, 241, 242, 243, 244, 245, 246, 247, 248\}$.
- Complexity for a distinguisher based on $(Z_1 = v)$ is more than 2^{40} (i.e., more than N^5) for precisely 2 values of v; when $v \in \{239, 240\}$.

3.2 Distinguishers based on sets of values of Z_1

Next, we attempt at combining the values of Z_1 in suitable subsets of $\{0, 1, \ldots, 255\}$ to construct a better distinguisher than the ones based on individual values $(Z_1 = v)$. The structure of the event considered for distinguishing WPA from RC4 in this case is "$e_S : (Z_1 \in S)$ where $S \subseteq \{0, 1, \ldots, 255\}$".

In this case however, the subset S may be quite large, and thus the probability $\Pr(e_S)$ in either distribution may not be small. In other words, the base probability p is not essentially small in this case, and thus the estimates for distinguisher complexity from [7, Theorem 2] may not work directly. To circumvent this issue, we propose the following result for estimating the complexity of a distinguisher for general p and small q.

Lemma 1. *Let X, Y be distributions, and suppose that the event e happens in X with probability p and in Y with probability $p(1 + q)$. Then for small q-value, $O(\frac{1-p}{pq^2})$ samples suffice to distinguish X from Y with a constant probability of success.*

Proof. Similar to the proof for [7, Theorem 2], with approximations on p, q reconsidered for general p. □

Now that we have a decent estimate for the distinguisher complexity, we may define a suitable set S for the target distinguishing event. As most of the 'good' (with complexity less than 2^{24}) distinguishers based on individual values of Z_1 are applicable for *even* values of the first byte, we assume that the distributions of WPA and RC4 differ the most in cases when Z_1 takes an even value. Based on this intuition, we pick the set S as the set of all even values $\{0, 2, 4, \ldots, 254\}$ within the range; thus defining the distinguishing event:

$$e_S \ : \ (Z_1 = 2k \text{ for } k = 0, 1, \ldots, 127).$$

Complexity of the distinguisher. We assume the distribution of $Z_1 \in S$ in RC4 as our base distribution X, and the distribution of $Z_1 \in S$ in WPA as the distribution Y. From our theoretical results on the distribution of Z_1 in WPA and RC4, as proved in Section 2.3, we estimate the following probabilities:

$$p = \Pr(e_S) \text{ in RC4 } \approx 0.499995, \ p(1 + q) = \Pr(e_S) \text{ in WPA } \approx 0.500713 \quad \Rightarrow \quad q \approx 0.001437 \approx 0.37/N.$$

The complexity of the distinguisher is estimated as $O(\frac{1-p}{pq^2})$, i.e., $O(N^2)$, as per Lemma 1, where the constant depends on the desired probability of success.

For $N = 256$, as in the case with practical WPA, we require an estimated $8N^2 = 2^{19}$ keystream packets to distinguish WPA from a generic instance of RC4 with more than 70% probability of success.

This is clearly the best distinguisher of WPA to date, improving the previous distinguishers of packet complexity more than 2^{40}, identified by Sepehrdad, Vaudenay and Vuagnoux [12, 13].

4 Conclusion

In this paper, we have presented an efficient algorithm that can distinguish the keystream of WPA from that of a generic instance of RC4 with a packet complexity of $O(N^2)$. In practice, our distinguisher requires approximately 2^{19} packets; thus making it the best known distinguisher of WPA to date. We clearly improve the previous WPA distinguishers identified by Sepehrdad, Vaudenay and Vuagnoux [12, 13], which require more than 2^{40} packets in practice. We have extensively experimented with our proposed distinguisher to verify its claimed packet complexity and probability of success.

The motivation of our distinguisher arises from the recent experimental observations on WPA by AlFardan, Bernstein, Paterson, Poettering and Schuldt [1], and this work puts forward an example how an experimental bias may lead to an efficient theoretical distinguisher in case of RC4-based protocols. WPA has been a time-tested protocol with wide-spread deployment in network security applications, and no simple distinguisher was ever mounted on the system. The observations on WPA biases in [1] have provided a timely opportunity to construct a simple and effective algorithm that can distinguish between WPA/TKIP and a generic instance of RC4 with a considerably low complexity.

We believe that the observations of [1] may give rise to more such results on WPA/TKIP, especially in the keylength dependent cases. For example, the single-valued distribution of the first byte $Z_1 = 129$ shows spikes in two opposite directions in WPA and generic RC4. The order of this difference in the bias is approximately $4/N^2$, which may potentially generate a WPA distinguisher of complexity $N^3/16$, i.e., approximately 2^{20} for $N = 256$. Proof for the keylength dependent bias in $Z_1 = 129$ will be interesting in this direction.

References

1. Nadhem AlFardan, Dan Bernstein, Kenny Paterson, Bertram Poettering, and Jacob Schuldt. On the security of RC4 in TLS. In *USENIX Security Symposium*, 2013. Presented at FSE 2013 as an invited talk [2] by Dan Bernstein. Full version of the research paper and relevant results are available online at `http://www.isg.rhul.ac.uk/tls/`.
2. Daniel Bernstein. Failures of secret-key cryptography. Invited talk at *FSE 2013*. Session chaired by Bart Preneel, 2013.
3. Scott R. Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the key scheduling algorithm of RC4. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 2259 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2001.
4. IEEE Computer Society. 802.11i™ – IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks –specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements, July 2004.
5. Subhamoy Maitra and Goutam Paul. New form of permutation bias and secret key leakage in keystream bytes of RC4. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 253–269. Springer, 2008.
6. Itsik Mantin. Analysis of the stream cipher RC4. Master's thesis, The Weizmann Institute of Science, Israel, 2001. Available online at `http://www.wisdom.weizmann.ac.il/~itsik/RC4/RC4.html`.
7. Itsik Mantin and Adi Shamir. A practical attack on broadcast RC4. In Mitsuru Matsui, editor, *FSE*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer, 2001.
8. Andrew Roos. A class of weak keys in the RC4 stream cipher. Two posts in sci.crypt, message-id `43u1eh$1j3@hermes.is.co.za` and `44ebge$llf@hermes.is.co.za`, 1995. Available online at `http://www.impic.org/papers/WeakKeys-report.pdf`.
9. Santanu Sarkar, Sourav Sen Gupta, Goutam Paul, and Subhamoy Maitra. Proving TLS-attack related open biases of RC4. Under submission, 2013.
10. Sourav Sen Gupta. *Analysis and Implementation of RC4 Stream Cipher*. PhD thesis, Indian Statistical Institute, 2013. Submitted on 12 July 2013.
11. Sourav Sen Gupta, Subhamoy Maitra, Goutam Paul, and Santanu Sarkar. (Non–)random sequences from (non–)random permutations – analysis of RC4 stream cipher. *Journal of Cryptology*, 2013. To appear. Published online in December 2012. DOI: 10.1007/s00145-012-9138-1.
12. Pouyan Sepehrdad. *Statistical and Algebraic Cryptanalysis of Lightweight and Ultra-Lightweight Symmetric Primitives*. PhD thesis No. 5415, École Polytechnique Fédérale de Lausanne (EPFL), 2012. Available online at `http://lasecwww.epfl.ch/~sepehrdad/Pouyan_Sepehrdad_PhD_Thesis.pdf`.

13. Pouyan Sepehrdad, Serge Vaudenay, and Martin Vuagnoux. Statistical attack on RC4 - distinguishing WPA. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 343–363. Springer, 2011.