Tornado Attack on RC4 with Applications to WEP & WPA ⋆

Pouyan Sepehrdad[1], Petr Sušil[1], Serge Vaudenay[1], and Martin Vuagnoux[2]

[1] EPFL, Lausanne, Switzerland
[2] base23 SA, Switzerland
pou.sepehrdad@gmail.com,
petr.susil@epfl.ch, serge.vaudenay@epfl.ch,
martin@vuagnoux.com

Abstract. In this paper, we construct several tools for building and manipulating pools of statistical correlations in the analysis of RC4. We develop a theory to analyze these correlations in an optimized manner. We leverage this theory to mount several attacks on IEEE 802.11 wireless communication protocols WEP and WPA. Based on several partial temporary key recovery attacks, we recover the full 128-bit temporary key of WPA by using 2^{42} packets. It works with complexity 2^{96}. Then, we describe a distinguisher for WPA with complexity 2^{42} and advantage 0.5 which uses 2^{42} packets. Moreover, we report extremely fast and optimized active and passive attacks against WEP. This was achieved through an extensive amount of theoretical and experimental analysis (capturing WiFi packets), refinement and optimization of all the former known attacks and methodologies against RC4. Our theory is supported and verified by a patch on top of Aircrack-ng. Our new attack improves its success probability drastically. Our active attack, based on ARP injection, requires 22 500 packets to gain success probability of 50% against a 104-bit WEP key, using Aircrack-ng in non-interactive mode. It runs in less than 5 seconds on an off-the-shelf PC. Using the same number of packets, Aicrack-ng yields around 3% success rate. Furthermore, we describe very fast passive only attacks by eavesdropping TCP/IPv4 packets in a WiFi communication. Our passive attack requires 27 500 packets. This is *much less than the number of packets* Aircrack-ng requires in *active mode* (around 37 500), which is a significant improvement. We believe that our analysis brings on further insight to the security of RC4.

1 Introduction

RC4 was designed by Rivest in 1987. It used to be a trade secret until it was anonymously disclosed in 1994. At present, RC4 is widely used in SSL/TLS, Microsoft Lotus, Oracle Secure SQL, Apple OCE, Microsoft Windows and Wi-Fi which is based on the IEEE 802.11 standard. IEEE 802.11 [24] used to be protected by WEP (Wired Equivalent Privacy) which is now replaced by WPA (Wi-Fi Protected Access), due to security weaknesses.

WEP uses RC4 with a pre-shared key. Each packet is encrypted by XORing it with the RC4 keystream. The RC4 key is a pre-shared key prepended with a 3-byte nonce known as the IV. This IV is sent in clear for self-synchronization. Indeed, the adversary knows that the key is constant except the IV which is known. Nowadays, WEP is considered as being terribly weak, since passive attacks can recover the full key by assuming that the first bytes of every plaintext frame is known. This happens to be the case due to the protocol specifications.

In order to fix this problem, the Wi-Fi Alliance has replaced WEP by WPA [24]. The peer authentication is based on IEEE 802.1X which accommodates a simple authentication mode based on a pre-shared key (WPA-PSK). The authentication creates a Temporary Key (TK). The TK then goes through a temporary key integrity protocol (TKIP) to derive per-packet keys (PPK). The idea is that the TK is changed into a TKIP-mixed Transmit Address and Key (TTAK) key to be used for a number of frames, limited to 2^{16}. Each frame applies a simple transformation to the TTAK and a counter TSC to derive the RC4 per-packet key PPK. Again, the 3 first bytes of the RC4 key are known (they depend on the counter). In addition to the key derivation, WPA provides a packet integrity protection scheme MIC [14]. Thus, only passive key recovery attacks can be considered.

⋆ This paper is the full version of our FSE 2013 [64] paper and the corrected version of our paper published at Eurocrypt 2011 [66].

1.1 Related Work

We recall three approaches for the cryptanalysis of RC4: attacks based on the weaknesses of the Key Scheduling Algorithm (KSA), attacks based on the weaknesses of the Pseudorandom Generator Algorithm (PRGA), and the blackbox analysis [65], which looks at RC4 as a blackbox and discovers weaknesses.

For the KSA, one of the first weaknesses published on RC4 was discovered by Roos [59] in 1995. This correlation relates the secret key bytes to the initial state of the PRGA. Maitra et al. [37] generalized Roos-type biases and introduced a related key distinguisher for RC4. Roos [59] and Wagner [81] identified classes of weak keys which reveal the secret key if the first bytes of the key are known. This property has been widely exploited to break WEP (see [6,15,21,35,34,65,4,72,79]). Another class of results concerns the inversion problem of the KSA: given the final state of the KSA, the problem is to recover the secret key [5,55].

Analysis of weaknesses in the PRGA have largely been motivated by distinguishing attacks [16,18,40,42] or initial state reconstruction from the keystream bytes [19,32,43,76] with complexity 2^{241} for the best state recovery attack. Relevant studies of the PRGA reveal biases in the keystream bytes in [41,57]. Mironov recommends in [44] that the first 512 initial keystream bytes must be discarded to avoid these weaknesses. Recently, Ohigashi et al. [52] showed that even if these initial bytes are discarded, RC4 can still be broken if used in a broadcast scheme.

In 1996, Jenkins published two biases in the PRGA of RC4 on his website [28], which were used in an attack by Klein later [31]. These biases were generalized by Mantin in his Master's Thesis [39]. In 2008, Paul, Rathi and Maitra [56] discovered a bias in the index which generates the first keystream word of RC4. Another bias in the PRGA was discovered by Maitra and Paul in [36]. Finally, Sepehrdad, Vaudenay and Vuagnoux [65] discovered 48 new correlations in the PRGA between state bytes, key bytes and the keystream and 9 new correlations between the key bytes and the keystream.

In practice, key recovery attacks on RC4 need to bind the KSA and the PRGA weaknesses to correlate secret key words to the keystream words. Some biases in the PRGA [31,56,36] have been successfully bound to the Roos correlation [59] to provide known plaintext attacks. Another approach is the blackbox analysis [65], which does not require any binding and can discover a correlation among the key bytes and the keystream directly. This was exploited in [65].

RC4 can also be used in broadcast schemes, when the same plaintext is encrypted with different keys. In this mode, the attacker often tries to find unconditional or conditional biases on the keystream (see [41,38,63,26,1,52] for the most relevant attacks.).

WEP Related Work. The WEP key recovery process is harder in practice than in theory. Indeed, some bytes of the keystream may be unknown (see the Appendix of [79] for a description of the known and unknown bytes in ARP and IP packets). Moreover, the theoretical success probabilities of these attacks have often been miscalculated and conditions to recover the secret key are not the same. For example, [72,79,4,65] check the most 10^6 probable keys instead of the first one as in [15,35,34,31,68,69]. Additionally, the IEEE 802.11 standard does not specify how the IVs should be chosen. Thus, some attacks consider randomly picked IVs and some consider incremental IVs (both little-endian and big-endian encoded). Some implementations specifically avoid some class of IVs which are weak with respect to some attacks.

To unify the results, we consider recovering a random 128-bit long secret key with random IVs. This often corresponds to the default IV behavior of the 802.11 GNU/Linux stack. We compare the previous and the new results using both a theoretical and an experimental approach.

- In [15], Fluhrer, Mantin and Shamir's (FMS) attack is only theoretically described. The authors postulated that 4 million packets would be sufficient to recover the secret key of WEP with the success probability of 50% with incremental IVs. Stubblefield, Ioannidis and Rubin [68,69] implemented this attack. They showed that between 5 million to 6 million packets are needed to recover the WEP secret key using the FMS attack. Note that in 2001, almost all wireless cards were using incremental IVs in big-endian mode.

- There is no proper theoretical analysis of the Korek [34,35] key recovery attacks. Only tools such as Aircrack-ng [11] use them, with no analysis. Aircrack-ng classifies the most probable secret keys and brute-forces them, to reach success probability of 50% with about 100 000 packets (random IVs). Note that the amount of the brute-forced keys depends on the value of the secret key and the *"Fudge"* factor (the number of trials on the key), a parameter chosen by the attacker. By default, between 1 000 to 1 000 000 keys are brute-forced. In this paper, we improve the conditions of the Korek attacks and prove their success probability.

– The ChopChop attack was introduced in [33,71]. It allows an attacker to interactively decrypt the last m bytes of an encrypted packet by sending $128 \times m$ packets in average to the network. The attack does not reveal the key and is not based on any special property of the RC4 stream cipher.

– In [31], Klein showed theoretically that his attack needs about 25 000 packets with random IVs to recover the WEP secret key with 50% success probability. Note that there is no practical implementation of the Klein attack, but both the PTW [72] and the VV07 [79] attacks, which theoretically improve the WEP key recovery process, need more than 25 000 packets. This shows that the theoretical success probability of the Klein attack was overestimated. We implemented this attack. Success probability of 50% was achieved for approximately 60 000 packets (random IVs).

– Tews, Weinmann and Pyshkin showed in [72] that the WEP secret key can be recovered with only 40 000 packets for the same success probability (random IVs). However, this attack brute-forces the most 10^6 probable secret keys. Thus, a comparison with the previous attacks is less obvious. Moreover, there is no theoretical analysis of this attack, only experimental results are provided by the authors.

– Vaudenay and Vuagnoux [79] presented an improvement to the previous attacks, where the same success probability can be reached with approximately 32 700 packets with random IVs. This attack also tests the 10^6 most probable secret keys. However, only experimental results are provided by the authors.

– According to [4], Beck and Tews re-implemented the attack in [79] in 2009, obtaining the same success probability with only 24 200 packets using Aircrack-ng in the *"interactive mode"*. Using this mode, lower number of packets is required (see Section 9 for more details on sequential distinguishers). No other previous attack used this mode, and therefore a comparison between this result and other results in the literature is not straightforward. The 10^6 most probable secret keys are brute-forced. We were not able to reproduce this result.

– In 2010, Sepehrdad, Vaudenay and Vuagnoux [65] described new key recovery attacks on RC4, which reduce the amount of packets to 9 800 packets. The most 10^6 probable keys are brute-forced as well. However, the IVs were not randomly chosen and some attacks such as the FMS were over represented.

– In 2011, Sepehrdad, Vaudenay and Vuagnoux [66] introduced an optimized key recovery attack on WEP, obtaining the same success probability as the previous attacks with only 4 000 packets, but they did not provide experimental verification of their results.

In this paper, we construct a precise theory behind our WEP attack. We show that the analysis in [66], claiming that success probability of 50% can be obtained with 4 000 packets should be re-examined. We illustrate that the variance of some random variables in [66] are not as expected and the assumption of the independence and distribution of a few random events in [66] are not correct, and thus 4 000 is not enough to break WEP in practice. All our analysis has been precisely checked through extensive experimentation. We show that we can recover a 128-bit long WEP key using 22 500 packets in less than 5 seconds using an ordinary PC. With less number of packets, a successful attack will require a longer period, because it needs to brute-force more keys.

WPA Related Work. WPA was proposed as a replacement for WEP in 2003 [24]. WPA uses a different secret key for every encrypted packet. Since 2003, a several cryptanalysis results were published against WPA, but most such attacks work only if some special features of WPA are enabled (for instance QoS), or if the same plaintext in encrypted under many different keys (may not be easily achievable). Currently, dictionary attacks [11] and recovering the PIN code of WPS [80,8] by brute-force (see below) are the main techniques that break WPA in practice. If the user chooses a safe password and WPS is disabled, we are not aware of any method that can perform a key recovery attack on WPA in a short period of time. Below, we list the most well-known attacks on WPA in the literature:

– *Dictionary Attack*: Through eavesdropping the network, the goal of the attacker is to get a WPA handshake [25,11]; the hash of the key is communicated between the client and the Access Point (AP) when the client begins the connection. The attacker can wait or launch a deauthenticate-attack against the client. When he gets the hash, he can try to find the key with a dictionary attack, a rainbow attack [51] or one of the multiple attacks that exist on hashed keys.

– A flaw in WiFi Protected Setup (WPS) is known from the end of 2011 by Tactical Network Solutions (TNS) [80]. From this exploit, the WPA password can be recovered in 2-10 hours. This attack only works if WPS (PIN method) is supported and enabled by the AP. Another recent attack by Bongard [8] exploits weak randomization, or the lack of randomization, in a key used to authenticate hardware PINs on some implementations of WPS, allowing anyone to quickly collect enough information to guess the PIN using offline calculations. By calculating the correct PIN, rather than attempting to brute-force the numerical password, the new attack circumvents defences instituted by companies. While the previous attack requires up to 11 000 guesses to find the correct PIN to access the router's WPS functionality, the new attack only requires a single guess and a series of offline calculations, which take a few seconds to finish.

– In 2009, Beck and Tews released an attack on WPA [4]. This is not a key recovery attack, but still exploits weaknesses in TKIP to allow the attacker to decrypt ARP packets and to inject traffic into a network, even allowing her to perform a DoS (Denial of Service) attack or an ARP poisoning. In order to be practical, the attack requires some additional quality of services features (described by IEEE 802.11e) to be enabled.

– The Ohigashi-Morii Attack [53] is an improvement of the Beck-Tews attack on WPA-TKIP. Indeed, this attack is efficient for all modes of WPA and not just those with QoS features. The time to inject a fake packet is reduced to approximately 15 minutes to 1 minute at the best. For this attack, a man-in-the-middle attack is superposed to the Beck-Tews attack, to reduce the execution time of the attack. In [75], the time complexity of Ohigashi-Morii attack was improved. This new attack focuses on a new vulnerability of QoS packet processing. This attack still works even if the Access Point (AP) does not support IEEE 802.11e.

– The Hole196 vulnerability was found by Airtight Networks [48] in 2010. The name "Hole196" refers to the page number in the IEEE 802.11 Standard (Revision, 2007) where the vulnerability is buried. This attack is not a key recovery attack. The attacker has to be an authorized user of the network. All Wi-Fi networks using WPA or WPA2, regardless of the authentication (PSK or 802.1x) and encryption (AES) they use, are vulnerable.

– An attack against the Michael message integrity code of WPA was presented in [3], that allows an attacker to reset the internal Message Integrity Check (MIC) state. It concatenates a known message with an unknown message which keeps the unknown MIC valid for a new packet.

– In 2004, Moen, Raddum and Hole [45] discovered that the recovery of at least two RC4 packet keys in WPA leads to a full recovery of the temporal key and the message integrity check key. Once from the same segment of 2^{16} consecutive packets two RC4 keys are successfully recovered, the Moen, Raddum and Hole attack can be applied. This leads to a TK key recovery attack on WPA with complexity 2^{104} using 2 packets.

– Paterson, et al. [54] observed very large, IV dependant biases in the RC4 keystream when the algorithm is keyed according to the WPA specification. They leveraged these biases together with similar techniques presented in [1,26] (used to attack RC4 in TLS), to mount a statistical plaintext recovery attack on WPA, in the situation where the same plaintext is encrypted in many different frames, i.e., RC4 in "broadcast attack" setting. They were able to recover the first 256 bytes of a frame, using $2^{24} - 2^{30}$ encrypted frames, depending on the success probability. This attack does not recover the TK of WPA. Later, in [61], Sen Gupta, et al. revisited the correlation of initial keystream bytes in WPA to the first three bytes of the RC4 key, which are known from the IV. Using these correlations, they improved the data complexity of the attack in [54] for few keystream bytes.

– In [27], Ito, et al. focused on the state information and investigated various linear correlations among the unknown state information, the first three bytes of the RC4 key, and the keystream bytes in both generic RC4 and WPA. Particularly, those linear correlations are effective for the state recovery attack since they include the first known three-byte keys (IV-related) information.

– Recently, Vanhoef, et al. [78] introduced another attack on WPA. Their attack works on RC4 in broadcast scheme model, i.e., for the attack to work, the same packet needs to be encrypted with different keys. To satisfy this requirement, they introduced a method to generate a large number of identical packets: If the IP of the victim is known and incoming connections towards it is not blocked, they can simply send identical packets to the victim. Otherwise, they induce the victim into opening a TCP connection to an attacker-controlled server. This connection is then used to transmit identical packets to the victim. Next, they use a large number of correlations in RC4 keystream to decrypt some packets and derive the TKIP MIC value. Given the plaintext data and its MIC value, they could efficiently derive the MIC key [77]. It is then explained how the MIC key can be used to inject and decrypt packets. In practice, the attack can be executed within an hour. This attack does not recover the WPA temporary key (TK).

We extend Moen, Raddum and Hole attack. We first recover several weak bytes of the key and then we apply Moen, Raddum and Hole attack. As a result, we propose a key recovery attack against WPA with complexity 2^{96} using 2^{42} packets.

1.2 Our Overall Contribution

In this paper, we construct tools and a theory for building and manipulating a pool of statistical correlations in RC4. With our theory, we analyze several statistical strategies for a partial key recovery on WEP and WPA. We apply them to recover some weak bits of the WPA key TK by using 2^{42} packets. We then build a full session key recovery attack against WPA with complexity 2^{96} and using 2^{42} packets. Later, we transform our partial key recovery attack into a distinguisher for WPA. Our distinguisher was further improved by [60] using another technique. We apply our analysis to WEP and show experimentally that the best attacks so far can still be improved. We review some errors in our previous publications [65,66] and verify our results through experiments.

Structure of the Paper. We first present RC4, WEP, WPA and Aircrack-ng in Section 2. Next, the general principle of the attacks on WEP and WPA is described in Section 3. We then introduce some useful definitions and lemmas in Section 4. Some weaknesses on RC4 are described in the form of lemmas in Section 5. As an example and for more clarification, two significant statistical biases in RC4 are elaborated in Section 6 for the target key bytes. Then, we study key recovery attacks to be able to recover some “weak bits” of the temporary key of WPA in Section 7. Then, we present a full temporary key recovery attack for WPA in Section 7.4. We also introduce a distinguisher for WPA in Section 7.5. We present an optimized attack on WEP in Section 8 and then we compare our experimental results with Aircrack-ng 1.1 in Section 9 and finally we conclude in Section 10.

2 Description of the Algorithms and Protocols

2.1 Description of RC4 and Notations

RC4 consists of two algorithms: the Key Scheduling Algorithm (KSA) and the Pseudo Random Generator Algorithm (PRGA). RC4 has a state defined by two registers (words) i and j and an array (of N words) S defining a permutation over $\mathbf{Z}_N$. RC4 KSA generates an initial state for the PRGA from a random key K of L words as described in Fig. 1.

Note that, we define all the operators such as addition, and multiplication in the ring of integers modulo N represented as $\mathbf{Z}/N\mathbf{Z}$, or $\mathbf{Z}_N$, where $N = 256$ (i.e. *words* are *bytes*). Thus, $x+y$ should be read as $(x+y) \bmod N$.

Throughout this paper, we denote $\bar{K}[i] := K[0] + \cdots + K[i]$. Note that, we recover $\bar{K}[i]$'s, instead of $K[i]$'s, because this approach increases the success probability of key recovery (see [79] for more details). The variable z denote the keystream derived from the key K using RC4. The first bytes of a plaintext frame are often known (see [79]), as well as the IV (the first 3 bytes of the key K). That is, we assume that the adversary can use z and the IV in a known plaintext attack.

KSA starts with an array $\{0,1,\ldots,N-1\}$, where $N = 2^8$ and swaps N pairs, depending on the value of the secret key K. At the end, we obtain the initial state for the PRGA.

KSA

```
for i = 0 to N − 1 do
    S[i] ← i
end for
j ← 0
for i = 0 to N − 1 do
    j ← j + S[i] + K[i mod L]
    swap(S[i],S[j])
end for
```

PRGA

```
i ← 0
j ← 0
loop
    i ← i + 1
    j ← j + S[i]
    swap(S[i],S[j])
    output z_i = S[S[i] + S[j]]
end loop
```

Fig. 1. The KSA and the PRGA Algorithms of RC4

The PRGA's role is to generate a keystream of words of $\log_2 N$ bits, which will be XORed with the plaintext to obtain the ciphertext. Thus, RC4 computes the loop of the PRGA each time a new keystream word z_i is needed, according to the algorithm in Fig. 1. Note that each time a word of the keystream is generated, the internal state of RC4 is updated.

Sometimes, we consider an idealized version $\mathrm{RC4}^\star(t)$ of RC4 defined by a parameter t as shown in Fig. 2. Namely, after the round t, j is assigned randomly. This model has already been used in the literature, such as in [42,59,55].

$\mathsf{KSA}^\star(t)$

```
for i = 0 to N − 1 do
    S[i] ← i
end for
j ← 0
for i = 0 to N − 1 do
    if i ≤ t then
        j ← j + S[i] + K[i mod L]
    else
        j ← random
    end if
    swap(S[i],S[j])
end for
```

$\mathsf{PRGA}^\star$

```
i ← 0
j ← 0
loop
    i ← i + 1
    j ← random
    swap(S[i],S[j])
    output z_i = S[S[i] + S[j]]
end loop
```

Fig. 2. The $\mathsf{KSA}^\star(t)$ and the $\mathsf{PRGA}^\star$ algorithms of $\mathrm{RC4}^\star(t)$

Let $S_i[k]$ (resp. $S'_i[k]$) denote the value of the permutation defined by array S at index k, after the round i of the KSA (resp. the PRGA), where $S_{-1} := \{0,1,\ldots,N-1\}$. We also denote $S_{N-1} = S'_0$. Let j_i (resp. j'_i) be the value of j after the round i of the KSA (resp. PRGA) where the rounds are indexed with respect to i. Thus, the KSA has rounds

$0,1,\ldots,N-1$ and the PRGA has rounds $1,2,\ldots$. RC4 KSA and PRGA are defined by

$$\begin{array}{rl|rl}
\multicolumn{2}{c|}{\mathsf{KSA}} & \multicolumn{2}{c}{\mathsf{PRGA}} \\
j_{-1} &= 0 & j'_0 &= 0 \\
j_i &= j_{i-1}+S_{i-1}[i]+K[i \bmod L] & j'_i &= j'_{i-1}+S'_{i-1}[i] \\
S_{-1}[k] &= k & S'_0[k] &= S_{N-1}[k] \\
S_i[k] &= \begin{cases} S_{i-1}[j_i] & \text{if } k=i \\ S_{i-1}[i] & \text{if } k=j_i \\ S_{i-1}[k] & \text{otherwise} \end{cases} & S'_i[k] &= \begin{cases} S'_{i-1}[j'_i] & \text{if } k=i \\ S'_{i-1}[i] & \text{if } k=j'_i \\ S'_{i-1}[k] & \text{otherwise} \end{cases} \\
 & & z_i &= S'_i[S'_i[i]+S'_i[j'_i]]
\end{array}$$

In WEP and WPA attacks, the basis of the complexity measurement is the time it takes to compute the key value which is determined by the biased equation.

2.2 Description of WEP

WEP [22] uses a 3-byte IV concatenated to a secret key of 40 or 104 bits (5 or 13 bytes) as an RC4 key. Thus, the RC4 key size is either 64 or 128 bits. In this paper, we do not consider the 40-bit key variant. So, $L=16$. We have

$$K = K[0]\|K[1]\|K[2]\|K[3]\|\cdots\|K[15] = \mathsf{IV}_0\|\mathsf{IV}_1\|\mathsf{IV}_2\|K[3]\|\cdots\|K[15]$$

where IV_i represents the $(i+1)$-th byte of the IV and $K[3]\|...\|K[15]$ represents the fixed 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 is incremented by one each time a new 802.11b frame is encrypted. Sometimes, some particular values of the IV are skipped to thwart specific attacks based on the “weak IVs”. Thus, each packet uses a slightly different key.

To protect the integrity of the data, a 32-bit long CRC32 checksum called ICV is appended to the data. Similar to other stream ciphers, the resulting stream is XORed with the RC4 keystream and it is sent through the communication channel together with the IV in clear. On the receiver’s end, the ciphertext is again XORed with the shared key and the plaintext is recovered. The receiver checks the linear error correcting code and it either accepts the data or declines it.

It is well known [58,72,79] that a some portion of the plaintext is practically constant and that some other bytes can be predicted. 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. Thus, even if these attacks are called known plaintext attacks, they are ciphertext only in practice (see the Appendix of [79] for the structure of ARP and TCP/IPv4 packets).

2.3 Description of WPA

WPA includes a key hashing function [20] to defend against the Fluhrer, Mantin and Shamir attack [15], a Message Integrity Code (MIC) [14] and a key management scheme based on 802.1X [23] to avoid the key reuse and to ease the key distribution.

The 128-bit Temporal Key (TK) is a per-session key. It is derived from the key management scheme during authentication and is given as an input to the phase1 key hashing function (key mixing algorithm), together with a 48-bit Transmitter Address (TA) and a 48-bit TKIP Sequence Counter (TSC) which is sometimes called the IV. We will avoid this latter name to avoid any confusion with the first 3 bytes of the RC4 key (which indeed only depends on the TSC, but with a shorter length).

The TK can be used to encrypt up to 2^{48} packets. Every packet has a 48-bit index TSC which is split into IV32 and IV16. The IV32 counter is incremented every 2^{16} packets. The packet is encrypted using a 128-bit RC4KEY which is derived from the TK, TSC, and some other parameters (e.g. device addresses) which can be assumed as constants and known by the adversary for our purpose. Similar to WEP, the first three bytes of the RC4KEY only depend on the TSC, so they are not secret. The derivation works in two phases according to the standard [20]. The first phase does not depend on IV16 and is done once every 2^{16} packets for efficiency reasons. It derives a 80-bit key TTAK, called

TKIP-mixed Transmit Address and Key (TTAK) in the standard (but, is denoted P1K in the reference code). This is performed in a 2 step process: PHASE1_STEP1 and PHASE1_STEP2 (see [20]).

$$\mathsf{TTAK} = \mathsf{phase1}(\mathsf{TK}, \mathsf{TA}, \mathsf{IV32})$$

The second phase uses the TTAK, TK and the IV16 to derive a 96-bit key PPK which is then turned into the RC4KEY. This is performed in a 3 step process: PHASE2_STEP1, PHASE2_STEP2, and PHASE2_STEP3 (see [20]).

$$\mathsf{RC4KEY} = \mathsf{phase2}(\mathsf{TK}, \mathsf{TTAK}, \mathsf{IV16})$$

The key derivation of WPA based on a pre-shared key is depicted in Fig. 3 (without protocol parameters such as the transmitter address TA).

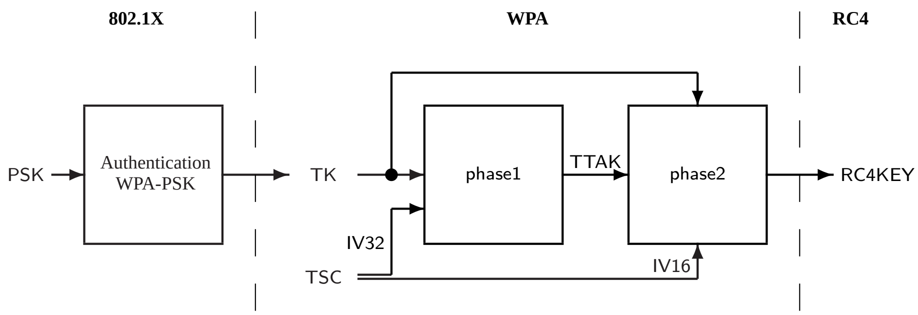

Fig. 3. The WPA Key Derivation based on the Pre-Shared Key Authentication Method

In what follows, we denote $K[i] = \mathsf{RC4KEY}[i \bmod 16]$ and $\mathsf{IV} = K[0]\|K[1]\|K[2]$ to use the same notations as in WEP. By convention, the TTAK and the PPK are considered as vectors of 16-bit words. The TK and the RC4KEY are considered as vectors of 8-bit words. Vectors are numbered starting from 0.

The RC4KEY is simply defined (PHASE2_STEP3 in [20]) from the PPK, TK and the IV16 by

$$\begin{array}{llll}
\mathsf{RC4KEY}[0] &= \mathsf{high8}(\mathsf{IV16}) & \mathsf{RC4KEY}[1] &= (\mathsf{high8}(\mathsf{IV16}) \text{ or } 0\text{x}20) \text{ and } 0\text{x}7\text{f} \\
\mathsf{RC4KEY}[2] &= \mathsf{low8}(\mathsf{IV16}) & \mathsf{RC4KEY}[3] &= \mathsf{low8}((\mathsf{PPK}[5] \oplus (\mathsf{TK}[1]\|\mathsf{TK}[0])) \gg 1) \\
\mathsf{RC4KEY}[4] &= \mathsf{low8}(\mathsf{PPK}[0]) & \mathsf{RC4KEY}[5] &= \mathsf{high8}(\mathsf{PPK}[0]) \\
\mathsf{RC4KEY}[6] &= \mathsf{low8}(\mathsf{PPK}[1]) & \mathsf{RC4KEY}[7] &= \mathsf{high8}(\mathsf{PPK}[1]) \\
\mathsf{RC4KEY}[8] &= \mathsf{low8}(\mathsf{PPK}[2]) & \mathsf{RC4KEY}[9] &= \mathsf{high8}(\mathsf{PPK}[2]) \\
\mathsf{RC4KEY}[10] &= \mathsf{low8}(\mathsf{PPK}[3]) & \mathsf{RC4KEY}[11] &= \mathsf{high8}(\mathsf{PPK}[3]) \\
\mathsf{RC4KEY}[12] &= \mathsf{low8}(\mathsf{PPK}[4]) & \mathsf{RC4KEY}[13] &= \mathsf{high8}(\mathsf{PPK}[4]) \\
\mathsf{RC4KEY}[14] &= \mathsf{low8}(\mathsf{PPK}[5]) & \mathsf{RC4KEY}[15] &= \mathsf{high8}(\mathsf{PPK}[5])
\end{array}$$

Note that a filter avoids the use of some weak IV classes. Actually, only the weak IV class discovered by Fluhrer, Mantin, and Shamir [15] are filtered.

2.4 Aircrack-ng

Aircrack-ng [11] is a program for WEP and WPA-PSK keys cracking. It can recover the keys once enough packets have been captured. It is the most widely downloaded cracking software in the world. It implements the standard Fluhrer, Mantin and Shamir's (FMS) attack [15] along with some optimisations such as Korek attacks [34,35], as well as the Physkin, Tews and Weinmann (PTW) attack [72]. We applied a patch to Aircrack-ng 1.1 to improve its success probability.

3 General Principle of the Attacks

Below we present the high-level description of our attacks on WEP and WPA. All these attacks are known-plaintext. However, in practice they are ciphertext only, because some messages (plaintext bytes) are known due to the IEEE 802.11 standard specifications.

It is known for years that many statistical events happening between the RC4 key bytes, the state bytes, the keystream and the IV are not distributed uniformly at random. These events are biased with some specific probability. We leverage these biased relations in our attacks.

Below we give an example of one of these biases (the Korek $\mathsf{A\_u13\_2}$ bias):

$$\bar{K}[3] = 1 - \sigma_3(2) \text{ if } S_2[3] = 2,\ S_2[1] = 0 \text{ and } z_1 = 3$$

This event happens with probability $P_u^3(3,2)$, where

$$\sigma_3(2) = S_0[1] + S_1[2]$$
$$P_u^3(3,2) = \left(\tfrac{N-1}{N}\right)^3 \left(\tfrac{N-2}{N}\right)^{N-4} + \tfrac{1}{N}\left(1 - \left(\tfrac{N-2}{N}\right)^{N-4}\right) \approx 35.9/N$$

Since $K[0], K[1]$ are known, $\sigma_3(2)$ can be computed by an attacker. We denote $f := 2 - \sigma_3(2)$ as the biased equation or the biased relation, and the event $g := (S_2[3] = 0$ and $z_2 = 0)$ as the condition of this bias. We later prove how these conditions lead to the above biased relation by describing the attack path. The attack path is the path which needs to be followed through the KSA and the PRGA for the biased equation to hold. In the example above, the probability that the attack path occurs is $35.9/N$.

To recover $\bar{K}[i]$, given a set of key bytes we already know and an index i, we assume that we have a list of d_i equations represented as $\bar{K}[i] = f_j$ together with its conditions denoted as g_j, where j is non-negative and $j < d_i$, such that

$$\mathsf{Pr}\left[\bar{K}[i] = f_j(z, \mathsf{clue}) | g_j(z, \mathsf{clue})\right] = p_j$$

for some probability $p_j \neq \frac{1}{N}$ and

$$\mathsf{Pr}\left[g_j(z, \mathsf{clue})\right] = q_j$$

We refer to f_j as the biased *equation* or *relation*, to g_j as the bias *condition* and to q_j as the bias *density*. clue represents some known bytes, such as state, or key bytes. We use the list of biases from Table 3. The mysterious function $\sigma_i(t)$ in Table 3 can be computed using the clue. The exact definition of this function is given in Lemma 8 later.

For simplicity, we assume that for some given i, z and clue, all suggested $f_j(z, \mathsf{clue})$ for j's such that $g_j(z, \mathsf{clue})$ holds are pairwise distinct. We further assume that the events $\bar{K}[i] = f_j(z, \mathsf{clue})$ with different i's are independent.

We classify RC4 biases into two categories: the conditional biases and the unconditional biases. We use these notions specifically in the WPA attack in Section 7. Although all the biases are conditional, i.e., there always exists a g, the unconditional category includes the biases whose condition density is very close to one. Any bias which is not an unconditional bias is a conditional bias. we place the $\mathsf{SVV\_10}$ and the Korek biases in the conditional category and the Klein-Improved bias in the unconditional category (see Appendix C).

3.1 Attack on WEP

We discuss several biases in the key bytes, the keystream, IV and the state bytes of RC4. We leverage these biases to vote for individual key bytes of RC4. In fact, we vote for $\bar{K}[i]$'s instead of $K[i]$'s. For the WEP attack, we first recover the value of $\bar{K}[15]$. This is done because we have a fundamental relation in RC4, which is as follows:

$$\bar{K}[i + 16j] = \bar{K}[i] + j\bar{K}[15] \tag{1}$$

for $0 \le i \le 15$ and $j = 0,1$ and 2. This means that if the value of $\bar{K}[15]$ is known, the biases for $\bar{K}[i+16j]$ can be used to vote for $\bar{K}[i]$. This helps us increase the probability of recovering $\bar{K}[i]$ correctly.

Next, we use the biases to vote for $\bar{K}[3]$ to $\bar{K}[14]$ sequentially. We do this sequentially, because if the value of $\bar{K}[3]$ is known, since $K[0]$, $K[1]$, $K[2]$ are also known (they make the IV), we can update the state to S_3. This will increase the success probability of recovering $\bar{K}[4]$. Hence, we first recover $\bar{K}[3]$ and then we update the state to S_3, then we recover $\bar{K}[4]$ and update the state to S_4, and we continue this process until we recover $\bar{K}[14]$.

The open problem we are trying to address in this paper is to derive an optimized method for voting, which leads to the highest success probability. To reach this goal, we need to compute the probability of every individual biased relation, together with devising a method to combine them together, in order to break WEP with the least number of packets. Finally, we need to compute the correct parameters to reach the least number of packets to break WEP.

3.2 Attack on WPA

Moen, et al. [45] discovered that the recovery of at least two RC4 packet keys in WPA leads to a full recovery of the temporal key and the message integrity check key. The results from [45] lead to an "easy" attack on WPA. According to the description of how an RC4KEY is derived (last paragraph of Section 2.3), to recover two RC4KEYs, we can just guess the 96-bit PPK and the 8 weak bits of the TK with an average complexity of 2^{103} until it generates the correct keystream. Then, we guess the 96-bit PPK of another packet in the same segment. Hence, the average complexity of the full attack is 2^{104}

In this paper, we improve this attack by recovering the weak bits of the TK separately: after having recovered the weak bits, we note that the 96-bit PPK is now enough to recalculate the RC4KEY. So, we can do an exhaustive search on the PPK for a given packet until we find the correct one. This works with average complexity of 2^{95}. We do not need to recover all key bytes to be able to discover the 8 weak temporary key bytes of WPA. It will be shown in Section 7, that we only need to recover $\bar{K}[15]$, $\bar{K}[3]$, $\bar{K}[13]$ and $\bar{K}[14]$. We use a similar technique as the WEP attack to recover these key bytes, but in this case, we only use the state S_2 for key recovery. Since, we need to recover two RC4KEYs, the average complexity of the attack is 2^{96}.

We are still trying to address the same open problems as described for WEP. However, the theory behind how to merge the biases for WPA, and how to set the parameters in an optimized manner get much more complicated in WPA compared to WEP.

4 Some Useful Definitions and Lemmas

In this section, we present some mathematical definitions and lemmas which are useful later in computing the success probability of our attacks on WEP and WPA.

Definition 1. *We denote*

$$\varphi(\lambda) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\lambda} e^{-\frac{x^2}{2}}\, dx = \frac{1}{2}\mathsf{erfc}\left(-\frac{\lambda}{\sqrt{2}}\right)$$

In particular, $\varphi(-\lambda/\sqrt{2}) = \frac{1}{2}\mathsf{erfc}(\frac{\lambda}{2})$.

Definition 2. *The gamma function over the field of complex numbers is an extension of the factorial function and is defined as:*

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

for $\mathsf{Re}(x) > 0$.

– The beta function, also called the Euler integral of the first kind, over the field of complex numbers is defined as:

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1} dt$$

for $\mathsf{Re}(a) > 0$ *and* $\mathsf{Re}(b) > 0$.

– *The incomplete beta function is a generalization of the beta function and is defined as:*

$$B(x;a,b) = \int_0^x t^{a-1}(1-t)^{b-1}dt$$

– *The regularized incomplete beta function is defined in terms of the incomplete beta function and the complete beta function as*

$$I_x(a,b) = \frac{B(x;a,b)}{B(a,b)}$$

– *We say that* X *has a negative binomial distribution if it has a probability mass function:*

$$\mathsf{Pr}[X = x] = \binom{x+r-1}{x}(1-p)^r p^x$$

where r *is a positive integer and* p *is real.* r *and* p *are both parameters of this distribution. Extending this definition by letting* r *to be real positive, the binomial coefficient can also be rewritten using the gamma function:*

$$\mathsf{Pr}[X = x] = \frac{\Gamma(x+r)}{x!\Gamma(r)}(1-p)^r p^x$$

This generalized distribution is called the Pólya distribution. We also have

$$E(X) = \frac{pr}{(1-p)} \qquad \textit{and} \qquad V(X) = \frac{pr}{(1-p)^2}$$

The cdf *of this distribution can be computed using the regularized incomplete beta function. In fact, we have*

$$F_X(x) = \mathsf{Pr}(X \le x) = 1 - I_p(x+1, r)$$

Definition 3. *Let* A, B *and* C *be three random variables over* $\mathbf{Z}_N$. *We say that* A *is biased towards* B *with bias* p *conditioned on an event* E *and we represent it as* $A \underset{E}{\overset{p}{=}} B$ *if*

$$\mathsf{Pr}(A - B = x|E) = \begin{cases} p & \textit{if} \quad x = 0 \\ \frac{1-p}{N-1} & \quad \textit{otherwise} \end{cases}$$

When $\mathsf{Pr}[E] = 1$, *it is denoted as* $A \overset{p}{=} B$.

Lemma 4. *Let* A, B *and* C *be random variables in* $\mathbf{Z}_N$ *such that*

$$A \overset{p_1}{=} B \qquad B \overset{p_2}{=} C$$

We assume that $A - B$ *and* $B - C$ *are independent. We have* $A \overset{P}{=} C$, *where*

$$P = \frac{1}{N} + \left(\frac{N}{N-1}\right)\left(p_1 - \frac{1}{N}\right)\left(p_2 - \frac{1}{N}\right) \overset{\mathsf{def}}{=} p_1 \otimes p_2$$

The operator $\otimes$ *is commutative and associative over* $[0,1]$, *where* 1 *is the neutral element.*

The proof of the above lemma is provided in Appendix A.1. From the above lemma and the associativity of $\otimes$, we deduce the corollary below:

Corollary 5. *Let A, B, C, D and E be random variables in $\mathbf{Z}_N$ such that*

$$A \stackrel{p_1}{=} B \qquad B \stackrel{p_2}{=} C \qquad C \stackrel{p_3}{=} D \qquad D \stackrel{p_4}{=} E$$

We assume that $A - B$, $B - C$, $C - D$ and $D - E$ are independent. We have $A \stackrel{P}{=} E$, where

$$P = p_1 \otimes p_2 \otimes p_3 \otimes p_4 = \frac{1}{N} + \left(\frac{N}{N-1}\right)^3 \cdot \prod_{i=1}^{4} \left(p_i - \frac{1}{N}\right)$$

For $p_4 = 1$, we obtain

$$P = p_1 \otimes p_2 \otimes p_3 = \frac{1}{N} + \left(\frac{N}{N-1}\right)^2 \cdot \prod_{i=1}^{3} \left(p_i - \frac{1}{N}\right)$$

We can extend the above corollary by adding new conditions.

Lemma 6. *Let A, B, C, D and E be random variables in $\mathbf{Z}_N$ and Cond and Cond' be two events such that*

$$A \stackrel{p_1}{=} B \qquad B \stackrel{p_2}{=} C \qquad C \underset{\mathsf{Cond}'}{\stackrel{p_3}{=}} S[D] \qquad D \stackrel{p_4}{=} E$$

Let for all, α, β, γ and δ, the events $A - B = \alpha$, $B - C = \beta$, $(C - S[D] = \gamma) \wedge \mathsf{Cond}'$ and $D - E = \delta$ be independent; furthermore, let

1. $((A = S[D]) \wedge \mathsf{Cond}) \Leftrightarrow ((A = S[D]) \wedge \mathsf{Cond}')$
2. $\Pr[\mathsf{Cond}] = \Pr[\mathsf{Cond}']$ *and* $\Pr[D = E | \mathsf{Cond}] = \Pr[D = E | \mathsf{Cond}']$
3. $\Pr[A = S[E] | A \neq S[D], D \neq E, \mathsf{Cond}] = \frac{1}{N-1}$

We have

$$\Pr[A = S[E] | \mathsf{Cond}] = p_1 \otimes p_2 \otimes p_3 \otimes p_4$$

The proof of this lemma is provided in Appendix A.2. We use the lemma above in Section 6 and also in analyzing the rest of the biases in Appendix C. Later, we make a heuristic assumption that the events 1, 2 and 3 occur.

5 Some Weaknesses in RC4

In this section, we introduce some more lemmas which specifically represent a weakness in RC4. They are very useful in the next sections.

The next lemma represents a relation between $\bar{K}[i]$ and the value of j_i.

Lemma 7. *In the KSA of RC4, we have*

$$\bar{K}[i] = j_i - \sum_{x=1}^{i} S_{x-1}[x]$$

Proof. We prove it by induction by using

$$j_i = j_{i-1} + S_{i-1}[i] + K[i]$$

□

The following lemma describes the probability that some state bytes remain at their position during RC4 state updates. Intuitively, S_t is the last state the attacker can recover. For instance, for WEP and WPA, since the IV is known, the attacker can initially compute up to state S_2, therefore $t = 2$ in this case. Later, when he recovers more key bytes sequentially, t will increase. Hence, from now on, any time we talk about the index t, we mean the index of the last state which the attacker is able to compute.

Lemma 8. *For any $0 < i < N$, and any $-2 < t < i$, the following five relations hold on RC4$^\star(t)$ for any set $(m_1, \ldots, m_b)$ of distinct m_j's such that $m_j \le t$ or $m_j > i-1$:*

$$P_A^b(i,t) \stackrel{\mathsf{def}}{=} \mathsf{Pr}\left[\bigwedge_{j=1}^{b} (S_{i-1}[m_j] = \cdots = S_{t+1}[m_j] = S_t[m_j])\right] = \left(\tfrac{N-b}{N}\right)^{i-t-1}$$

$$S_{i-1}[m_j] \stackrel{P_A^1}{=} S_t[m_j]$$

$$\sum_{x=1}^{i} S_{x-1}[x] \stackrel{P_B(i,t)}{=} \sigma_i(t) \quad \textit{with} \quad P_B(i,t) \stackrel{\mathsf{def}}{=} \prod_{k=0}^{i-t-1}\left(\frac{N-k}{N}\right) + \frac{1}{N}\left(1 - \prod_{k=0}^{i-t-1}\left(\frac{N-k}{N}\right)\right)$$

$$P_0 \stackrel{\mathsf{def}}{=} \mathsf{Pr}[S'_{i-1}[i] = \cdots = S'_1[i] = S_{N-1}[i] = \cdots = S_i[i]] = \left(\tfrac{N-1}{N}\right)^{N-2}$$

$$S'_{i-1}[i] \stackrel{P_0}{=} S_i[i]$$

where

$$\sigma_i(t) = \sum_{j=0}^{t} S_{j-1}[j] + \sum_{j=t+1}^{i} S_t[j]$$

The proof of this lemma is provided in Appendix A.3.

We use the Jenkins' correlation to construct the Klein-Improved attack (described in the next section). We introduce and prove this correlation below:

Lemma 9. ***(Jenkins' correlation [28]).*** *Assuming the internal state S_{N-1} is a random permutation, and j'_i is chosen randomly, then $z_i + S'_i[j'_i] \stackrel{P_J}{=} i$, where $P_J = \frac{2}{N}$.*

Proof.

$$\begin{aligned}\mathsf{Pr}[S'_i[j'_i] = i - z_i] &= \mathsf{Pr}[S'_i[j'_i] = i - z_i | S'_i[i] + S'_i[j'_i] = i] \;.\; \mathsf{Pr}[S'_i[i] + S'_i[j'_i] = i] \\ &\quad + \mathsf{Pr}[S'_i[j'_i] = i - z_i | S'_i[i] + S'_i[j'_i] \neq i] \;.\; \mathsf{Pr}[S'_i[i] + S'_i[j'_i] \neq i] \\ &= \tfrac{1}{N} + \tfrac{1}{N}\left(1 - \tfrac{1}{N}\right) \approx \tfrac{2}{N}\end{aligned}$$

□

The following lemma by Mantin is the most spectacular correlation ever found on RC4. Thanks to this lemma, the keystream of RC4 can be distinguished from random with only N packets.

Lemma 10. ***(Theorem** 1 **in [41])*** *Assume that the initial permutation $S'_0 = S_{N-1}$ is randomly chosen from the set of all the possible permutations over $\{0, \ldots, N-1\}$. Then, the probability that the second output word of RC4 is 0 is approximately $\frac{2}{N}$. In fact, we have $z_2 \stackrel{\frac{2}{N}}{=} 0$.*

The proof of this lemma is provided in Appendix A.4.

6 Two Significant Biases in RC4

In this section, we describe two significant biases in RC4 as an example of how we use such correlations in a successful attack, namely: the Klein-Improved bias (an unconditional bias) and the A_u15 bias (a conditional bias). The complete list of all such biases are presented in Appendix C. To get an intuition of the numerical values of the densities and probabilities of all the biases we use, we present them in Appendix B for some fixed values of i and t, where P, and g are the probability and the density of the biases respectively. For simplicity, we use the word Cond and the event $g(z, \mathsf{clue})$ (described in Section 4) interchangeably in this section. As we mentioned earlier, S_t represents the last state which is computable by the attacker. For instance, $K[0]$, $K[1]$ and $K[2]$ are initially known, therefore, the state up to S_2 can be computed. In the WEP attack, we recover $\bar{K}[3]$ first using S_2 and then using $\bar{K}[3]$ we update the state to S_3 and recover $\bar{K}[4]$. We continue this process until we recover $\bar{K}[14]$. On the other hand, for WPA, we set $t = 2$ all the time, and we only use S_2, to recover $\bar{K}[15]$, $\bar{K}[3]$, $\bar{K}[13]$ and $\bar{K}[14]$.

6.1 The Klein-Improved Attack

Klein [31] combined Jenkins' correlation for the PRGA and the weaknesses in the KSA to derive a correlation between the RC4 key bytes and the keystream. This bias was further improved in [79] by recovering $\bar{K}[i]$ instead of $K[i]$ to reduce the dependency between secret key bytes. We use Lemma 9 and explain how it can be merged with the weaknesses of the KSA (see Fig. 4).

The conditions, the attack path, the key recovery relation and the success probability of this attack are described below.

- **Conditions:** $(i - z_i) \notin \{S_t[t+1], \ldots, S_t[i-1]\}$ (Cond)
- **Attack path:** (see Fig. 4)
  - $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = S'_{i-1}[i] = S'_i[j'_i] = i - z_i$
- **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[i - z_i] - \sigma_i(t)$
- **Probability of success:** $P_{\mathrm{KI}}(i,t)$ (see below)

Exploiting the above correlation and the relations in the KSA and the PRGA, we have

1. $S'_i[j'_i] \overset{P_J}{=} i - z_i$ (Lemma 9)
2. $S'_i[j'_i] = S'_{i-1}[i]$
3. $S'_{i-1}[i] \overset{P_0}{=} S_i[i]$ (Lemma 8)
4. $S_i[i] = S_{i-1}[j_i]$
5. $S_{i-1}[j_i] \underset{\mathsf{Cond}'}{\overset{P_A^1}{=}} S_t[j_i]$ (where Cond' is the event that $j_i \le t$ or $j_i > i-1$.)
6. $j_i = \bar{K}[i] + \sum_{x=1}^{i} S_{x-1}[x]$ (Lemma 7)
7. $\sum_{x=1}^{i} S_{x-1}[x] \overset{P_B}{=} \sigma_i$ (Lemma 8)

We make the same heuristic assumption of independence as in Lemma 6 and Lemma 8. Then, we gain

$$P_{\mathrm{KI}}(i,t) = P_J \otimes P_0 \otimes P_A^1(i,t) \otimes P_B(i,t)$$

conditioned to Cond. Hence, the key recovery relation becomes

$$\bar{K}[i] \underset{\mathsf{Cond}}{\overset{P_{\mathrm{KI}}}{=}} S_t^{-1}[i - z_i] - \sigma_i(t)$$

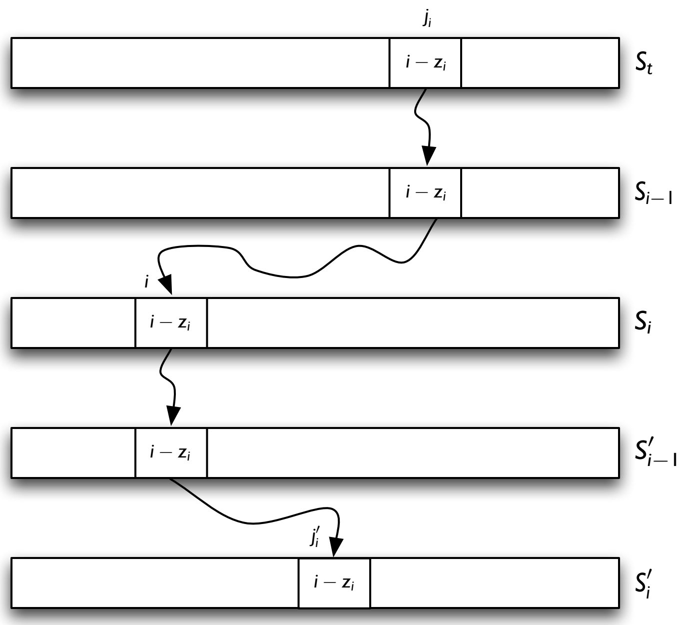

Fig. 4. RC4 state update in the Klein-Improved attack

6.2 The A_u15 Attack

Korek is the nickname of a hacker who described 20 key recovery attacks on RC4 [34,35]. A_u15 is one of the Korek attacks with the highest success probability. First, we introduce the conditions for this attack to succeed, the assumptions we make (attack path), the equation for the key recovery and the success probability. All other Korek attacks are described in Appendix C.

- **Conditions:** $S_t[i] = 0$ and $z_2 = 0$
- **Attack path:** (see Fig. 5)
  - $S_t[i] = \cdots = S_{i-1}[i]$
  - $S_i[2] = \cdots = S_{N-1}[2] = S'_1[2] = 0$
  - $j_i = 2$
- **Key recovery relation:** $\bar{K}[i] = 2 - \sigma_i(t)$
- **Probability of success:** $P_u^1(i,t)$ (see below)

We classify the conditions as

$$\mathsf{C}_1 : S_t[i] = 0 \quad \text{and} \quad \mathsf{C}_2 : z_2 = 0$$

We also classify the assumptions and the events and the key recovery equation as

$$\begin{cases} \mathsf{S}_1 : S_t[i] = \cdots = S_{i-1}[i] \\ \mathsf{S}_2 : S_i[2] = \cdots = S_{N-1}[2] = S'_1[2] \\ \mathsf{S}_3 : \bar{K}[i] = j_i - \sigma_i(t) \\ \mathsf{E}_1 : j_i = 2 \\ \mathsf{B} : \bar{K}[i] = 2 - \sigma_i(t) \end{cases}$$

Now, we compute the theoretical success probability of the attack. The goal is to estimate $\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2]$. So, we compute

$$\begin{aligned}\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2] &= \mathsf{Pr}[\mathsf{E}_1\mathsf{S}_3|\mathsf{C}] + \mathsf{Pr}[\mathsf{B}\neg\mathsf{S}_3|\mathsf{C}] \\ &= \mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_3\mathsf{C}]\ .\ \mathsf{Pr}[\mathsf{S}_3|\mathsf{C}] + \mathsf{Pr}[\mathsf{B}|\neg\mathsf{S}_3\mathsf{C}]\ .\ (1-\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}])\end{aligned}$$

Now,

$$\begin{aligned}\mathsf{Pr}[\mathsf{B}|\neg\mathsf{S}_3\mathsf{C}] &= \mathsf{Pr}[\mathsf{B}\neg\mathsf{E}_1|\neg\mathsf{S}_3\mathsf{C}] \\ &\approx \mathsf{Pr}[\mathsf{B}\neg\mathsf{E}_1|\mathsf{C}] \\ &= \mathsf{Pr}[\mathsf{B}|\neg\mathsf{E}_1\mathsf{C}]\ .\ \mathsf{Pr}[\neg\mathsf{E}_1|\mathsf{C}] \\ &\approx \tfrac{1}{N-1}\,(1-\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}])\end{aligned}$$

Overall,

$$\begin{aligned}\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2] &\approx \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}]\ .\ \mathsf{Pr}[\mathsf{S}_3|\mathsf{C}] + \left(\tfrac{1-\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}]}{N-1}\right)\ .\ (1-\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}]) \\ &= \mathsf{Pr}(\mathsf{E}_1|\mathsf{C})\ .\ \left(\tfrac{N\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}]-1}{N-1}\right) + \left(\tfrac{1-\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}]}{N-1}\right)\end{aligned}$$

Then, we approximate $\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}] \approx P_B(i,t)$ and we also have

$$\begin{aligned}\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] &= \mathsf{Pr}(\mathsf{C}_1|\mathsf{E}_1\mathsf{C}_2)\left(\tfrac{\mathsf{Pr}(\mathsf{E}_1|\mathsf{C}_2)}{\mathsf{Pr}(\mathsf{C}_1|\mathsf{C}_2)}\right) \\ &\approx \mathsf{Pr}(\mathsf{C}_1|\mathsf{E}_1\mathsf{C}_2) \\ &= \mathsf{Pr}(\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2|\mathsf{E}_1\mathsf{C}_2) + \mathsf{Pr}(\mathsf{C}_1\neg(\mathsf{S}_1\mathsf{S}_2)|\mathsf{E}_1\mathsf{C}_2) \\ &\approx \mathsf{Pr}(\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2|\mathsf{E}_1\mathsf{C}_2) + \tfrac{1}{N}(1-\mathsf{Pr}(\mathsf{S}_1\mathsf{S}_2|\mathsf{E}_1\mathsf{C}_2)) \\ &\approx \mathsf{Pr}(\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2|\mathsf{E}_1\mathsf{C}_2) + \tfrac{1}{N}\left(1-P_A^1(i,t)\ .\ \left(\tfrac{N-1}{N}\right)^{N-i}\right)\end{aligned}$$

$$\begin{aligned}\mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2|\mathsf{E}_1\mathsf{C}_2] &= \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{E}_1|\mathsf{C}_2]}{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_2]}\right) \\ &= \mathsf{Pr}[\mathsf{C}_2|\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{E}_1]\ .\ \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_2]\ .\ \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_2]}\right)\end{aligned}$$

Deploying Lemma 10, we obtain

$$\mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2|\mathsf{E}_1\mathsf{C}_2] = \frac{1}{2}P_A^1(i,t)\left(\frac{N-1}{N}\right)^{N-i}$$

Therefore, overall we have

$$\begin{aligned}P_u^1(i,t) \overset{\mathsf{def}}{=} \mathsf{Pr}[\mathsf{B}|\mathsf{C}_1\mathsf{C}_2] &= \left(\frac{NP_B(i,t)-1}{N-1}\right) \\ &\quad .\ \left[\frac{1}{2}P_A^1(i,t)\left(\frac{N-1}{N}\right)^{N-i} + \frac{1}{N}\left(1-P_A^1(i,t)\left(\frac{N-1}{N}\right)^{N-i}\right)\right] \\ &\quad + \left(\frac{1-P_B(i,t)}{N-1}\right)\end{aligned}$$

All the other correlations described in Appendix C and Table 3 are manipulated similarly. In the next few sections, we describe how to combine these biases to mount optimal key recovery attacks against WEP and WPA.

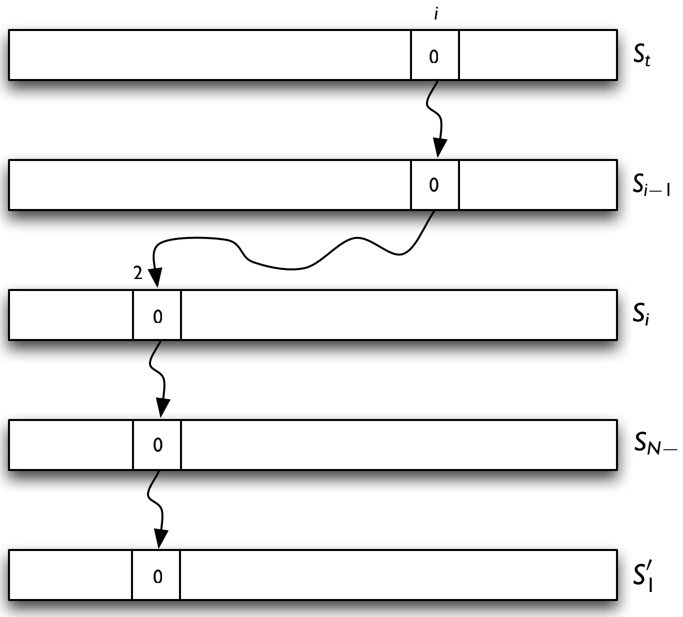

Fig. 5. RC4 state update in the A_u15 attack

7 Attacks on the WPA Protocol

In this section, we use the two biases described in the previous section and all the biases described in Appendix C to mount a key recovery attack against WPA. We first recover 8 bits of the WPA temporary key, and then use it to mount a key recovery attack against the full key. Recovering those 8 bits is performed in two steps: we first recover 7 of such bits (the first attack), and then the last bit (the second attack).

There are 8 bits of the TK that we call *weak*, because they have a simple relation with the bits of the PPK. These bits consist of the 7 most significant bits of the $\mathsf{TK}[0]$ and the least significant bit of the $\mathsf{TK}[1]$.

7.1 The First Attack: Recovering 7 Weak Bits of the TK

Given $\nu = (\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$, the adversary can compute $K[3] = \bar{K}[3] - \bar{K}[2]$ and $K[14] = \bar{K}[14] - \bar{K}[13]$. In WPA, we have

$$\mathsf{PPK}[5] = K[15] \| K[14]$$
$$K[3] = \mathsf{low8}((\mathsf{PPK}[5] \oplus (\mathsf{TK}[1] \| \mathsf{TK}[0])) \gg 1)$$

So, given ν, the adversary can compute $x = \mathsf{high7}(\mathsf{TK}[0])$ by

$$x = \mathsf{low7}((\bar{K}[3] - \bar{K}[2]) \oplus ((\bar{K}[14] - \bar{K}[13]) \gg 1))$$

We denote $N_\nu = 2^{32}$ the total number of possible ν's and $N_{\underline{x}} = 2^{\underline{7}}$ the total number of possible x's. Also let k be the total number of agglomerated biases we can use to vote for $\bar{K}[3]$, $\bar{K}[13]$ and $\bar{K}[14]$.

We can recover the 7 weak bits as follows: for each candidate value x (normally distributed), each packet m and each $\ell = 1, \ldots, k$ if the agglomerated bias condition holds, the biased equation ℓ gives us the value of the RC4 key byte on packet m, which is correct with probability p_ℓ. We let x be the suggested value of the 7 weak bits computed as explained. We let $X_{x,m,\ell}$ be some magical coefficient a_ℓ (to be optimized later) if the biased equation is indeed correct and 0 otherwise. We let $Y_x = \sum_{m=1}^{n}\sum_{\ell=1}^{k} X_{x,m,\ell}$, where n is the total number of packets to be used. Clearly, the correct value for ν is suggested with probability p_ℓ and others are obtained randomly. We assume incorrect ones are suggested with the same probability $\frac{1-p_\ell}{N_\nu-1}$.

If x is not the correct value, it is not suggested for sure when ν is correct. Since $\mathsf{low7}((\bar{K}[3]-\bar{K}[2])\oplus((\bar{K}[14]-\bar{K}[13])\gg 1))$ is balanced, this incorrect x has $\frac{N_\nu}{N_x}$ values ν belonging to the set of $N_\nu - 1$ incorrect ones. So, x is suggested with probability $\frac{N_\nu}{N_x}\times\frac{1-p_\ell}{N_\nu-1}$. Consequently, the $X_{x,m,\ell}$ for incorrect x's are random variables with the expected values

$$a_\ell q_\ell N_\nu \frac{1-p_\ell}{N_x(N_\nu-1)}$$

if x is not the correct value, where q_ℓ is the ℓ-th bias density.

If x is the correct value, it is suggested with probability p_ℓ for the correct ν and when ν is one of the $\frac{N_\nu-N_x}{N_x}$ (incorrect) preimages of x by $\mathsf{low7}((\bar{K}[3]-\bar{K}[2])\oplus((\bar{K}[14]-\bar{K}[13])\gg 1))$; that is, with overall probability $p_\ell + \frac{N_\nu-N_x}{N_x}\times\frac{1-p_\ell}{N_\nu-1}$. So, the $X_{x,m,\ell}$ for the correct x are random variables with expected values

$$a_\ell q_\ell N_\nu \frac{1-p_\ell}{N_x(N_\nu-1)} + a_\ell q_\ell \frac{N_\nu p_\ell - 1}{N_\nu - 1}$$

The difference between these two expected values is important. This is also the case for the difference of variances. Since every x is suggested with the probability roughly $\frac{q_\ell}{N_x}$, we assume that the variance of a bad $X_{x,m,\ell}$ can be approximated by $\frac{q_\ell}{N_x}\left(1-\frac{q_\ell}{N_x}\right)a_\ell^2$. Let Δ be the operator making the difference between the distributions for a good x and a bad one. We have

$$\begin{aligned}
E(Y_{x\ \mathsf{bad}}) &= \frac{n}{N_x\left(1-\frac{1}{N_\nu}\right)}\sum_\ell a_\ell q_\ell (1-p_\ell)\\
E(Y_{x\ \mathsf{good}}) &= E(Y_{x\ \mathsf{bad}}) + \Delta E(Y)\\
\Delta E(Y) &= \frac{n}{1-\frac{1}{N_\nu}}\sum_\ell a_\ell q_\ell \left(p_\ell - \frac{1}{N_\nu}\right)\\
V(Y_{x\ \mathsf{bad}}) &\approx n\sum_\ell a_\ell^2 \frac{q_\ell}{N_x}\left(1-\frac{q_\ell}{N_x}\right)\\
V(Y_{x\ \mathsf{good}}) &= V(Y_{x\ \mathsf{bad}}) + \Delta V(Y)\\
\Delta V(Y) &\approx \frac{n}{1-\frac{1}{N_\nu}}\sum_\ell a_\ell^2 q_\ell \left(p_\ell - \frac{1}{N_\nu}\right)
\end{aligned}$$

where $E(Y_{x\ \mathsf{bad}})$ and $V(Y_{x\ \mathsf{bad}})$ denote the expected value and the variance of a Y_x variable for any bad x respectively. Here, we remove the subscript x of Y_x in $\Delta E(Y)$, since this does not depend on a specific value for x. Let λ be such that $\Delta E(Y) = \lambda\sqrt{V(Y_{x\ \mathsf{bad}}) + V(Y_{x\ \mathsf{good}})}$. The probability that the correct Y_x is lower than an arbitrary wrong Y_x is $\rho = \varphi(-\lambda)$. See Section 4 for the definition of φ. That is, the expected number of wrong x's with larger Y_x is

$$r = (N_x - 1)\varphi(-\lambda) \tag{2}$$

So,

$$n = \frac{\lambda^2 \sum_\ell a_\ell^2 \left[2\left(\frac{q_\ell}{N_x}\right)\left(1-\frac{q_\ell}{N_x}\right)\left(1-\frac{1}{N_\nu}\right)^2 + q_\ell\left(p_\ell-\frac{1}{N_\nu}\right)\left(1-\frac{1}{N_\nu}\right)\right]}{\left(\sum_\ell a_\ell q_\ell\left(p_\ell - \frac{1}{N_\nu}\right)\right)^2}$$

By computing the derivative of n with respect to a_ℓ and set it to zero, we conclude that the optimal value of n is reached for

$$a_\ell = a_{\mathsf{opt}} \stackrel{\mathsf{def}}{=} \frac{\left(p_\ell - \frac{1}{N_\mathrm{v}}\right)}{\left(p_\ell - \frac{1}{N_\mathrm{v}}\right) + \frac{2}{N_x}\left(1 - \frac{1}{N_\mathrm{v}}\right)\left(1 - \frac{q_\ell}{N_x}\right)}$$

Hence, we obtain

$$n = n_{\mathsf{opt}} \stackrel{\mathsf{def}}{=} \frac{\lambda^2\left(1 - \frac{1}{N_\mathrm{v}}\right)}{\sum_\ell a_\ell q_\ell \left(p_\ell - \frac{1}{N_\mathrm{v}}\right)} \tag{3}$$

In [66], it was assumed that $\Delta V(Y) = 0$ and the value for n_{opt} and a_{opt} were different. However, our experiments have shown that this approximation was not sound. This is why we integrate $\Delta V(Y)$ here. The attack works as follows:

The goal is to recover $(\bar{K}_3, \bar{K}_{13}, \bar{K}_{14})$, using $\bar{K}[0]$, $\bar{K}[1]$, $\bar{K}[2]$.
Initialize the Y_x counters to 0.
for $m = 1$ to n **do**
  for $\ell = 1$ to k **do**
    if the bias condition holds **then**
      Compute the suggested value for $(\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$.
      Compute x, using the tuple above.
      Increment Y_x by a_ℓ.
    end if
  end for
end for
Output $x = \arg\max_x Y_x$.

Clearly, the time complexity is nk. The complexity is measured in terms of the number of times the **if** structure is executed. This should have a complexity which is essentially equivalent to executing the PHASE2 of the key derivation. The memory complexity has the order of magnitude of N_x. Here is another variant of the algorithm:

The goal is to recover $(\bar{K}_3, \bar{K}_{13}, \bar{K}_{14})$, using $\bar{K}[0]$, $\bar{K}[1]$, $\bar{K}[2]$.
Initialize a table y_x^μ to 0.
for $\ell = 1$ to k **do**
  for all μ's that satisfy the g_ℓ conditions **do**
    Compute x.
    Increment y_x^μ by a_ℓ.
  end for
end for
Initialize the Y_x counters to 0.
for $m = 1$ to n **do**
  for all x **do**
    Compute μ.
    Increment Y_x by y_x^μ.
  end for
end for
Output $x = \arg\max_x Y_x$.

where μ is the vector of all z_i's and clue bytes appearing in any of the biased relations. Now, the time complexity is $N_\mu k + N_x n$ and the memory complexity is $N_\mu N_x$, where N_μ is N raised to the power of the number of z_i bytes and the clue bytes appearing in any of the biased relations. So, the complexity is

$$c = \min(nk, N_\mu k + N_x n) \tag{4}$$

The two complexity curves intersect for $n = N_\mu \frac{k}{k-N_x} \approx N_\mu$ when $N_x \ll k$.

For $\nu = (\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$, we have $N_\nu = 2^{32}$, $N_\mu = 2^{48}$ and $N_x = 2^7$. The complexities with and without using conditional biases are summarized in Table 1. As we can see, when ignoring the conditional biases we need about 65% more packets, but the complexity is much lower because k is smaller. So, the conditional biases do not seem to be useful in this case.

7.2 The Second Attack: Recovering One Weak Bit of the TK

Let $x = \mathsf{low1}(\mathsf{TK}[1])$ be the last weak bit. Given the IV and also $\nu = (\bar{K}[2], \bar{K}[3], \bar{K}[14], \bar{K}[15])$, we deduce x by

$$x = \mathsf{high1}((\bar{K}[3] - \bar{K}[2]) \oplus (\bar{K}[15] - \bar{K}[14]))$$

So, we apply the first attack with $N_x = 2$. Since $\bar{K}[15]$ in involved, we have more biases. We have r, n and c from Eq. (2), Eq. (3) and Eq. (4) respectively.

For $\nu = (\bar{K}[2], \bar{K}[3], \bar{K}[14], \bar{K}[15])$, we have $N_\nu = 2^{32}$, $N_\mu = 2^{120}$ and $N_x = 2$. The complexities are summarized in Table 1. Again, conditional biases are not very useful. We can also see that this choice of ν leads to a much better attack than the one from Section 7.1 in terms of n, but the complexity is slightly higher. This is due to a larger k.

7.3 Merging the First and the Second Attacks

In this section, we merge the first and the second attack on WPA, to recover its 8 weak bits. Given two attacks for recovering independent x^1 (resp. x^2) random variables leading to Y_{x^1} (resp. Y_{x^2}), c^1 (resp. c^2), n^1 (resp. n^2) and λ^1 (resp. λ^2), one problem is to merge the sorted lists of x^1 and x^2 to obtain a sorted list of all (x^1, x^2) pairs. The problem is to find the best ordering of these pairs to minimize the expected complexity for finding the good pair in an exhaustive search going through this list. One can follow the approach by Junod-Vaudenay [30]. They proved that the best mixing paradigm consists of sorting the (x^1, x^2) following their likelihood ratio, which is obtained by multiplying the likelihood ratio of x^1 and of x^2. They showed that this ranking procedure minimizes the costs of the attacks exhaustive search.

We assume that all Y_{x^i}'s are independent, normally distributed with the variance either $V(Y_{x^i\ \mathsf{bad}})$ or $V(Y_{x^i\ \mathsf{good}}) = V(Y_{x^i\ \mathsf{bad}}) + \Delta V(Y_{x^i})$ and the expected value either $E(Y_{x^i\ \mathsf{bad}})$ or $E(Y_{x^i\ \mathsf{good}}) = E(Y_{x^i\ \mathsf{bad}}) + \Delta E(Y_{x^i})$. Given x^i, the ratio for x^i being the correct value based on the observation Y_{x^i} is

$$\frac{\Pr[Y_{x^i}|x^i\ \mathsf{good}]}{\Pr[Y_{x^i}|x^i\ \mathsf{wrong}]} = \frac{\frac{1}{\sqrt{2\pi V\left(Y_{x^i\ \mathsf{good}}\right)}} e^{-\frac{\left(Y_{x^i} - E\left(Y_{x^i\ \mathsf{good}}\right)\right)^2}{2V\left(Y_{x^i\ \mathsf{good}}\right)}}}{\frac{1}{\sqrt{2\pi V\left(Y_{x^i\ \mathsf{bad}}\right)}} e^{-\frac{\left(Y_{x^i} - E\left(Y_{x^i\ \mathsf{bad}}\right)\right)^2}{2V\left(Y_{x^i\ \mathsf{bad}}\right)}}}$$

$$= \sqrt{\frac{V\left(Y_{x^i\ \mathsf{bad}}\right)}{V\left(Y_{x^i\ \mathsf{good}}\right)}} e^{\frac{\left(Y_{x^i} - E\left(Y_{x^i\ \mathsf{bad}}\right)\right)^2}{2V\left(Y_{x^i\ \mathsf{bad}}\right)} - \frac{\left(Y_{x^i} - E\left(Y_{x^i\ \mathsf{good}}\right)\right)^2}{2V\left(Y_{x^i\ \mathsf{good}}\right)}}$$

So, when multiplying some terms of this form for the pairs of values, sorting them by decreasing product is equivalent to sorting them by decreasing value of

$$\begin{aligned}&\tfrac{1}{2}\left(\tfrac{1}{V_{1b}} - \tfrac{1}{V_{1g}}\right) Y_{x^1}^2 + \left(\tfrac{E_{1g}}{V_{1g}} - \tfrac{E_{1b}}{V_{1b}}\right) Y_{x^1} + \tfrac{1}{2}\left(\tfrac{1}{V_{2b}} - \tfrac{1}{V_{2g}}\right) Y_{x^2}^2 + \left(\tfrac{E_{2g}}{V_{2g}} - \tfrac{E_{2b}}{V_{2b}}\right) Y_{x^2} \\ &= a\left(Y_{x^1} - \beta_1\right)^2 + b\left(Y_{x^2} - \beta_2\right)^2\end{aligned}$$

where

$$\begin{aligned} V_{1g} &= V(Y_{x^1\ \text{good}}) & V_{2g} &= V(Y_{x^2\ \text{good}}) \\ V_{1b} &= V(Y_{x^1\ \text{bad}}) & V_{2b} &= V(Y_{x^2\ \text{bad}}) \\ \Delta V_1 &= \Delta V(Y_{x^1}) & \Delta V_2 &= \Delta V(Y_{x^2}) \end{aligned}$$

$$\begin{aligned} E_{1g} &= = E(Y_{x^1\ \text{good}}) & E_{2g} &= E(Y_{x^2\ \text{good}}) \\ E_{1b} &= = E(Y_{x^1\ \text{bad}}) & E_{2b} &= E(Y_{x^2\ \text{bad}}) \end{aligned}$$

$$\begin{aligned} a &= \tfrac{1}{2}\left(\tfrac{1}{V_{1b}} - \tfrac{1}{V_{1g}}\right) & b &= \tfrac{1}{2}\left(\tfrac{1}{V_{2b}} - \tfrac{1}{V_{2g}}\right) \\ \beta_1 &= \left(\tfrac{V_{1g}E_{1b} - V_{1b}E_{1g}}{\Delta V_1}\right) & \beta_2 &= \left(\tfrac{V_{2g}E_{2b} - V_{2b}E_{2g}}{\Delta V_2}\right) \end{aligned}$$

So we let $Y_{x^1,x^2} = a(Y_{x^1} - \beta_1)^2 + b(Y_{x^2} - \beta_2)^2$. With the same assumptions as in [30], Y_{x^1,x^2} is distributed with the Generalized-χ^2 distribution [9,10]. The average number of the wrong (x^1, x^2) pairs with higher score than the good one is

$$r = (N_{x^1}N_{x^2} - 1)\ .\ \Pr\left(Y_{x^1,x^2\ \text{good}} - Y_{x^1,x^2\ \text{bad}} < 0\right)$$

Thus, we define a new random variable

$$\Delta Y_{x^1,x^2} = \sum_{m=1}^{2}\sum_{j=b,g} a_{mj}\left[\frac{(Y_{x^m j} - \beta_m)^2}{V_{mj}}\right]$$

where

$$\begin{aligned} a_{1g} &= aV_{1g} & a_{1b} &= -aV_{1b} & Y_{x^i g} &= Y_{x^i\ \text{good}} \\ a_{2g} &= bV_{2g} & a_{2b} &= -bV_{2b} & Y_{x^i b} &= Y_{x^i\ \text{bad}} \end{aligned}$$

$\Delta Y_{x^1,x^2}$ is a quadratic form in independent normal random variables. It can be expressed as the linear combination

$$\Delta Y_{x^1,x^2} = \sum_{m=1}^{2}\sum_{j=b,g} a_{mj}X_{mj}^2 \tag{5}$$

where X_{mj}'s are independent and normally distributed random variables with variance one. We write

$$t_{mj}^2 = \frac{\left(E(Y_{mj}) - \beta_m\right)^2}{V(Y_{mj})} = t_{mj}'^2\ .\ n$$

The characteristic function of a quadratic form in independent normal random variables $\Delta Y_{x^1,x^2}$ is given by Davies [9]:

$$\varphi_{\Delta Y_{x^1,x^2}}(u) = E(e^{iu\Delta Y_{x^1,x^2}}) = \frac{e^{iu\left(\sum_{m=1}^{2}\sum_{j=b,g}\frac{a_{mj}t_{mj}^2}{1-2iua_{mj}}\right)}}{\prod_{m=1}^{2}\prod_{j=b,g}(1-2iua_{mj})^{\frac{1}{2}}}$$

If $E(|\Delta Y_{x^1,x^2}|)$ is finite, it follows from Gil-Pelaez [17] that

$$F_{\Delta Y_{x^1,x^2}}(w) = \Pr(\Delta Y_{x^1,x^2} < w) = \frac{1}{2} - \int_{-\infty}^{\infty} \mathrm{Im}\left(\frac{\varphi_{\Delta Y_{x^1,x^2}}(u)e^{-iuw}}{2\pi u}\right) du$$

Substituting what we have, one derives

$$F_{\Delta Y_{x^1,x^2}}(0) = \mathsf{Pr}(\Delta Y_{x^1,x^2} < 0) = \frac{1}{2} - \int_{-\infty}^{\infty} \mathsf{Im}\left(\frac{e^{iu\left(\sum_{m=1}^{2}\sum_{j=b,g}\frac{a_{mj}t_{mj}^2}{1-2iua_{mj}}\right)}}{2\pi u \prod_{m=1}^{2}\prod_{j=b,g}(1-2iua_{mj})^{\frac{1}{2}}} \right) du$$

Finally, setting r, the value of n can be numerically computed.

It might be of interest to evaluate n analytically. In Eq. (5), the X_i^2's follow the non-centralized χ^2 distribution. Our experiments revealed that their non-centrality parameters are large. Let n_i and t_i^2 be their corresponding degrees of freedom and non-centrality parameters respectively. It was shown in [46] that when $n_i \to \infty$ or $t_i^2 \to \infty$, the non-centralized χ^2 random variable can be approximated by normal distribution with the same expected value and variance. Using this approach, the above integral can be avoided. Hence,

$$E(\Delta Y_{x^1,x^2}) \approx \sum_{m=1}^{2}\sum_{j=b,g} a_{mj}\left(1+t_{mj}^2\right)$$

$$V(\Delta Y_{x^1,x^2}) \approx \sum_{m=1}^{2}\sum_{j=b,g} 2a_{mj}^2\left(1+2t_{mj}^2\right)$$

To find n, we need to solve the following equation.

$$\left(\frac{-E(\Delta Y_{x^1,x^2})}{\sqrt{V(\Delta Y_{x^1,x^2})}}\right) = \varphi^{-1}\left(\frac{r}{N_{x^1}N_{x^2}-1}\right)$$

Thus, we derive

$$n \approx \left[\frac{1}{\mu}\varphi^{-1}\left(\frac{r}{N_{x^1}N_{x^2}-1}\right)\right]^2$$

where

$$\mu = \frac{\sum_{m=1}^{2}\sum_{j=b,g} a_{mj}t'_{mj}}{\sqrt{\sum_{m=1}^{2}\sum_{j=b,g} 4a_{mj}^2 t'_{mj}}}$$

We can use these merging rules to merge the two previous attacks. We have $c = c^1 + c^2$ by using Eq. (4) for c^1 and c^2. We obtain the results in Table 1.

Table 1 represents the corresponding complexities when merging the previous attacks to recover the 8 weak bits of the TK. We also compare these attack using a merged set ν directly. As we can see, merging the attacks with small ν's (reference 3 in Table 1) is much better than making a new attack with a merged ν (reference 4).

7.4 Temporary Key Recovery Attack on WPA

The results from [45] lead to an "easy" attack on WPA: guess the 96-bit PPK and the 8 weak bits of the TK with an average complexity of 2^{103} until it generates the correct keystream. Then, guess the 96-bit PPK of another packet in the same segment (with the weak bits already known). Then, apply the method of [45] to recover the TK. We improve this attack by recovering the weak bits of the TK separately: we know from Table 1 that we can recover the weak bits of the TK by using 2^{42} packets. After having recovered the weak bits, we note that the 96-bit PPK is now enough

Table 1. The complexities of several attacks to recover $\log_2 N_x$ bits of the TK. We compare them when including conditional biases and without. We provide the number of packets n, the running time complexity c, the expected number r of the better wrong values, as well as the parameters k, λ and N_ν. Except when $N_x = 2$, for which it would not make any sense, we target $r = \frac{1}{2}$ (that is, the correct value has the higher score in half of the cases). We used $\bar{K}[0], \bar{K}[1], \bar{K}[2]$.

	reference	ν	n	c	r	N_x	k	λ	N_ν	N_μ	cond. biases
1u		$(\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$	$2^{42.10}$	$2^{42.10}$	$\frac{1}{2}$	2^7	1	2.66	2^{32}	N^6	without
1c		$(\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$	$2^{41.38}$	$2^{53.10}$	$\frac{1}{2}$	2^7	$2^{11.72}$	2.66	2^{32}	N^8	with
2u		$(\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[14])$	$2^{40.38}$	$2^{45.38}$	$\frac{1}{4}$	2	2^5	0.67	2^{32}	N^{15}	without
2c		$(\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[14])$	$2^{39.12}$	$2^{55.85}$	$\frac{1}{4}$	2	$2^{16.73}$	0.67	2^{32}	N^{17}	with
3u	merge 1u+2u		$2^{41.83}$	$2^{46.87}$	$\frac{1}{2}$	2^8					without
3c	merge 1c+2c		$2^{41.22}$	$2^{57.99}$	$\frac{1}{2}$	2^8					with
4u		$(\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$	$2^{51.72}$	$2^{57.72}$	$\frac{1}{2}$	2^8	2^6	2.88	2^{40}	N^{17}	without
4c		$(\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$	$2^{51.05}$	$2^{72.69}$	$\frac{1}{2}$	2^8	$2^{21.64}$	2.88	2^{40}	N^{19}	with

to recalculate the $\mathsf{RC4KEY}$. So, we can do an exhaustive search on the PPK for a given packet until we find the correct one. This works with average complexity of 2^{95}. We do it twice to recover the PPK of two packets in the same segment. Given these two PPK sharing the same $\mathsf{IV32}$, we recover the TK by using the method of [45]. Therefore, we can recover the temporary key TK and decrypt all packets with complexity 2^{96}. The number of packets needed to recover the weak bits is 2^{42}.

7.5 Distinguishing WPA

RC4 can be distinguished using N packets [41] and since WPA's output is already an output of RC4, it can be simply distinguished from random using a few packets. However, the distinguisher of [41], based on the bias of z_2, can not distinguish two protocols that are both using RC4. In this section, we are using all the biases on RC4 together with some weaknesses in the structure of WPA and mount a distinguishing attack on WPA. This distinguisher is also capable of distinguishing WPA from other protocols using RC4. The first attack can be turned into a distinguisher as follows. The expected value and the variance of the correct Y_x are

$$E(Y_{x\ \mathsf{good}}) = E(Y_{x\ \mathsf{bad}}) + \lambda\sqrt{V(Y_{x\ \mathsf{bad}}) + V(Y_{x\ \mathsf{good}})}$$
$$V(Y_{x\ \mathsf{good}}) = V(Y_{x\ \mathsf{bad}}) + \Delta V(Y)$$

Lets extend our notations by defining

$$\gamma = \left(\frac{V(Y_{x\ \mathsf{good}})}{V(Y_{x\ \mathsf{bad}})}\right)$$

The random variable Y_x of the good counter is larger than

$$T = E(Y_{x\ \mathsf{bad}}) + \lambda'\sqrt{V(Y_{x\ \mathsf{bad}}) + V(Y_{x\ \mathsf{good}})}$$

with probability $\varphi\left((\lambda-\lambda')\sqrt{1+\frac{1}{\gamma}}\right)$. Now, if we replace the WPA packets by a sequence generated by RC4 fed with random keys, all the counters have the expected value $E(Y_{x\ \mathsf{bad}})$ and the variance approximately $V(Y_{x\ \mathsf{bad}})$. The probability that a given counter exceeds T is $\varphi(-\lambda'\sqrt{1+\gamma})$. The probability that any counter exceeds this is lower than $N_x\varphi(-\lambda'\sqrt{1+\gamma})$. So, the condition $\max_x Y_x > T$ makes a distinguisher of the same n and c as in the first attack and with $\mathsf{Adv} \geq \beta$, where

$$\beta = \varphi\left((\lambda-\lambda')\sqrt{1+\frac{1}{\gamma}}\right) - N_x\varphi\left(-\lambda'\sqrt{1+\gamma}\right) \tag{6}$$

Finally, we find the optimal λ' which maximizes the advantage.

$$\lambda' = \frac{\sqrt{\left(1+\frac{1}{\gamma}\right)^2 \lambda^2 + \left(\gamma - \frac{1}{\gamma}\right)\left[\left(1+\frac{1}{\gamma}\right)\lambda^2 + 2\ln\left(N_x\sqrt{\gamma}\right)\right]} - \left(1+\frac{1}{\gamma}\right)\lambda}{\left(\gamma - \frac{1}{\gamma}\right)}$$

We use the same values as before and target $\mathsf{Adv} \geq \frac{1}{2}$. We use Eq. (3) for n, Eq. (4) for c and Eq. (6) for a lower bound β of the advantage. The performances of the distinguishers are summarized in Table 2. Again, the attack based on $\nu = (\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[14])$ is better in terms of the number of packets, but is not in terms of the complexity. It works using $2^{41.23}$ packets and complexity of $2^{46.23}$. The one based on $\nu = (\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$ works with 50% more packets ($2^{41.83}$) with no conditional biases, but with a much better complexity of $2^{41.83}$.

Table 2. The complexities of several distinguishers for WPA. We compare them when including conditional biases and without. We list the number of packets n, the running time complexity c, the bound on the advantage β, as well as the parameters k, λ and N_ν. We target $\beta = \frac{1}{2}$. We used $\bar{K}[0], \bar{K}[1], \bar{K}[2]$.

	ν	n	c	β	N_x	k	λ	N_ν	N_μ	cond. biases
1u	$(\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$	$2^{41.83}$	$2^{41.83}$	0.5	2^7	1	2.42	2^{32}	N^6	without
1c	$(\bar{K}[2], \bar{K}[3], \bar{K}[13], \bar{K}[14])$	$2^{41.11}$	$2^{52.83}$	0.5	2^7	$2^{11.72}$	2.42	2^{32}	N^8	with
2u	$(\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[14])$	$2^{41.23}$	$2^{46.23}$	0.5	2	2^5	1.28	2^{32}	N^{15}	without
2c	$(\bar{K}[15], \bar{K}[2], \bar{K}[3], \bar{K}[14])$	$2^{40.97}$	$2^{57.70}$	0.5	2	$2^{16.73}$	1.28	2^{32}	N^{17}	with

Our distinguisher has recently been improved by Sen Gupta et al. [60]. Their distinguisher requires only 2^{19} packets to distinguish WPA from any other protocol based on RC4.

8 Tornado Attack on WEP

In this section, we present an attack on WEP using the theory we already built.

8.1 Passive vs. Active Attacks

For WEP, we consider both passive and active adversaries. In an active attack, the attacker eavesdrops the ARP packets and since most of the plaintext bytes are known (up to the 32-nd byte), she can compute $z_1, \ldots, z_{32}$ values using the ciphertext. It is also possible to inject data into the network. Because the ARP replies expire quickly (resetting the ARP cache), it usually takes only a few seconds or minutes until an attacker can capture an ARP request and start re-injecting it [72]. However, active attacks are detectable by Intrusion Detection systems (IDS). Moreover, some network cards require extra driver patches to be able to inject data into the traffic. This is not available for all network cards. On the other hand, a passive attacker can eavesdrop the wireless communication channel for TCP/IPv4 packets, without any need to inject data. The caveat is that more data frames are unknown in this case compared to the ARP packets (see the Appendix of [79]). As presented in Table 3, the Klein-Improved attack requires z_i to recover $\bar{K}[\mathrm{i}]$. Hence in reality, we are not able to use this attack to recover some bytes of the key. Instead, we use Korek attacks, since they only need z_1 and z_2 to operate. To summarize, we need more packets in a passive attack compared to an active attack. The difference in the number of packets is presented in Fig. 10.

8.2 Technical Details

We recover $\bar{K}[15]$, $\bar{K}[3], \ldots, \bar{K}[14]$ sequentially. We initially set $t = 2$ and use S_2 to recover $\bar{K}[15]$. Using the same state, we recover $\bar{K}[3]$. Next, we update the state to S_3 and recover $\bar{K}[4]$. We repeat the same steps until we recover $\bar{K}[14]$. When the full key is recovered, we test it. If it is not correct, we test more keys by re-voting (see below for more details). We call this attack Tornado Attack, because we use a theory from tornado analysis.

We apply the first attack on WPA (see Section 7.1) with $x = \nu$: we only recover the key bytes which are the same for all packets. This attack produces a ranking of all the possible x's in the form of a list $\mathcal{L}$ by decreasing order of likelihood. The attack works as in Fig. 6.

Compute the ranking $\mathcal{L}_{15}$ for $\bar{K}[15]$, using $\bar{K}[0], \bar{K}[1], \bar{K}[2]$.
Truncate $\mathcal{L}_{15}$ to its first ρ_{15} terms.
for each $\bar{k}_{15}$ in $\mathcal{L}_{15}$ **do**
    Run the recursive attack on the input $\bar{k}_{15}$.
end for
Stop: Attack failed.
Recursive attack with input $(\bar{k}_{15}, \bar{k}_3, \ldots, \bar{k}_{i-1})$**:**
if The input is only $\bar{k}_{15}$ **then**
    Set $i = 3$.
end if
if $i \leq i_{\max}$ **then**
    Compute the ranking $\mathcal{L}_i$ for $\bar{k}[i]$, having $(\bar{k}[0], \ldots, \bar{k}[i-1], \bar{k}[15])$.
    Truncate $\mathcal{L}_i$ to its first ρ_i terms.
    for each $\bar{k}_i$ in $\mathcal{L}_i$ **do**
        Run the recursive attack on the input $(\bar{k}_{15}, \bar{k}_3, \ldots, \bar{k}_{i-1}, \bar{k}_i)$.
    end for
else
    for each $\bar{k}_{i_{\max}+1}, \ldots, \bar{k}_{14}$ **do**
        Test the key $(\bar{k}_3, \ldots, \bar{k}_{14}, \bar{k}_{15})$ and stop if it is correct.
    end for
end if

Fig. 6. Tornado attack on WEP

In the following, we compute the values of the ρ_i's, such that the success probability becomes 50% and the attack complexity is minimized.

Let $N_x = N_\nu = N$, r_i and c_i be their parameters following Eq. (2,4). Let R_i be the rank of the correct $\bar{k}_i$ value in $\mathcal{L}_i$. Let define a random variable $U_{ij} = 1_{(Y_{x^i \text{ good}} < Y_{x^i \text{ bad}^j})}$, where $Y_{x^i \text{ bad}^j}$ is the j-th bad counter in attacking $\bar{K}[i]$. Hence, we have

$$R_i = \sum_{j=1}^{N_x - 1} U_{ij}$$

The expected value and the variance of this random variable are:

$$r_i = E(R_i) = (N_x - 1)\varphi(-\lambda_i)$$

$$\text{and} \tag{7}$$

$$E(R_i^2) = E(R_i) + (N_x - 1)(N_x - 2)\ .\ E(U_{i1}.U_{i2})$$

where

$$E(U_{i1}.U_{i2}) = \frac{1}{\sqrt{2\pi V(Y_{x^i \text{ good}})}} \int_{-\infty}^{\infty} e^{-\frac{\left(Y - E(Y_{x^i \text{ good}})\right)^2}{2V(Y_{x^i \text{ good}})}} \left(1 - \varphi\left(\frac{Y - E(Y_{x^i \text{ bad}})}{\sqrt{V(Y_{x^i \text{ bad}})}}\right)^2\right) dY$$

This finally yields

$$V(R_i) = (N_x-1)\varphi(-\lambda_i) + (N_x-1)(N_x-2)\ .\ E(U_{i1}.U_{i2}) - (N_x-1)^2\varphi(-\lambda_i)^2 \tag{8}$$

In [66], U_{i1} and U_{i2} were incorrectly assumed to be independent, leading to

$$V(R_i) \approx (N_x-1)\varphi(-\lambda_i)(1-\varphi(\lambda_i)) \approx r_i$$

which did not match our experiments. Now, the fundamental question is: What is the distribution of R_i?

8.3 Analysis Based on Pólya Distribution

In [66], it was assumed that the distribution of R_i is normal. Running a few experiments, we noticed that in fact it is following a distribution very close to the Poisson distribution. A revealing observation was that the variance of the distribution was much higher than the expected value. A number of distributions have been devised for series in which the variance is significantly larger than the mean [2,13,49], frequently on the basis of complex biological models [7]. The first of these was the negative binomial, which arose in deriving the Poisson series from the point binomial [70,82]. We use a generalized version of the negative binomial distribution called the Pólya distribution. The main application of the Pólya distribution is in *Tornado Outbreaks* [74] and *Hail Frequency* analysis [73].

In most climates, the probability of hail is small. If the mean hail frequency ranges on an interval $f_1 < f < f_2$ for all climates, it was observed that for values of f near f_1 the hail storms are quite scattered through each year. For this case, the hail storms might be considered independent of each other. In this instance, the series of annual frequencies of hail events are expected to follow the Poisson distribution of rare events. On the other hand, if the mean hail frequency is near f_2, then it seems reasonable to assume that the successive hail storms may no longer be independent, and if one storm had hail, the next storm would be more likely to have hail as well. The introduction of dependence between successive storms leads to the negative binomial distribution [73]. Similarly, tornadoes tend to cluster within years and follow a Pólya process rather than a Poisson process in areas where frequency of the occurrence is high [74].

This observation led us to find out that R_i is in fact following the Pólya distribution (see Definition 2). To be more precise, if two events occur with Poisson distribution and their expected values are very low, then it can be assumed that those events are happening independently. On the other hand, for Poisson events with high expected values (approximated as normal), the occurrence of the former event may increase the probability of the latter. In such cases, the overall distribution would be the Pólya distribution. Regarding the current problem, the events $(Y_{x^i\ \mathrm{good}} < Y_{x^i\ \mathrm{bad}^j})$ and $(Y_{x^i\ \mathrm{good}} < Y_{x^i\ \mathrm{bad}^{j'}})$ are not independent. Therefore, they tend to follow the Pólya distribution. Since $E(R_i)$ and $V(R_i)$ are known from Eq. (7) and Eq. (8), the values p_i and r_i for attacking $\overline{K}[i]$ can be simply computed by

$$p_i = \left(1 - \frac{E(R_i)}{V(R_i)}\right) \qquad \text{and} \qquad r_i = \left(\frac{E(R_i)^2}{V(R_i) - E(R_i)}\right)$$

As a proof of concept, we have sketched the probability distribution of R_3 for 5 000 packets. The corresponding parameters for the Pólya distribution are $p = 0.9839$ and $r = 0.356$ (see Fig. 7). As can be observed, those two distributions are extremely close. Also,

$$u_i \stackrel{\mathsf{def}}{=} \Pr[R_i \le \rho_i - 1] = 1 - I_{p_i}(\rho_i, r_i)$$

where I is the regularized incomplete beta function. Overall, the success probability is

$$u = u_{15} \prod_{i=3}^{i_{\max}} u_i$$

and the complexity is

$$c = c_{15} + \rho_{15}\left(c_3 + \rho_3\left(c_4 + \rho_4\left(\cdots c_{i_{\max}} + \rho_{i_{\max}} N^{14-i_{\max}} \cdots\right)\right)\right)$$

To be able to compare our results with the state of the art, we set $u = 50\%$. To approximate the optimal choice of ρ's, let $i_{\max} = 14$. We have to deal with the following optimization problem:

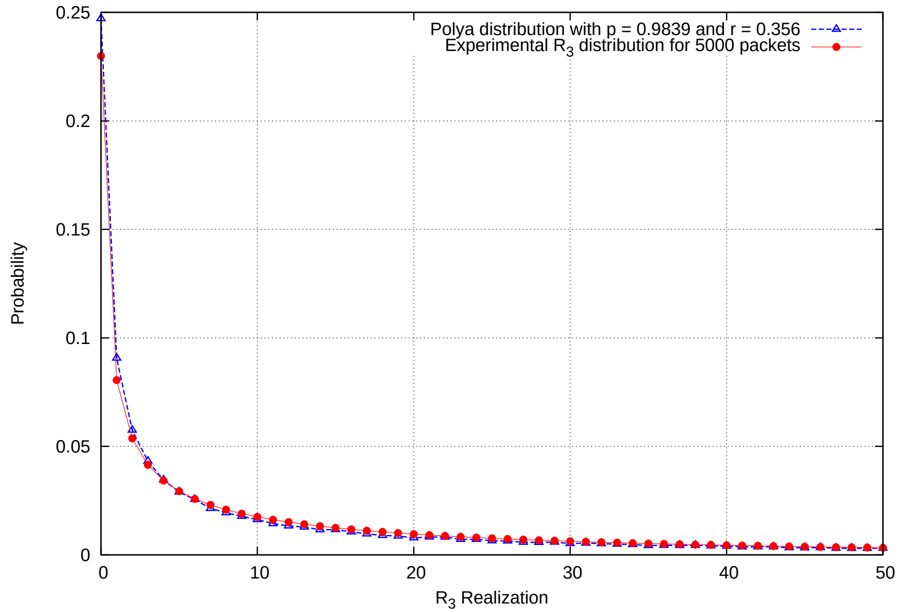

Fig. 7. R_3 distribution using 5 000 packets following the Pólya distribution

$$\text{Minimize } c \text{ in terms of the } \rho_i\text{'s, with the constraint that } u = \prod_{i=3}^{15}(1 - I_{p_i}(\rho_i, r_i)) = \frac{1}{2}$$

To solve this optimization problem, we compare three distinct approaches:

– To obtain the probability of 50%, we let the probabilities u_i's be equal for all $i \in \{3, \ldots, 15\}$. Hence, we set

$$(1 - I_{p_i}(\rho_i, r_i)) = 2^{\left(\frac{-1}{i_{\max}-1}\right)} = 0.9481$$

and we find the corresponding ρ_i's. This approach does not yield the optimal solution, but at least it gives a benchmark on what we should expect.

– Another approach is to use Lagrange multipliers to find the optimal solution. We used the fmincon function in Maltab with Sequential Quadratic Programming [50] (SQP) algorithm as the default algorithm to compute the local minimum. This algorithm was very fast and stable compared to the Genetic algorithm which is explained next. Since this algorithm needs a starting point x0 for its computations, we used the GlobalSearch class which iterates the fmincon function multiple times using random vectors for x0. Simultaneously, it checks how the results merge towards the global minimum. The drawback of any Lagrange multiplier approach is that the algorithm should be fed with a continuous objective function. This is because it has to compute derivatives. Since we need integer values for ρ_i's in practice, we had to relax the outputs by the ceil function to round up the ρ_i's found by this approach. Therefore, it does not guarantee that the optimal solution is found, but it finds an answer very close to optimal. As our experiments revealed, this algorithm most often sets $\rho_{14} = N$. So, using this approach, $i_{\max} = 13$ and we do not often need to vote for $\bar{K}[14]$.

– The last approach is to find an algorithm which can handle discrete functions, i.e., it accepts integers as input. One option is to use a Genetic algorithm. We used the ga function in Matlab for this purpose. Since these algorithms are evolutionary, the drawback is that with the same parameters, each run outputs different results. So, we have to run the algorithm multiple times and pick the best solution. The other drawback is that it can find a local minimum and does not guarantee to find the global optima. As can be observed in Fig. 8, this method is not as stable as the other approaches. Plus the experiment time is much longer than the other methods. To obtain a stable result, the parameters of the Genetic algorithm should be carefully set. This approach often yields a high value for ρ_{15}, but it is often less than N.

Using the empirical distribution of R_i's and by deploying the Genetic algorithm approach, we computed the experimental curve for the complexity. We have depicted the result of all these three approaches in Fig. 8.

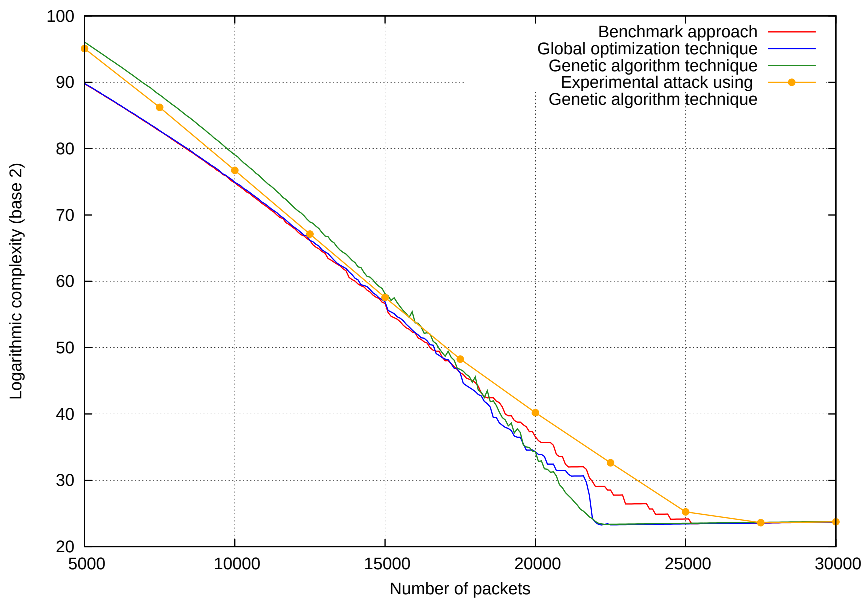

Fig. 8. Theoretical and experimental logarithmic complexity in terms of the data complexity for breaking a WEP key with probability at least 50% with respect to three distinct optimization approaches: the Benchmark approach, the Global optimization technique and the Genetic algorithm technique.

We call the optimized key ranking attack on RC4, *"Tornado Attack"*, since R_i's follow a similar distribution as tornadoes occurrences.

Recovering $\bar{\mathbf{K}}[15]$, Theory vs Practice. Recovering $\bar{K}[15]$ is a crucial step in the WPA and WEP attacks. We compare the theoretical and experimental success probability of recovering $\bar{K}[15]$. In this specific example, we only evaluate the first element in the sorted list. In [66], it is assumed that $Y_{x\text{ good}} - Y_i$ is independent for all bad i's and was deduced that the good x had a top Y_x with probability $(1-\varphi(-\lambda))^{N-1}$. Running some experiments, we observed different results which invalidate this model. Fig. 9 represents the success probability of this attack with respect to the number of packets, theoretically and experimentally. Since we already know that the distribution of the rank is the Pólya distribution, we obtain

$$\Pr[R_{15} = 0] = (1 - p_{15})^{r_{15}}$$

The difference between these two curves are a result of the dependency between the biases. In all our analysis, we assumed that the biases are independent, which may not be the case for some cases in practice. This difference can be observed in Fig. 9.

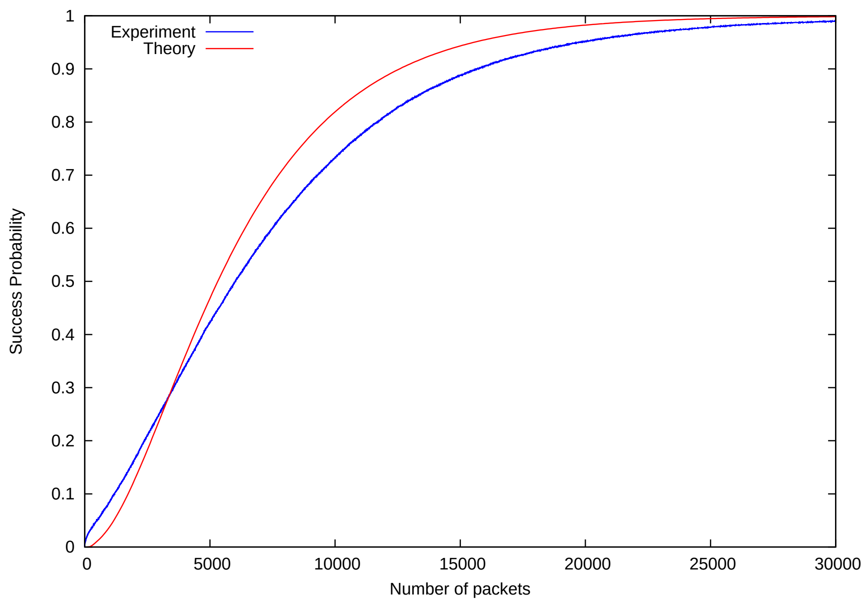

Fig. 9. The success probability of recovering $\bar{K}[15]$ as the top element in the voted list in theory and practice.

9 Experimental Attack & Comparison with Aircrack-ng

Fig. 10 represents a comparison between Aircrack-ng and our new attack. We used an Intel Xeon Processor W5590 at 3.33Ghz with 8M Cache for the comparison. For the attack on WEP and WPA, we used the biases up to $\bar{K}[34]$. For any $i > 34$, the probabilities are getting very close to the uniform distribution. It can still improve the overall success rate of the attack, but this improvement is not significant and it further increases the computational cost of the attack. The IVs are picked pseudo-randomly using SNOW 2.0 stream cipher [12].

In the previous section, we computed the success probability of recovering $\bar{K}[15]$ and drew the curve for when $\bar{K}[15]$ is the top element in the sorted list. But for a comparison with Aircrack-ng, we let the attack run for maximum 5 seconds. If the key is not found in that time period, we assume that the attack fails. If we do not restrict the attack time frame, it runs for ever by going exhaustively over all elements in the sorted lists.

As can be observed, our passive attack even outperforms Aircrack-ng running in active mode. This gives significant advantage to the attacker, since for some network cards, the driver has to be patched so that the network card can inject packets, and in some cases such a patch is not available at all. Moreover, the active attacks are detectable by intrusion detection systems. Similarly, passive attacks can be performed from a large distance. Moreover, the TCP/IPv4 packets can be captured with much higher rate than ARP packets. As a rule of thumb, in a high traffic network, (for instance the user is downloading a movie), if we consider TCP/IPv4 packets with maximum size around 1500 bytes, in a 20 Mbit/sec wireless network, it takes almost 10 seconds to capture 22 500 packets. This amount is already enough to find a key with our improved Aircrack-ng in less than 5 seconds.

WEP key recovery process is harder in practice than in theory. This is because the biases in RC4 are not independent, and several bytes of the keystream are unknown in ARP and TCP/IP packets. Therefore, the theoretical analysis is more complex if the dependencies are considered. Also, some bytes of the keystream have to be guessed, and the proportion of TCP/IP packets to ARP packets varies for every network and attack (passive vs. active). The a priori probability of guessing those bytes correctly can not be precisely determined, and we had to leverage some heuristics to deal with this problem; since this proportion also depends on the traffic itself, finding the ρ which is optimized for every network is not feasible. We leveraged some heuristics to set the ρ to obtain a high success rate in practice. Moreover, Aircrack-ng is not an interactive software. The interaction with the user may allow to tweak the ρ and/or wait for more packets to come. This trade-off should also be considered in real life applications.

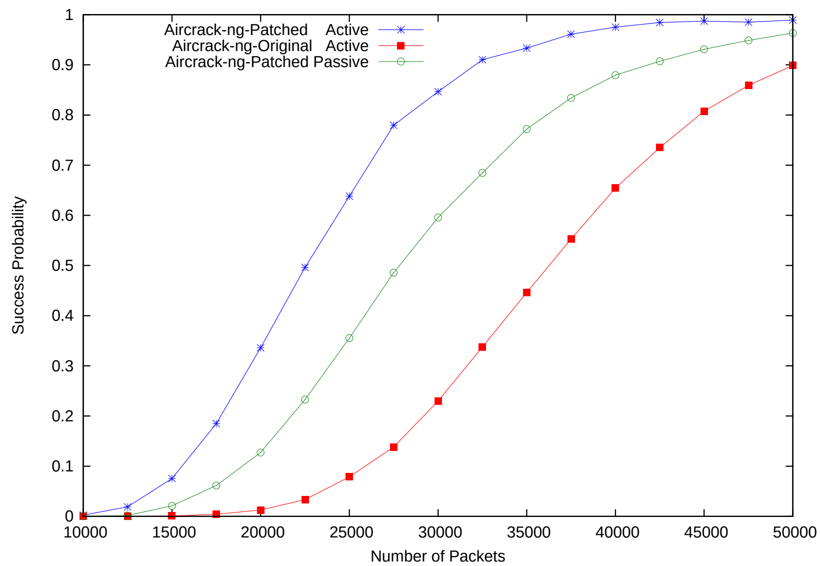

Fig. 10. Our attacks success probability (both active and passive attacks) with respect to the number of packets compared to Aircrack-ng in active attack mode.

The Algorithm described in Section 8 is recursive. This recursion is very expensive in practice, since with a wrong guess on a key byte, all the subsequent key bytes with higher indices are recovered incorrectly (in theory), so we need to recompute the vote for each of them again. In practice, we observed that a wrong guess of a key byte *does not* influence the recovery of subsequent key bytes significantly. For instance, even with a wrong guess on $\bar{K}[3]$, in many cases, we could still recover all the subsequent bytes correctly. This is because a wrong guess for $\bar{K}[3]$ mandates only 16 wrong swaps out of 256 iterations of the KSA. A further improvement to our work can be to adjust our theory to consider such cases. Hence, in our implementation, we perform a recursive attack to only find the best key candidate, and if it turns out to be a wrong key, we then use the pre-computed voted list to perform an exhaustive search, with no re-voting.

A Sequential Distinguisher Approach. Previously, we assumed that a *fixed* number of packets is given to the adversary and his goal was to maximize the success probability. Changing the perspective, one can look at the problem as fixing the success probability and searching for the minimum average number of packets to reach that probability. This idea was initially used by Davies and Murphy [47] to decrease the complexity of their attack against DES. With this type of model in mind, the notion of $n_{\max}$*-limited generic sequential non-adaptive distinguisher* was defined by Junod in [29], where $n_{\max}$ is an upper bound for the allowed number of packets in that context.

We can also use the notion of sequential distinguishers for RC4 key recovery. Mapping the definition of an $n_{\max}$-limited generic sequential non-adaptive distinguisher in [29] to our attack, the new attack works as follows: The attacker eavesdrops a small number of packets from the channel and then runs an attack similar to the one described in the previous section. If it fails, then he waits for more packets to come and runs the attack again. This procedure is iterated again and again. The attacker stops when he finds the correct key or the threshold $n_{\max}$ number of packets is reached. If the former occurs, it outputs 1 (success), otherwise it outputs 0 (failure). This attack mode was already used in Aircrack-ng and also in [4]. It is referred to as the *"interactive mode"*. This approach turns out to be more efficient in terms of the average number of packets compared to the other types of distinguishers. In fact, Siegmund [67] has proved the following theorem (see [29] for details).

Theorem 11. *For a simple hypothesis testing against a simple alternative with independent, identically distributed observations, a sequential probability ratio test is optimal in the sense of minimizing the expected number of samples among all tests having no larger error probabilities.*

Using this technique, we can decrease the average number of packets to reach the success probability of 50%. For instance, we can drop the data complexity of our fastest attack (i.e., with all $\rho_i = 1$) in Fig. 8 from 27 500 to 22 500 packets in average using this approach to gain the success probability of 50%. We also give another example next to illustrate how the number of packets can be dropped using this technique.

As an example, using 23 000 packets and the attack from the previous section, we computed the almost optimized ρ_i's derived from the Genetic algorithm approach in practice to gain the success probability of 50%. We set

$$\begin{array}{lllll} \rho_3 = 2 & \rho_4 = 1 & \rho_5 = 1 & \rho_6 = 2 & \rho_7 = 2 \\ \rho_8 = 1 & \rho_9 = 2 & \rho_{10} = 1 & \rho_{11} = 1 & \rho_{12} = 4 \\ \rho_{13} = 2 & \rho_{14} = 86 & \rho_{15} = 1 & & \end{array}$$

Next, we run the attack in the interactive mode with the above ρ_i's for a lot of WEP keys and find the minimal value of $n_{\max}$ which yields 50% success rate. Our experiments showed that $n_{\max} = 22\,000$. Consequently, We run the same attack in the interactive mode with $n_{\max} = 22\,000$ for recovering different WEP keys K_i leading to some n_i to succeed. Then, we compute the statistical average of the number of packets n_i when it succeeds and $n_{\max}$ for the attacks which fail. The average number of packets we obtained in practice was 19 800 packets, which is much less than fixing the number of packets and maximizing the success probability.

An open problem is to analyze the theoretical complexity of the sequential distinguisher approach described above and compare it with the experimental results. We leave this to future work.

10 Conclusion

We deployed a framework to manipulate pools of biases for RC4 which can be used to break the WPA protocol. In the case of the 8 weak bits of the TK, we have shown a simple distinguisher and a partial key recovery attack working with 2^{42} packets and a practical complexity. This can be used to improve the attack by Moen-Raddum-Hole [45] to mount a full temporary key recovery attack of complexity 2^{96} using 2^{42} packets. So far, this is the best temporal key recovery attack against WPA. In a future work, we plan to study further key recovery attacks to recover more pieces of the TK with a complexity lower than 2^{96}.

We have shown that conditional biases are not very helpful for breaking WPA, but they are really useful against WEP. For WEP, we recover the secret key with the success rate of 50% by using 22 500 packets in a few seconds.

The attack is still feasible with less packets, but runs for a longer period.

References

1. T. AlFardan, D.J. Bernstein, K.G. Paterson, B. Poettering, and J.C.N. Schuldt. On the Security of RC4 in TLS. In *USENIX Security Symposium*. USENIX Association, 2013.
2. F.J. Anscombe. Sampling theory of the negative binomial and logarithmic series distributions. *Biometrika*, 37(3-4):358–382, 1950.
3. M. Beck. Enhanced TKIP Michael Attacks, 2010. http://download.aircrack-ng.org/wiki-files/doc/enhanced_tkip_michael.pdf.
4. M. Beck and E. Tews. Practical Attacks Against WEP and WPA. In *WISEC*, pages 79–86. ACM, 2009.
5. E. Biham and Y. Carmeli. Efficient Reconstruction of RC4 Keys from Internal States. In *FSE*, volume 5086, pages 270–288. Springer, 2008.
6. A. Bittau. Additional Weak IV Classes for the FMS Attack, 2003. http://www.netstumbler.org/showthread.php?postid=89036#pos%t89036.
7. C.I. Bliss and R.A. Fisher. Fitting the Negative Binomial Distribution to Biological Data. *Biometrika*, 9:176–200, 1953.
8. D. Bongard. Offline bruteforce attack on WiFi Protected Setup. PasswordsCon, 2014. https://passwordscon.org/wp-content/uploads/2014/08/Dominique_Bongard.pdf.

9. R.B. Davies. Numerical inversion of a characteristic function. *Biometrika*, 60(2):415–417, 1973.
10. R.B. Davies. The distribution of a linear combination of chi-squared random variables. *Applied Statistics*, 29:323–333, 1980.
11. C. Devine and T. Otreppe. Aircrack-ng, accessed October 22, 2011. http://www.aircrack-ng.org/.
12. P. Ekdahl and T. Johansson. A New Version of the Stream cipher SNOW. In *SAC*, volume 2595, pages 47–61. Springer, 2002.
13. W. Feller. On a general class of "contagious" distributions. *Ann. Math. Stat.*, 14:389–400, 1943.
14. N. Ferguson. fel: an Improved MIC for 802.11 WEP. IEEE doc. 802.11-2/020r0, 2002.
15. S.R. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In *SAC*, volume 2259, pages 1–24. Springer, 2001.
16. S.R. Fluhrer and D.A. McGrew. Statistical Analysis of the Alleged RC4 Keystream Generator. In *FSE*, volume 1978, pages 19–30. Springer, 2001.
17. J. Gil-Pelaez. Note on the inversion theorem. *Biometrika*, 38(3/4):481–482, 1951.
18. J.Dj. Golic. Linear Statistical Weakness of Alleged RC4 Keystream Generator. In *EUROCRYPT*, volume 1233, pages 226–238. Springer, 1997.
19. J.Dj. Golic. Iterative Probabilistic Cryptanalysis of RC4 Keystream Generator. In *ACISP*, volume 1841, pages 220–223. Springer, 2000.
20. R. Housley, D. Whiting, and N. Ferguson. Alternate Temporal Key Hash. IEEE doc. 802.11-02/282r2, 2002.
21. D. Hulton. Practical Exploitation of RC4 Weaknesses in WEP Environments, 2001. http://www.dartmouth.edu/ madory/RC4/wepexp.txt.
22. IEEE. IEEE Std 802.11, Standards for Local and Metropolitan Area Networks: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
23. IEEE. 802.1x: Standards for Local and Metropolitan Area Networks: Port-Based Access Control, 2001. Draft 3.
24. IEEE. ANSI/IEEE standard 802.11i, Amendment 6 Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications, 2003. Draft 3.
25. IEEE. IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.
26. T. Isobe, T. Ohigashi, Y. Watanabe, and M. Morii. Full Plaintext Recovery Attack on Broadcast RC4. In *FSE*. Springer, 2013.
27. R. Ito and A. Miyaji. New Linear Correlations related to State Information of RC4 PRGA using IV in WPA. In *FSE*, volume 9054, pages 557–576, 2015.
28. R. Jenkins. ISAAC and RC4, 1996. http://burtleburtle.net/bob/rand/isaac.html.
29. P. Junod. On the Optimality of Linear, Differential, and Sequential Distinguishers. In *EUROCRYPT*, volume 2656, pages 255–271. Springer, 2003.
30. P. Junod and S. Vaudenay. Optimal Key Ranking Procedures in a Statistical Cryptanalysis. In *FSE*, volume 2656, pages 235–246. Springer, 2003.
31. A. Klein. Attacks on the RC4 Stream Cipher. *Design, Codes, and Cryptography*, 48:269–286, 2008.
32. L.R. Knudsen, W. Meier, B. Preneel, V. Rijmen, and S. Verdoolaege. Analysis Methods for (Alleged) RC4. In *ASIACRYPT*, volume 1514, pages 327–341. Springer, 1998.
33. Korek. chopchop (experimental WEP attacks). http: //www.netstumbler.org/showthread.php?t=12489.
34. Korek. Need Security Pointers, 2004. http://www.netstumbler.org/showthread.php?postid=89036#pos%t89036.
35. Korek. Next Generation of WEP Attacks?, 2004. http://www.netstumbler.org/showpost.php?p=93942&postcount=%35.
36. S. Maitra and G. Paul. New Form of Permutation Bias and Secret Key Leakage in Keystream Bytes of RC4. In *FSE*, volume 5086, pages 253–269. Springer, 2008.
37. S. Maitra, G. Paul, S. Sarkar, M. Lehmann, and W. Meier. New Results on Generalization of Roos-Type Biases and Related Keystreams of RC4. In *AFRICACRYPT*, volume 7918. Springer, 2013.
38. S. Maitra, G. Paul, and S. Sen Gupta. Attack on Broadcast RC4 Revisited. In *FSE*, volume 6733, pages 199–217. Springer, 2011.
39. I. Mantin. Analysis of the Stream Cipher RC4. Master's thesis, Weizmann Institute of Science, 2001.
40. I. Mantin. Predicting and Distinguishing Attacks on RC4 Keystream Generator. In *EUROCRYPT*, volume 3494, pages 491–506. Springer, 2005.
41. I. Mantin and A. Shamir. A Practical Attack on Broadcast RC4. In *FSE*, volume 2355, pages 152–164. Springer, 2001.
42. A. Maximov. Two Linear Distinguishing Attacks on VMPC and RC4A and Weakness. In *FSE*, volume 3557, pages 342–358. Springer, 2005.
43. A. Maximov and D. Khovratovich. New State Recovery Attack on RC4. In *CRYPTO*, volume 5157, pages 297–316. Springer, 2008.
44. I. Mironov. Not So Random Shuffles of RC4. In *CRYPTO*, volume 2442, pages 304–319. Springer, 2002.
45. V. Moen, H. Raddum, and K.J. Hole. Weaknesses in the Temporal Key Hash of WPA. *Mobile Computing and Communications Review*, 8:76–83, 2004.
46. R. Muirhead. *Aspects of Multivariate Statistical Theory*. Wiley, 2005.
47. S. Murphy and D. Davies. Pairs and triples of DES S-boxes. *Journal of Cryptology*, 8:1–25, 1995.

48. AirTight Networks. WPA2 Hole196 Vulnerability: Exploits and Remediation Strategies, 2012. http://www.airtightnetworks.com/fileadmin/pdf/whitepaper/WPA2-Hole196-Vulnerability.pdf.
49. J. Neyman. On a new class of "contagious" distributions, applicable in entomology and bacteriology. *Ann. Math. Stat.*, 10:35–57, 1939.
50. J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, second edition, 2006.
51. P. Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. In *CRYPTO*, volume 2729, pages 617–630. Springer, 2003.
52. T. Ohigashi, T. Isobe, Y. Watanabe, and M. Morii. How to Recover Any Byte of Plaintext on RC4. In *SAC*. Springer, 2013.
53. T. Ohigashi and M. Morii. A practical message falsification attack on WPA. In *JWIS*, pages 5A–4. CDROM, 2009.
54. K. Paterson, B. Poettering, and J. Schuldt. Plaintext Recovery Attacks Against WPA/TKIP. In *FSE*, volume 8540, pages 325–349, 2014.
55. G. Paul and S. Maitra. Permutation After RC4 Key Scheduling Reveals the Secret. In *SAC*, volume 4876, pages 360–377. Springer, 2007.
56. G. Paul, S. Rathi, and S. Maitra. On Non-Negligible Bias of the First Output Byte of RC4 towards the First Three Bytes of the Secret Key. *Design, Codes, and Cryptography*, 49:123–134, 2008.
57. S. Paul and B. Preneel. A New Weakness in the RC4 Keystream Generator and an Approach. In *FSE*, volume 3017, pages 245–259. Springer, 2004.
58. J. Postel and J. Reynolds. A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, 1988. http://www.cs.berkeley.edu/∼daw/my-posts/my-rc4-weak-keys.
59. A. Roos. A Class of Weak Keys in RC4 Stream Cipher (sci.crypt), 1995. http://marcel.wanda.ch/Archive/WeakKeys.
60. S. Sen Gupta, S. Maitra, and W. Meier. Distinguishing WPA. In *Cryptology ePrint Archive*, 2013. http://eprint.iacr.org/2013/476.pdf.
61. S. Sen Gupta, S. Maitra, W. Meier, G. Paul, and S. Sarkar. Dependence in IV-Related Bytes of RC4 Key Enhances Vulnerabilities in WPA. In *FSE*, volume 8540, pages 350–369, 2014.
62. S. Sen Gupta, S. Maitra, G. Paul, and S. Sarkar. Proof of Empirical RC4 Biases and New Key Correlations. In *SAC*, volume 7118, pages 151–168. Springer, 2011.
63. S. Sen Gupta, S. Maitra, G. Paul, and S. Sarkar. (Non-)Random Sequences from (Non-)Random Permutations - Analysis of RC4 stream cipher. *Journal of Cryptology*, 2014.
64. P. Sepehrdad, P. Sušil, S. Vaudenay, and M. Vuagnoux. Smashing WEP in a Passive Attack. In *FSE*. Springer, 2013.
65. P. Sepehrdad, S. Vaudenay, and M. Vuagnoux. Discovery and Exploitation of New Biases in RC4. In *SAC*, volume 6544, pages 74–91. Springer, 2010.
66. P. Sepehrdad, S. Vaudenay, and M. Vuagnoux. Statistical Attack on RC4: Distinguishing WPA. In *EUROCRYPT*, volume 6632, pages 343–363. Springer, 2011.
67. D. Siegmund. *Sequential analysis - tests and confidence intervals*. Springer, 1985.
68. A. Stubblefield, J. Ioannidis, and A.D. Rubin. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. *Network and Distributed System Security Symposium (NDSS)*, 2002.
69. A. Stubblefield, J. Ioannidis, and A.D. Rubin. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). *ACM Transactions on Information and System Security (TISSEC)*, 7(2), 2004.
70. Student. On the error of counting with a haemocytometer. *Biometrika*, 5:351–360, 1907.
71. E. Tews. Attacks on the WEP Protocol. In *Cryptology ePrint Archive*, 2007. http://eprint.iacr.org/2007/471.pdf.
72. E. Tews, R. Weinmann, and A. Pyshkin. Breaking 104 Bit WEP in Less Than 60 Seconds. In *WISA*, volume 4867, pages 188–202. Springer, 2007.
73. H.C.S. Thom. The Frequency of Hail Occurrence. *Theoretical and Applied Climatology*, 8:185–194, 1957.
74. H.C.S. Thom. Tornado Probabilities. *American Meteorological Society*, pages 730–736, 1963.
75. Y. Todo, Y. Ozawa, T. Ohigashi, and M. Morii. Falsification Attacks against WPA-TKIP in a Realistic Environment. *IEICE Transactions*, 95-D(2):588–595, 2012.
76. V. Tomasevic, S. Bojanic, and O. Nieto-Taladriz. Finding an Internal State of RC4 Stream Cipher. *Information Sciences: an International Journal*, 177:1715–1727, 2007.
77. M. Vanhoef and F. Piessens. Practical Verification of WPA-TKIP Vulnerabilities. In *ASIACSS*, pages 427–436, 2013.
78. M. Vanhoef and F. Piessens. All Your Biases Belong to Us: Breaking RC4 in WPA-TKIP and TLS. In *USENIX Security*, pages 97–112, 2015.
79. S. Vaudenay and M. Vuagnoux. Passive-only Key Recovery Attacks on RC4. In *SAC*, volume 4876, pages 344–359. Springer, 2007.
80. S. Viehböck. Brute forcing Wi-Fi Protected Setup: When poor design meets poor implementation, 2011. http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.
81. D. Wagner. Weak Keys in Rc4 (sci.crypt), 1995. http://www.cs.berkeley.edu/∼daw/my-posts/my-rc4-weak-keys.
82. L. Whitaker. On the Poisson law of small numbers. *Biometrika*, 10:36–71, 1914.

A Proof of Lemmas

In this section, we provide the proof of Lemma 4, Lemma 6, Lemma 8, and Lemma 10.

A.1 Proof of Lemma 4

Proof. For $x \neq 0$, we have

$$\begin{aligned}
\Pr[C-A=x] &= \sum_{y} \Pr[B-A=y] \,.\, \Pr[C-B=x-y] \\
&= \sum_{\substack{y\neq 0 \\ y\neq x}} \Pr[B-A=y] \,.\, \Pr[C-B=x-y] + \Pr[A=B] \,.\, \Pr[C-B=x] + \Pr[B-A=x] \,.\, \Pr[B=C] \\
&= (N-2)\left(\tfrac{1-p_1}{N-1}\right)\left(\tfrac{1-p_2}{N-1}\right) + p_1\left(\tfrac{1-p_2}{N-1}\right) + p_2\left(\tfrac{1-p_1}{N-1}\right)
\end{aligned}$$

which does not depend on x. Then,

$$\Pr[A=C] = 1 - \sum_{x\neq 0} \Pr[C-A=x] = \frac{1}{N} + \left(\frac{N}{N-1}\right)\left(p_1 - \frac{1}{N}\right)\left(p_2 - \frac{1}{N}\right)$$

So, $A \stackrel{p}{=} C$.

The $\otimes$ operation is trivially commutative over $[0,1]$ and 1 is the neutral element. Below, we show that it is also associative over $[0,1]$. We simply show that $(p_1 \otimes p_2) \otimes p_3 = p_1 \otimes (p_2 \otimes p_3)$.

$$\begin{aligned}
(p_1 \otimes p_2) \otimes p_3 &= \tfrac{1}{N} + \left(\tfrac{N}{N-1}\right) \cdot \left[\tfrac{1}{N} + \left(\tfrac{N}{N-1}\right)\left(p_1 - \tfrac{1}{N}\right)\left(p_2 - \tfrac{1}{N}\right) - \tfrac{1}{N}\right] \cdot \left(p_3 - \tfrac{1}{N}\right) \\
&= \tfrac{1}{N} + \left(\tfrac{N}{N-1}\right)^2 \cdot \left(p_1 - \tfrac{1}{N}\right)\left(p_2 - \tfrac{1}{N}\right)\left(p_3 - \tfrac{1}{N}\right)
\end{aligned}$$

and

$$\begin{aligned}
p_1 \otimes (p_2 \otimes p_3) &= \tfrac{1}{N} + \left(\tfrac{N}{N-1}\right)\left(p_1 - \tfrac{1}{N}\right) \cdot \left[\tfrac{1}{N} + \left(\tfrac{N}{N-1}\right)\left(p_2 - \tfrac{1}{N}\right)\left(p_3 - \tfrac{1}{N}\right) - \tfrac{1}{N}\right] \\
&= \tfrac{1}{N} + \left(\tfrac{N}{N-1}\right)^2 \cdot \left(p_1 - \tfrac{1}{N}\right)\left(p_2 - \tfrac{1}{N}\right)\left(p_3 - \tfrac{1}{N}\right)
\end{aligned}$$

Hence, the $\otimes$ operator is associative.

□

A.2 Proof of Lemma 6

Proof. We have

$$\begin{aligned}
\Pr[A=S[D]=S[E]|\mathsf{Cond}] &= \left(\tfrac{1}{\Pr[\mathsf{Cond}]}\right) \cdot \Pr[(A=S[D]) \wedge \mathsf{Cond}, D=E] \\
&= \left(\tfrac{1}{\Pr[\mathsf{Cond}']}\right) \cdot \Pr[(A=S[D]) \wedge \mathsf{Cond}', D=E] \\
&= \left(\tfrac{1}{\Pr[\mathsf{Cond}']}\right) \sum_{\substack{\alpha,\beta,\gamma,\delta \\ \alpha+\beta+\gamma=0 \\ \delta=0}} \Pr[A-B=\alpha, B-C=\beta, (C-S[D]=\gamma) \wedge \mathsf{Cond}', D-E=\delta] \\
&= \sum_{\substack{\alpha,\beta,\gamma,\delta \\ \alpha+\beta+\gamma=0 \\ \delta=0}} \Pr[A-B=\alpha] \cdot \Pr[B-C=\beta] \cdot \Pr[C-S[D]=\gamma|\mathsf{Cond}'] \cdot \Pr[D-E=\delta] \\
&= (p_1 \otimes p_2 \otimes p_3) \cdot p_4
\end{aligned}$$

We also have,

$$\begin{aligned}
\mathsf{Pr}[A \neq S[D], D \neq E | \mathsf{Cond}] &= 1 - \mathsf{Pr}[A = S[D] | \mathsf{Cond}] - \mathsf{Pr}[D = E | \mathsf{Cond}] + \mathsf{Pr}[A = S[D], D = E | \mathsf{Cond}] \\
&= 1 - \mathsf{Pr}[A = S[D] | \mathsf{Cond}'] - \mathsf{Pr}[D = E | \mathsf{Cond}'] + \mathsf{Pr}[A = S[D], D = E | \mathsf{Cond}'] \\
&= \mathsf{Pr}[A \neq S[D], D \neq E | \mathsf{Cond}']
\end{aligned}$$

Moreover,

$$\begin{aligned}
\mathsf{Pr}[A = S[E], A \neq S[D] | \mathsf{Cond}] &= \mathsf{Pr}[A = S[E], A \neq S[D], D \neq E | \mathsf{Cond}] \\
&= \mathsf{Pr}[A = S[E] | A \neq S[D], D \neq E, \mathsf{Cond}] \cdot \mathsf{Pr}[A \neq S[D], D \neq E | \mathsf{Cond}] \\
&= \left(\tfrac{1}{N-1}\right) \cdot \mathsf{Pr}[A \neq S[D], D \neq E | \mathsf{Cond}'] \\
&= \left(\frac{1}{(N-1)\cdot \mathsf{Pr}[\mathsf{Cond}']}\right) \sum_{\substack{\alpha,\beta,\gamma,\delta \\ \alpha+\beta+\gamma\neq 0 \\ \delta \neq 0}} \mathsf{Pr}[A - B = \alpha, B - C = \beta, (C - S[D] = \gamma) \wedge \mathsf{Cond}', D - E = \delta] \\
&= \left(\tfrac{1}{N-1}\right) \sum_{\substack{\alpha,\beta,\gamma,\delta \\ \alpha+\beta+\gamma\neq 0 \\ \delta \neq 0}} \mathsf{Pr}[A - B = \alpha] \cdot \mathsf{Pr}[B - C = \beta] \cdot \mathsf{Pr}[C - S[D] = \gamma | \mathsf{Cond}'] \cdot \mathsf{Pr}[D - E = \delta] \\
&= \left(\tfrac{1}{N-1}\right) \cdot (1 - p_1 \otimes p_2 \otimes p_3) \cdot (1 - p_4)
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathsf{Pr}[A = S[E] | \mathsf{Cond}] &= (p_1 \otimes p_2 \otimes p_3) \cdot p_4 + \left(\tfrac{1}{N-1}\right) \cdot (1 - p_1 \otimes p_2 \otimes p_3) \cdot (1 - p_4) \\
&= p_1 \otimes p_2 \otimes p_3 \otimes p_4
\end{aligned}$$

□

A.3 Proof of Lemma 8

Proof. Note that $S_{i-1}[m_j] = S_t[m_j]$ is equivalent to $S_{i-1}[m_j] = \cdots = S_{t+1}[m_j] = S_t[m_j]$, because if m_j is moved, it can not come back to the same position, due to the restrictions $m_j \leq t$ or $m_j > i-1$. Furthermore, $P_A^b(i,t)$ is defined as the probability that a set of bytes corresponding to a set of indices $(m_1, \ldots, m_b)$ are not swapped from S_t to S_{i-1}. Since $m_j \leq t$ or $m_j > i-1$, they will not be selected by the index i from S_t to S_{i-1}. Hence, they can only be picked by the index j which moves uniformly at random by the definition of RC4$^\star(t)$. So, this is correct with probability $\left(\frac{N-b}{N}\right)^{i-t-1}$. In fact, we have

$$S_{i-1}[m_j] \overset{P_A^1}{=} S_t[m_j]$$

That is because

$$\Pr_{x \neq y}[S_{i-1}[m_j] = y | S_t[m_j] = x] = \frac{1}{N-1}$$

Since we know up to state S_t, we have to estimate $\sum_{x=1}^{i} S_{x-1}[x]$ with the state bytes in S_t. The first term in $P_B(i,t)$ is the probability that $S_{x-1}[x]$ can be approximated as $S_t[x]$ for $x > t+1$. The second term is the probability that at least one of these approximations is wrong, but the result holds with uniform probability. We can also assume that

$$\Pr_{y \neq \sigma_i(t)}\left[\sum_{x=1}^{i} S_{x-1}[x] = y\right] = \frac{1}{N-1}$$

P_0 is the probability that index i is not swapped from S_i to S'_{i-1}. This probability depends only on the values of j and j', which change uniformly at random in RC4$^\star(t)$. There are $N-2$ state updates, so the overall probability is $\left(\frac{N-1}{N}\right)^{N-2}$. We also have

$$\Pr_{x\neq y}[S'_{i-1}[i]=y|S_i[i]=x]=\frac{1}{N-1}$$

This leads to $S'_{i-1}[i]\overset{P_0}{=}S_i[i]$.

$\square$

A.4 Proof of Lemma 10

Proof. First, we show that if $S_{N-1}[2]=0$ and $S_{N-1}[1]\neq 2$, we obtain $z_2=0$. Assume $S'_0[1]=\alpha$ and $S'_0[\alpha]=\beta$, then $i=1$ and $j'_1=S'_0[1]=\alpha$, so we swap $S'_0[1]$ and $S'_0[\alpha]$. In the next iteration, $i=2$ and $j'_2=\alpha+S'_1[2]=\alpha$, that is because we assumed $S_{N-1}[1]\neq 2$ and $S_{N-1}[2]=0$, so $S'_1[2]=0$. Then, we swap $S'_1[2]$ and $S'_1[\alpha]$ and z_2 is computed as $z_2=S'_2[S'_2[2]+S'_2[\alpha]]=S'_2[\alpha]=0$. Finally,

$$\begin{aligned}
\Pr[z_2=0] &= \Pr[z_2=0|S'_0[2]=0,S'_0[1]\neq 2]\ .\ \Pr[S'_0[2]=0,S'_0[1]\neq 2]\\
&\quad+\Pr[z_2=0|S'_0[2]\neq 0\vee S'_0[1]=2]\ .\ \Pr[S'_0[2]\neq 0\vee S'_0[1]=2]\\
&=\frac{1}{N}\left(\frac{N-1}{N}\right)+\frac{1}{N}\left[\left(\frac{N-1}{N}\right)+\frac{1}{N}-\frac{1}{N}\left(\frac{N-1}{N}\right)\right]\\
&=\frac{1}{N}\left(\frac{N-1}{N}\right)\left(2-\frac{1}{N}\right)+\frac{1}{N^2}\\
&\approx\frac{2}{N}
\end{aligned}$$

If $x\neq 0$, we also have

$$\Pr[z_2=x]=\frac{1-\Pr[z_2=0]}{N-1}=\frac{N-2}{N(N-1)}$$

$\square$

B Some Numerical Values of RC4 Correlations

reference	i	t	$g(i,t)$	$P(i,t)$
Klein − Improved	3	2	1.000000	$1.37/N$
A_u15	3	2	0.000030	$48.2/N$
A_s13	3	2	0.000015	$36.3/N$
A_u13_1	3	2	0.000019	$36.3/N$
A_u13_2	3	2	0.000019	$35.9/N$
A_u13_3	3	2	0.000015	$35.9/N$
A_s5_1	3	2	0.000044	$14.07/N$
A_s5_2	3	2	0.000015	$14.07/N$
A_s5_3	3	2	0.000021	$14.07/N$
A_u5_1	3	2	0.000035	$14.07/N$
A_u5_2	3	2	0.000020	$26.8/N$
A_u5_3	3	2	0.000033	$9.7/N$
A_s3	3	2	0.000015	$5.8/N$
A_4_s13	4	3	0.000061	$18.4/N$
A_4_u5_1	4	3	0.000029	$14.2/N$
A_4_u5_2	4	3	0.000021	$14.2/N$
A_neg_1	3	2	0.000004	0
A_neg_2	3	2	0.001443	0
A_neg_3	3	2	0.000547	0
A_neg_4	3	2	0.000006	0
SVV_10	16	2	0.003650	$9.4/N$

C Classification of Biases

In this section, we classify the statistical correlations in RC4. We only report those which are exploitable against WEP and WPA. Most of the biases reported against RC4 in [65] are not exploitable, because they do not bind the secret key with the keystream. They often require extra bytes (state or keystream), which are unknown to the attacker. We elaborate each correlation individually and extract the probability that it holds in our model. The list includes the improved version of the Klein attack in [79] (elaborated in Section 6.1) and the improved version of 19 biases by Korek [35,34] (A_u15 was elaborated in Section 6.2) and the SVV_10, the improved bias of Sepehrdad, Vaudenay and Vuagnoux in [65]. All the probabilities are new. The path for each bias is described. Due to the similarity of several paths and for simplicity, in several cases we do not repeat the same formulas again. The reader should refer to Appendix D for the formulas to compute the corresponding probabilities.

Korek is the nickname of a hacker who discovered 20 key recovery attacks similar to the FMS attack [15]. Korek classified them into three categories. The first group of attacks uses only z_1 and the state of the array S_{i-1} (i.e., $K[0], K[1] \ldots, K[i-1]$) of the KSA to recover the secret key $K[i]$ (typically the FMS attack). The second class of attacks uses the second byte of the keystream z_2. Ultimately, the last one highlights the improbable secret key bytes. They are called *negative attacks* or *impossible attacks*. We only mention 19 such correlations, since the conditions of the attack A_u5_4 are rarely satisfied in practice except for $i = 6$ when $t = 2$, in which its corresponding success probability is very close to $1/N$.

C.1 The A_s5_1 Attack

- **Conditions:** $S_t[1] < t+1$, $S_t[1] + S_t[S_t[1]] = i$, $z_1 \neq \{S_t[1], S_t[S_t[1]]\}$ and $(S_t^{-1}[z_1] < t+1 \text{ or } S_t^{-1}[z_1] > i-1)$
- **Attack path:** (see Fig. 11)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = \alpha$
  - $S_t[\alpha] = \cdots = S_{i-1}[\alpha] = S_i[\alpha] = \cdots = S_{N-1}[\alpha] = \beta$
  - $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = \cdots = S_{N-1}[i]$
- **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[z_1] - \sigma_i(t)$
- **Probability of success:** $\mathsf{Kor}_2^3(i,t)$ (see Appendix D)

This attack is the generalization of the FMS attack. It works as follows: Let $S_t[1] = \alpha$, $S_t[\alpha] = \beta$ and also assume $\alpha + \beta = i$ by the conditions. Following the attack path, these two values are maintained at the same position through the entire KSA algorithm. The attack path also mandates that $S_i[i]$ is maintained until the state S_{N-1}. At the first iteration of the PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = \alpha$. Then, $S_{N-1}[1]$ and $S_{N-1}[\alpha]$ are swapped. Finally, we have $z_1 = S'_1[S'_1[1] + S'_1[\alpha]] = S'_1[\alpha+\beta] = S'_1[i] = S_i[i] = S_{i-1}[j_i] = S_t[j_i]$. Hence, we obtain $z_1 = S_t[j_i]$ and so $j_i = S_t^{-1}[z_1]$. Since we also have $\bar{K}[i] = j_i - \sigma_i(t)$, we conclude from the previous equation that $\bar{K}[i] = S_t^{-1}[z_1] - \sigma_i(t)$. The last condition on z_1 is to filter out some incorrect events leading to the same results. The condition $\{S_t[1], S_t^{-1}[z_1]\} < t+1$ is to make $\{\alpha, j_i\} < t+1$, so it is not trivially swapped during the KSA iterations. We also should make sure that $z_1 \neq \{\alpha, \beta\}$. Thus, we need the condition $z_1 \neq \{S_t[1], S_t[S_t[1]]\}$.

C.2 The A_s13 Attack

- **Conditions:** $S_t[1] = i$, $(S_t^{-1}[0] < t+1 \text{ or } S_t^{-1}[0] > i-1)$ and $z_1 = i$
- **Attack path:** (see Fig. 12)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = i$
  - $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = \cdots = S_{N-1}[i] = 0$
- **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[0] - \sigma_i(t)$
- **Probability of success:** $\mathsf{Kor}_1^2(i,t)$ (see Appendix D)

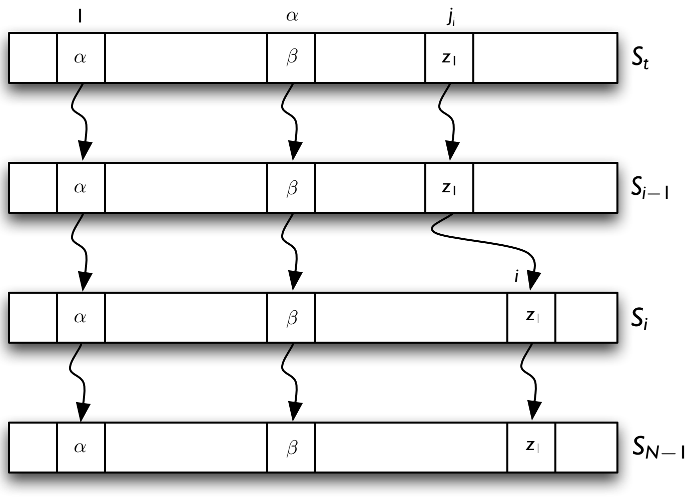

Fig. 11. RC4 state update in the A_s5_1 attack

In this attack, the PRGA automatically makes aure that $S'_1[1] = 0$. Let $S_{N-1}[i] = \gamma$. We show that $\gamma = 0$. At the first step of the PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = i$. So, we swap $S_{N-1}[1]$ and $S_{N-1}[i]$. To compute z_1, we have $z_1 = S'_1[S'_1[1] + S'_1[i]] = S'_1[\gamma + i] = i$, because from the conditions we have $z_1 = i$, leading to $\gamma = 0$. We already know that $S_{i-1}[j_i] = S_i[i]$ and following the attack path, we assume that $S_{i-1}[j_i] = S_t[j_i]$, and $S_{i-1}[j_i] = 0$. Thus, $S_t[j_i] = 0$. Finally, we obtain $j_i = S_t^{-1}[0]$. Using a similar approach as described in the previous attacks, we obtain $\bar{K}[i] = S_t^{-1}[0] - \sigma_i(t)$.

C.3 The A_u13_1 Attack

- **Conditions:** $S_t[1] = i$, ($S_t^{-1}[1-i] < t+1$ or $S_t^{-1}[1-i] > i-1$) and $z_1 = 1-i$
- **Attack path:** (see Fig. 13)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = i$
  - $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = \cdots = S_{N-1}[i] = 1-i$
- **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[z_1] - \sigma_i(t)$
- **Probability of success:** $\mathsf{Kor}_1^2(i,t)$ (see Appendix D)

At the first step of the PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = i$, so we swap $S_{N-1}[1]$ and $S_{N-1}[i]$. To compute z_1, we have $z_1 = S'_1[S'_1[1] + S'_1[i]] = S'_1[1] = 1-i$. We already know that $S_{i-1}[j_i] = S_i[i]$ and following the attack path we assume that $S_{i-1}[j_i] = S_t[j_i]$ and $S_{i-1}[j_i] = 1-i$. Thus, $S_t[j_i] = 1-i$. Finally, we obtain $j_i = S_t^{-1}[1-i]$. Using a smilar approach as described in the previous attacks, we obtain $\bar{K}[i] = S_t^{-1}[z_1] - \sigma_i(t)$.

C.4 The A_u5_1 Attack

- **Conditions:** $S_t[1] = i$, $S_t^{-1}[z_1] < t+1$, $S_t^{-1}[S_t^{-1}[z_1] - i] \neq 1$, ($S_t^{-1}[S_t^{-1}[z_1] - i] < t+1$ or $S_t^{-1}[S_t^{-1}[z_1] - i] > i-1$), $z_1 \neq \{i, 1-i, S_t^{-1}[z_1] - i\}$ and $S_t^{-1}[z_1] \neq 2i$
- **Attack path:** (see Fig. 14)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = i$

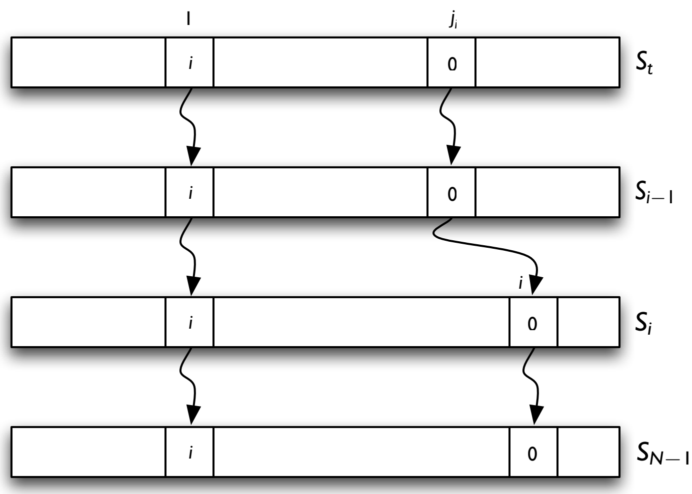

Fig. 12. RC4 state update in the A_s13 attack

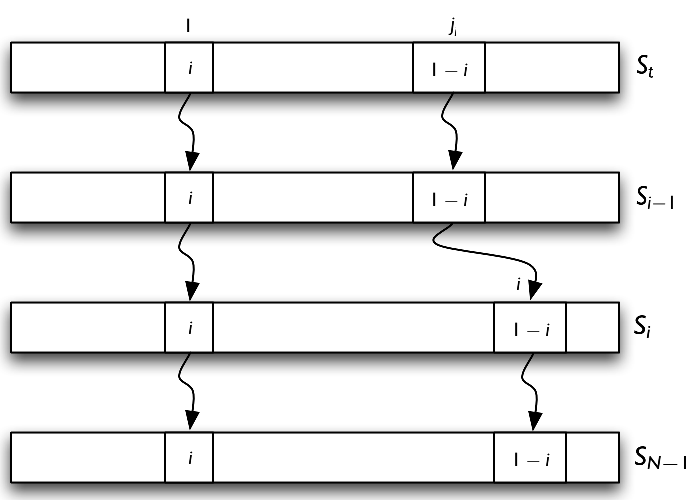

Fig. 13. RC4 state update in the A_u13_1 attack

- Assuming $S_t^{-1}[z_1] = \alpha$, we should have $S_t[\alpha] = \cdots = S_{i-1}[\alpha] = S_i[\alpha] = \cdots = S_{N-1}[\alpha] = z_1$.
- $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = \cdots = S_{N-1}[i] = S_t^{-1}[z_1] - i$

– **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[S_t^{-1}[z_1] - i] - \sigma_i(t)$

– **Probability of success:** $\mathsf{Kor}_2^3(i,t)$ (see Appendix D)

At the first stage of the PRGA, we have $i = 1$ and $j_1' = S_{N-1}[1] = i$. So we swap $S_{N-1}[1]$ and $S_{N-1}[i]$. We know that $S_i[i] = S_{i-1}[j_i] = S_t^{-1}[z_1] - i$ and $j_i = S_t^{-1}[S_t^{-1}[z_1] - i]$. We compute $z_1 = S_1'[S_1'[1] + S_1'[i]] = S_1'[S_t^{-1}[z_1] - i + i] = S_1'[S_1^{-1}[z_1]] = z_1$. Therefore, we have $\bar{K}[i] = S_t^{-1}[S_t^{-1}[z_1] - i] - \sigma_i(t)$. The condition $z_1 \neq \{i, 1-i\}$ is to filter out the attacks A_u13_1 and A_s13. $S_t^{-1}[z_1] < t+1$, because otherwise z_1 would be swapped in the next iterations of the KSA. If $S_t^{-1}[S_t^{-1}[z_1] - i] = 1$, then $j_i = 1$ and so $S_{i-1}[1]$ and $S_{i-1}[i]$ will be swapped in the i-th steps of the KSA.

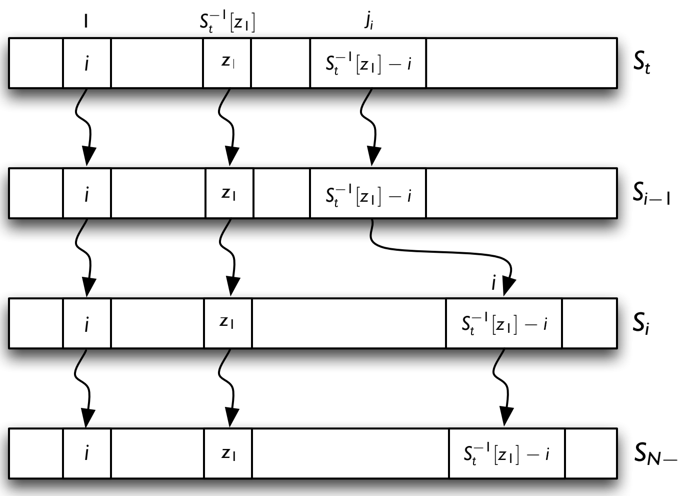

Fig. 14. RC4 state update in the A_u5_1 attack

C.5 The A_u5_2 Attack

– **Conditions:** $S_t[i] = 1$ and $z_1 = S_t[2]$

– **Attack path:** (see Fig. 15)

- $S_t[i] = \cdots = S_{i-1}[i] = S_i[1] = \cdots = S_{N-1}[1] = 1$
- $S_t[2] = \cdots = S_{N-1}[2] = z_1$
- $j_i = 1$

– **Key recovery relation:** $\bar{K}[i] = 1 - \sigma_i(t)$

– **Probability of success:** $P_u^2(i,t)$ (see Appendix D)

In this attack, we assume $j_i = 1$. Following the attack path, we know that $j_1' = S_{N-1}[1] = 1$. In the next PRGA iteration no swap is made since $S_{N-1}[1]$ and $S_{N-1}[1]$ are to be swapped. Hence, $z_1 = S_1'[S_1'[1] + S_1'[1]] = S_1'[2] = S_t[2]$. Finally, the key recovery equation becomes $\bar{K}[i] = 1 - \sigma_i(t)$.

We classify the conditions as

$$\mathsf{C}_1 : S_t[i] = 1 \quad \text{and} \quad \mathsf{C}_2 : z_1 = S_t[2]$$

We also classify the attack path assumptions and the key recovery equation as

$$\begin{cases} \mathsf{S}_1 : S_t[i] = \cdots = S_{i-1}[i] \\ \mathsf{S}_2 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_3 : S_t[2] = \cdots = S_{N-1}[2] \\ \mathsf{S}_4 : \bar{K}[i] = j_i - \sigma_i(t) \\ \mathsf{E}_1 : j_i = 1 \\ \mathsf{B} : \bar{K}[i] = 1 - \sigma_i(t) \end{cases}$$

Now, we compute the theoretical success probability of the attack. The goal is to estimate $\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2]$. Using a similar approach as A_u15, we end up with

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2] = \mathsf{Pr}(\mathsf{E}_1|\mathsf{C}) \; . \; \left(\frac{NP_B(i,t)-1}{N-1}\right) + \left(\frac{1-P_B(i,t)}{N-1}\right)$$

where

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] \approx \mathsf{Pr}(\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3|\mathsf{E}_1\mathsf{C}_2) + \frac{1}{N}\left(1 - P_A^2(i,t) \; . \; \left(\frac{N-2}{N}\right)^{N-i-1}\right)$$

$$\begin{aligned} \mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3|\mathsf{E}_1\mathsf{C}_2] &= \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{E}_1|\mathsf{C}_2]}{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_2]}\right) \\ &= \mathsf{Pr}[\mathsf{C}_2|\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{E}_1] \; . \; \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_2] \; . \; \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_2]}\right) \end{aligned}$$

Since $\mathsf{Pr}[\mathsf{C}_2]$ is not uniformly distributed, we use the following lemma to compute its value. Then, we approximate

$$\mathsf{Pr}[\mathsf{C}_2] \approx \left(\frac{N-1}{N}\right)^{t-2} . \; \mathsf{Pr}[z_1 = \bar{K}[2]+3]$$

Lemma 12. ***(Theorem** 3 **in [56])*** *For any arbitrary secret key, the correlation between the key bytes and the first byte of the keystream output is given by*

$$\mathsf{Pr}[z_1 = \bar{K}[2]+3] = \xi = \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^N \left(1 - \frac{1}{N} + \frac{1}{N^2}\right) + \frac{1}{N^2} + 1\right]$$

Deploying the above lemma, we obtain

$$\begin{aligned} \mathsf{Pr}[\mathsf{C}_1\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3|\mathsf{E}_1\mathsf{C}_2] &= \left(\frac{N}{N-1}\right)^{t-2} \cdot \frac{N}{\xi}\left(\frac{1}{N}\cdot\frac{1}{N}\left(\frac{N-2}{N}\right)^{N-1-i} . \; P_A^2(i,t)\right) \\ &= \frac{1}{N\xi}P_A^2(i,t)\left(\frac{N}{N-1}\right)^{t-2}\left(\frac{N-2}{N}\right)^{N-1-i} \end{aligned}$$

Therefore, overall we have

$$\begin{aligned} \mathsf{Pr}[\mathsf{B}|\mathsf{C}_1\mathsf{C}_2] = &\frac{1}{N}\left(\frac{NP_B(i,t)-1}{N-1}\right) \\ &\cdot \left[\frac{1}{\xi}P_A^2(i,t)\left(\frac{N}{N-1}\right)^{t-2}\left(\frac{N-2}{N}\right)^{N-1-i} + \left(1 - P_A^2(i,t)\left(\frac{N-2}{N}\right)^{N-i-1}\right)\right] \\ &+ \left(\frac{1-P_B(i,t)}{N-1}\right) \end{aligned}$$

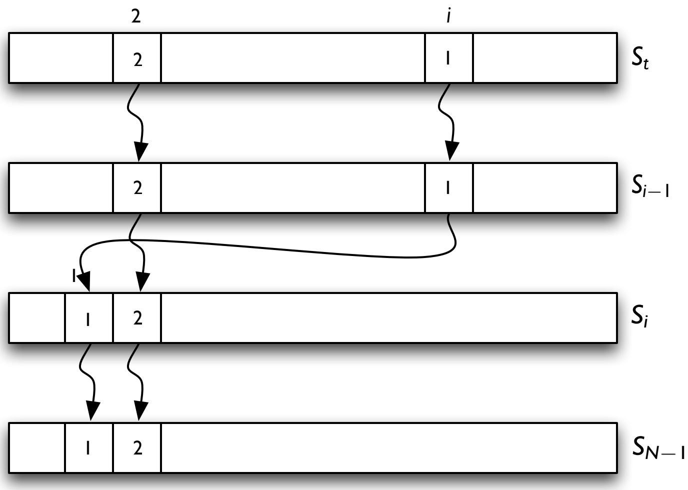

Fig. 15. RC4 state update in the A_u5_2 attack

C.6 The A_u13_2 Attack

- **Conditions:** $S_t[i] = i$, $S_t[1] = 0$ and $z_1 = i$
- **Attack path:** (see Fig. 16)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[i] = \cdots = S_{N-1}[i] = 0$
  - $S_t[i] = \cdots = S_{i-1}[i] = S_i[1] = \cdots = S_{N-1}[1] = i$
  - $j_i = 1$
- **Key recovery relation:** $\bar{K}[i] = 1 - \sigma_i(t)$
- **Probability of success:** $P_u^3(i,t)$ (see Appendix D)

This attack is very similar to the previous attack. Again, we assume $j_i = 1$. We know that $S_{i-1}[1] = 0$ and $S_{i-1}[i] = i$. At the i-th stage of the KSA state update, right after the swap, we have $S_i[1] = i$ and $S_i[i] = 0$. We assume these two values are maintained until the last iteration of the KSA. In the PRGA, initially $i = 1$ and $j'_1 = S_{N-1}[1] = i$. So, we swap $S_{N-1}[1]$ and $S_{N-1}[i]$. We then compute $z_1 = S'_1[S'_1[1] + S'_1[i]] = i$. Hence, the key recovery equation is $\bar{K}[i] = 1 - \sigma_i(t)$.

We classify the conditions as

$$\mathsf{C}_1 : S_t[i] = i \quad \text{and} \quad \mathsf{C}_2 : S_t[1] = 0 \quad \text{and} \quad \mathsf{C}_3 : z_1 = i$$

We also classify the attack path assumptions and the key recovery equation as

$$\begin{cases} \mathsf{S}_1 : S_t[i] = \cdots = S_{i-1}[i] \\ \mathsf{S}_2 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_3 : S_t[1] = \cdots = S_{i-1}[1] \\ \mathsf{S}_4 : S_i[i] = \cdots = S_{N-1}[i] \\ \mathsf{S}_5 : \bar{K}[i] = j_i - \sigma_i(t) \\ \mathsf{E}_1 : j_i = 1 \\ \mathsf{B} : \bar{K}[i] = 1 - \sigma_i(t) \end{cases}$$

Now, we compute the theoretical success probability of the attack. The goal is to estimate $\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2,\mathsf{C}_3]$. Using a similar approach as A_u15, we end up with

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2,\mathsf{C}_3] = \mathsf{Pr}(\mathsf{E}_1|\mathsf{C})\ .\ \left(\frac{NP_B(i,t)-1}{N-1}\right) + \left(\frac{1-P_B(i,t)}{N-1}\right)$$

where

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] \approx \mathsf{Pr}(\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4|\mathsf{E}_1\mathsf{C}_3) + \frac{1}{N}\left(1 - P_A^2(i,t)\ .\ \left(\frac{N-2}{N}\right)^{N-i-1}\right)$$

$$\begin{aligned}\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4|\mathsf{E}_1\mathsf{C}_3] &= \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{E}_1|\mathsf{C}_3]}{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3]}\right)\\ &= \mathsf{Pr}[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{E}_1]\ .\ \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_3]\ .\ \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3]}\right)\\ &= \left(\frac{N-1}{N}\right)^{t+1}\left(\frac{N-2}{N}\right)^{N-1-i}\ .\ P_A^2(i,t)\end{aligned}$$

$\mathsf{Pr}[\mathsf{C}_3]$ is uniformly distributed in this case and we also have

$$\mathsf{Pr}[S_t[i]=i] = \left(\frac{N-1}{N}\right)^{t+1}$$

Therefore, overall we obtain

$$\begin{aligned}\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3] = &\left(\frac{NP_B(i,t)-1}{N-1}\right)\\ &.\ \left[\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{N-2}{N}\right)^{N-1-i}\ .\ P_A^2(i,t) + \frac{1}{N}\left(1-P_A^2(i,t)\left(\frac{N-2}{N}\right)^{N-i-1}\right)\right]\\ &+\left(\frac{1-P_B(i,t)}{N-1}\right)\end{aligned}$$

C.7 The A_u13_3 Attack

- **Conditions:** $S_t[i]=i$, $S_t[1]=1-i$ and $z_1=1-i$
- **Attack path:** (see Fig. 17)
  - $S_t[1]=\cdots=S_{i-1}[1]=S_i[i]=\cdots=S_{N-1}[i]=1-i$
  - $S_t[i]=\cdots=S_{i-1}[i]=S_i[1]=\cdots=S_{N-1}[1]=i$
  - $j_i=1$
- **Key recovery relation:** $\bar{K}[i]=1-\sigma_i(t)$
- **Probability of success:** $P_u^3(i,t)$ (see Appendix D)

This attack is following the same steps as the previous attack, but with different parameters. Again, we assume $j_i=1$. We know that $S_{i-1}[1]=1-i$ and $S_{i-1}[i]=i$. At the i-th stage, after the swap we have $S_i[1]=i$ and $S_i[i]=1-i$. According to the attack path, these two values are maintained through the entire KSA algorithm. In the PRGA, for $i=1$, we have $j'_1=S_{N-1}[1]=i$. Hence, $S_{N-1}[1]$ and $S_{N-1}[i]$ are swapped. Finally, $z_1=S'_1[S'_1[1]+S'_1[i]]=S'_1[1]=1-i$. Hence, the key recovery equation is $\bar{K}[i]=1-\sigma_i(t)$.

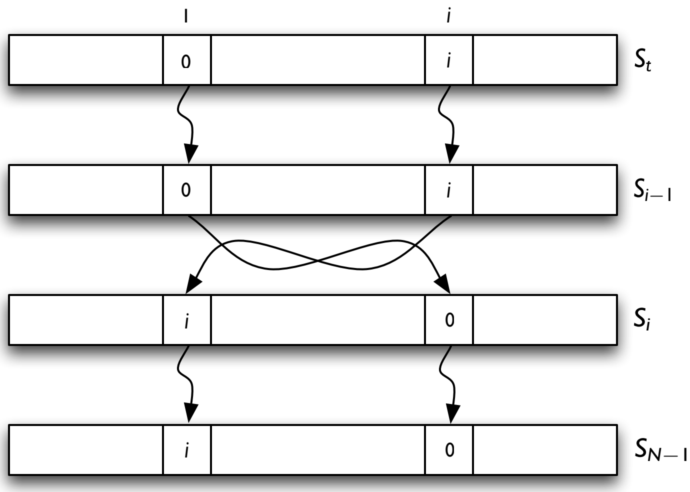

Fig. 16. RC4 state update in the A_u13_2 attack

We classify the conditions as

$$\mathsf{C}_1 : S_t[i] = i \quad \text{and} \quad \mathsf{C}_2 : S_t[1] = 1 - i \quad \text{and} \qquad \mathsf{C}_3 : z_1 = 1 - i$$

We also classify the attack path assumptions and the key recovery equation as

$$\begin{cases} \mathsf{S}_1 : S_t[i] = \cdots = S_{i-1}[i] \\ \mathsf{S}_2 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_3 : S_t[1] = \cdots = S_{i-1}[1] \\ \mathsf{S}_4 : S_i[i] = \cdots = S_{N-1}[i] \\ \mathsf{S}_5 : K[i] = j_i - \sigma_i(t) \\ \mathsf{E}_1 : j_i = 1 \\ \mathsf{B} : K[i] = 1 - \sigma_i(t) \end{cases}$$

Now, we compute the theoretical success probability of the attack. The goal is to estimate $\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3]$. Using a similar approach as A_u15, we end up with

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3] = \mathsf{Pr}(\mathsf{E}_1|\mathsf{C}) \;.\; \left(\frac{NP_B(i,t) - 1}{N-1}\right) + \left(\frac{1 - P_B(i,t)}{N-1}\right)$$

where

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] \approx \mathsf{Pr}(\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4|\mathsf{E}_1\mathsf{C}_3) + \frac{1}{N}\left(1 - P_A^2(i,t) \;.\; \left(\frac{N-2}{N}\right)^{N-i-1}\right)$$

$$\begin{aligned} \mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4|\mathsf{E}_1\mathsf{C}_3] &= \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{E}_1|\mathsf{C}_3]}{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3]}\right) \\ &= \mathsf{Pr}[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{E}_1] \;.\; \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_3] \;.\; \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3]}\right) \\ &= \left(\frac{N-1}{N}\right)^{t+1} \left(\frac{N-2}{N}\right)^{N-1-i} \;.\; P_A^2(i,t) \end{aligned}$$

$\Pr[C_3]$ is uniformly distributed in this case and we also have

$$\Pr[S_t[i] = i] = \left(\frac{N-1}{N}\right)^{t+1}$$

Therefore, overall we have

$$\begin{aligned}\Pr[B|C_1C_2C_3] = & \left(\frac{NP_B(i,t)-1}{N-1}\right) \\ & \cdot \left[\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{N-2}{N}\right)^{N-1-i} . P_A^2(i,t) + \frac{1}{N}\left(1-P_A^2(i,t)\left(\frac{N-2}{N}\right)^{N-i-1}\right)\right] \\ & + \left(\frac{1-P_B(i,t)}{N-1}\right)\end{aligned}$$

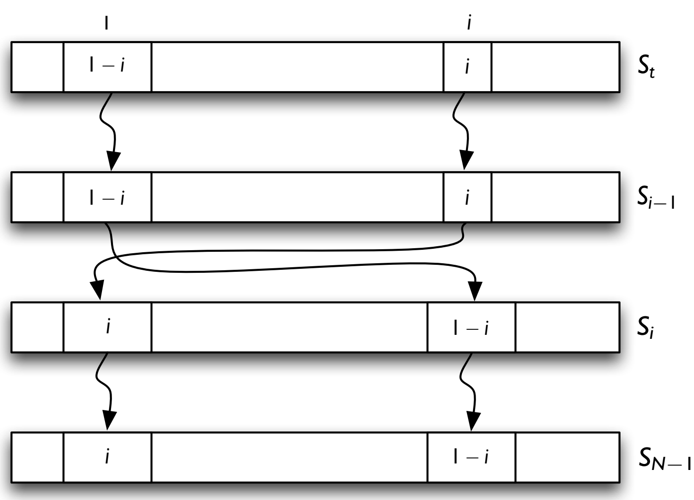

Fig. 17. RC4 state update in the A_u13_3 attack

C.8 The A_u5_3 Attack

- **Conditions:** $S_t[i] = i$, $S_t^{-1}[z_1] \neq 1$, $S_t^{-1}[z_1] < t+1$ and $z_1 = S_t[S_t[1]+i]$
- **Attack path:** (see Fig.18)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[i] = \cdots = S_{N-1}[i] = S_t^{-1}[z_1] - i$
  - $S_t[i] = \cdots = S_{i-1}[i] = S_i[1] = \cdots = S_{N-1}[1] = i$
  - $S_t^{-1}[z_1] = \cdots = S_{i-1}^{-1}[z_1] = S_i^{-1}[z_1] = \cdots = S_{N-1}^{-1}[z_1]$
  - $j_i = 1$

– **Key recovery relation:** $\bar{K}[i] = 1 - \sigma_i(t)$
– **Probability of success:** $P_u^5(i,t)$ (see Appendix D)

This attack is the extension of the A_u13_2 and the A_u13_3 attacks. Again, we assume $j_i = 1$. We know that $S_{i-1}[1] = S_t^{-1}[z_1] - i$ and $S_{i-1}[i] = i$. At the i-th stage, after the swap we have $S_i[1] = i$ and $S_i[i] = S_t^{-1}[z_1] - i$. We assume these two values and $S_t^{-1}[z_1]$ are maintained through the entire KSA. In the PRGA, initially $i = 1$ and $j_1' = S_{N-1}[1] = i$. So, $S_{N-1}[1]$ and $S_{N-1}[i]$ are swapped. Then, $z_1 = S_1'[S_1'[1] + S_1'[i]] = S_1'[S_t^{-1}[z_1] - i + i] = z_1$. Hence, the key recovery equation is $\bar{K}[i] = 1 - \sigma_i(t)$. The condition $S_t^{-1}[z_1] \neq 1$ is to filter the attack A_u13_3.

We classify the conditions as

$$\mathsf{C}_1 : S_t[i] = i \quad \text{and} \quad \mathsf{C}_2 : S_t^{-1}[z_1] \neq 1, S_t^{-1}[z_1] < t+1 \quad \text{and} \quad \mathsf{C}_3 : z_1 = S_t[S_t[1] + i]$$

We also classify the attack path assumptions and the key recovery equation as

$$\begin{cases} \mathsf{S}_1 : S_t[i] = \cdots = S_{i-1}[i] \\ \mathsf{S}_2 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_3 : S_t[1] = \cdots = S_{i-1}[1] \\ \mathsf{S}_4 : S_i[i] = \cdots = S_{N-1}[i] \\ \mathsf{S}_5 : S_t^{-1}[z_1] = \cdots = S_{N-1}^{-1}[z_1] \\ \mathsf{S}_6 : \bar{K}[i] = j_i - \sigma_i(t) \\ \mathsf{E}_1 : j_i = 1 \\ \mathsf{B} : \bar{K}[i] = 1 - \sigma_i(t) \end{cases}$$

Now, we compute the theoretical success probability of the attack. The goal is to estimate $\Pr[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2,\mathsf{C}_3]$. So, we compute

$$\begin{aligned} \Pr[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2,\mathsf{C}_3] &= \Pr[\mathsf{E}_1\mathsf{S}_6|\mathsf{C}] + \Pr[\mathsf{B}\neg\mathsf{S}_3|\mathsf{C}] \\ &= \Pr[\mathsf{E}_1|\mathsf{S}_6\mathsf{C}] \,.\, \Pr[\mathsf{S}_6|\mathsf{C}] + \Pr[\mathsf{B}|\neg\mathsf{S}_6\mathsf{C}] \,.\, (1 - \Pr[\mathsf{S}_6|\mathsf{C}]) \\ &\approx \Pr[\mathsf{E}_1|\mathsf{S}_6\mathsf{C}] \,.\, \Pr[\mathsf{S}_6|\mathsf{C}] + \left(\tfrac{1-\Pr[\mathsf{E}_1|\mathsf{S}_6\mathsf{C}]}{N-1}\right) \,.\, (1 - \Pr[\mathsf{S}_6|\mathsf{C}]) \\ &= \Pr(\mathsf{E}_1|\mathsf{S}_6\mathsf{C}) \,.\, \left(\tfrac{N\Pr[\mathsf{S}_6|\mathsf{C}]-1}{N-1}\right) + \left(\tfrac{1-P_B(i,t)}{N-1}\right) \end{aligned}$$

We then approximate $\Pr[\mathsf{S}_6|\mathsf{C}] \approx P_B(i,t)$ and we also have

$$\begin{aligned} \Pr[\mathsf{E}_1|\mathsf{S}_6\mathsf{C}] &\approx \Pr(\mathsf{E}_1|\mathsf{C}) \\ &= \Pr(\mathsf{C}_1\mathsf{C}_2|\mathsf{E}_1\mathsf{C}_3)\left(\tfrac{\Pr(\mathsf{E}_1|\mathsf{C}_3)}{\Pr(\mathsf{C}_1\mathsf{C}_2|\mathsf{C}_3)}\right) \\ &\approx \Pr(\mathsf{C}_1\mathsf{C}_2|\mathsf{E}_1\mathsf{C}_3) \\ &= \Pr(\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3) + \Pr(\mathsf{C}_1\mathsf{C}_2\neg(\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5)|\mathsf{E}_1\mathsf{C}_3) \\ &\approx \Pr(\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3) + \tfrac{1}{N}(1 - \Pr(\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3)) \\ &\approx \Pr(\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3) + \tfrac{1}{N}\left(1 - P_A^1(i,t) \,.\, \left(\tfrac{N-1}{N}\right)^{N-i}\right) \end{aligned}$$

$$\begin{aligned} \Pr[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3] &= \left(\frac{\Pr[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5\mathsf{E}_1|\mathsf{C}_3]}{\Pr[\mathsf{E}_1|\mathsf{C}_3]}\right) \\ &= \Pr[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5\mathsf{E}_1] \,.\, \left(\frac{\Pr[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5\mathsf{E}_1]}{\Pr[\mathsf{C}_3] \,.\, \Pr[\mathsf{E}_1|\mathsf{C}_3]}\right) \end{aligned}$$

$$\Pr[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2,\mathsf{C}_3] = \Pr(\mathsf{E}_1|\mathsf{S}_6\mathsf{C}) \,.\, \left(\frac{NP_B(i,t)-1}{N-1}\right) + \left(\frac{1-P_B(i,t)}{N-1}\right)$$

where

$$\Pr[\mathsf{E}_1|\mathsf{S}_6\mathsf{C}] \approx \Pr(\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3) + \frac{1}{N}\left(1 - P_A^3(i,t) \,.\, \left(\frac{N-3}{N}\right)^{N-i-1}\right)$$

$$\begin{aligned}
\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3] &= \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5\mathsf{E}_1|\mathsf{C}_3]}{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3]}\right) \\
&= \mathsf{Pr}[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5\mathsf{E}_1]\ .\ \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_3]\ .\ \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3]}\right)
\end{aligned}$$

$\mathsf{Pr}[\mathsf{C}_3]$ is uniformly distributed in this case and we also have

$$\mathsf{Pr}[S_t[i]=i] = \left(\frac{N-1}{N}\right)^{t+1}$$

Finally,

$$\begin{aligned}
\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_3] &= \mathsf{Pr}[\mathsf{C}_3|\mathsf{E}_1]\left(\frac{\mathsf{Pr}[\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_3]}\right) = \mathsf{Pr}[\mathsf{C}_3|\mathsf{E}_1] \\
&= \mathsf{Pr}[\mathsf{C}_3|\mathsf{E}_1\mathsf{C}_1\mathsf{C}_2]\ .\ \mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2|\mathsf{E}_1] + \mathsf{Pr}[\mathsf{C}_3|\mathsf{E}_1\overline{\mathsf{C}_1\mathsf{C}_2}]\ .\ \mathsf{Pr}[\overline{\mathsf{C}_1\mathsf{C}_2}|\mathsf{E}_1] \\
&= \mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2|\mathsf{E}_1] + \frac{1}{N}\left(1-\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2|\mathsf{E}_1]\right) \\
&= \left(1-\frac{1}{N}\right)\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2|\mathsf{E}_1] + \frac{1}{N}
\end{aligned}$$

This leads to

$$\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{S}_4\mathsf{S}_5|\mathsf{E}_1\mathsf{C}_3] = \frac{\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{t}{N}\right)\left(\frac{N-3}{N}\right)^{N-1-i}P_A^3(i,t)}{\left(1-\frac{1}{N}\right)\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{t}{N}\right)+\frac{1}{N}}$$

Therefore, overall we have

$$\begin{aligned}
\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3] &= \left(\frac{NP_B(i,t)-1}{N-1}\right) \\
&\quad .\ \left[\frac{\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{t}{N}\right)\left(\frac{N-3}{N}\right)^{N-1-i}}{\left(1-\frac{1}{N}\right)\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{t}{N}\right)+\frac{1}{N}}\ .\ P_A^3(i,t) + \frac{1}{N}\left(1-P_A^3(i,t)\left(\frac{N-3}{N}\right)^{N-i-1}\right)\right] \\
&\quad + \left(\frac{1-P_B(i,t)}{N-1}\right)
\end{aligned}$$

C.9 The A_s3 Attack

- **Conditions:** $S_t[1]\neq 2$, $S_t[2]\neq 0$, $S_t[2]+S_t[1]<t+1$, $S_t[2]+S_t[S_t[2]+S_t[1]]=i$, $S_t^{-1}[z_2]\neq\{1,2,S_t[1]+S_t[2]\}$, $S_t[1]+S_t[2]\neq\{1,2\}$ and $(S_t^{-1}[z_2]<t+1$ or $S_t^{-1}[z_2]>i-1)$
- **Attack path:** (see Fig. 19)
  - $S_t[1]=\cdots=S_{i-1}[1]=S_i[1]=\cdots=S_{N-1}[1]$
  - $S_t[2]=\cdots=S_{i-1}[2]=S_i[2]=\cdots=S_{N-1}[2]$
  - $S_t[S_t[1]+S_t[2]]=\cdots=S_{i-1}[S_{i-1}[1]+S_{i-1}[2]]=S_i[S_i[1]+S_i[2]]=\cdots=S_{N-1}[S_{N-1}[1]+S_{N-1}[2]]$
  - $S_t[j_i]=\cdots=S_{i-1}[j_i]=S_i[i]=\cdots=S_{N-1}[i]=z_2$
- **Key recovery relation:** $\overline{K}[i]=S_t^{-1}[z_2]-\sigma_i(t)$
- **Probability of success:** $\mathsf{Kor}_3^4(i,t)$ (see Appendix D)

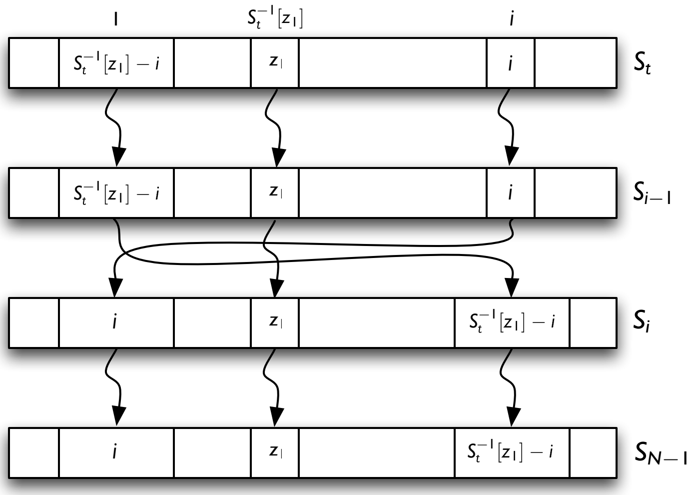

Fig. 18. RC4 state update in the A_u5_3 attack

In the first iteration of the PRGA, $i=1$ and $j'_1=S_{N-1}[1]=\alpha$, then $S_{N-1}[1]$ and $S_{N-1}[\alpha]$ are swapped. Let $S'_1[2]=\beta$. At the next stage, $i=2$ and $j'_2=S'_1[2]+\alpha=\alpha+\beta$. Then, $S'_1[2]=\beta$ and $S'_1[\alpha+\beta]$ are swapped. Using one of the conditions, we have $S_t[2]+S_t[S_t[2]+S_t[1]]=i$. Therefore, we can write $\beta+S[\beta+\alpha]=i$. So, $S[\alpha+\beta]=i-\beta$. We have $S_i[i]=S_{i-1}[j_i]$ and $j_i=S_t^{-1}[z_2]$, so $S_i[i]=z_2$. If we look at how z_2 is generated, we have $z_2=S'_2[S'_2[i]+S'_2[j'_2]]=S'_2[S'_2[2]+S'_2[\alpha+\beta]]=S'_2[i-\beta+\beta]=S'_2[i]=S_i[i]=z_2$. Using the same formulas as the previous attacks we get $K[i]=S_t^{-1}[z_2]-\sigma_i(t)$. The condition $S_t[1]\neq 2$ prevents $S_t[2]$ to be swapped in the first iteration of the PRGA. The condition $S_t[2]\neq 0$ prevents z_2 to be anything except $S'_2[i]$, otherwise $z_2=i-\beta$. The condition $S_t[1]+S_t[2]<t+1$ prevents $S_t[1]+S_t[2]$ to be swapped in the next iterations of the KSA. The index of z_2 should not be $1,2$ or $S_t[1]+S_t[2]$, because then these values would modified at one stage of the algorithm. So, we need to have $S_t^{-1}[z_2]\neq\{1,2,S_t[1]+S_t[2]\}$.

C.10 The A_s5_2 Attack

- **Conditions:** $S_t[2]+S_t[1]=i$, $S_t^{-1}[S_t[1]-S_t[2]]\neq\{1,2\}$, $(S_t^{-1}[S_t[1]-S_t[2]]<t+1$ or $S_t^{-1}[S_t[1]-S_t[2]]>i-1)$ and $z_2=S_t[1]$
- **Attack path:** (see Fig. 20)
  - $S_t[1]=\cdots=S_{i-1}[1]=S_i[1]=\cdots=S_{N-1}[1]$
  - $S_t[2]=\cdots=S_{i-1}[2]=S_i[2]=\cdots=S_{N-1}[2]$
  - $S_t[j_i]=\cdots=S_{i-1}[j_i]=S_i[i]=\cdots=S_{N-1}[i]$
- **Key recovery relation:** $K[i]=S_t^{-1}[S_t[1]-S_t[2]]-\sigma_i(t)$
- **Probability of success:** $\mathrm{Kor}_2^3(i,t)$ (see Appendix D)

In the first stage of the PRGA, $i=1$ and $j'_1=S_{N-1}[1]=\alpha$. Then, $S_{N-1}[1]$ and $S_{N-1}[\alpha]$ are swapped. In the next iteration, $i=2$ and $j'_2=S'_1[2]+\alpha=\alpha+\beta=i$, where β is $S'_1[2]$ and from the conditions. We also know that $\alpha+\beta=i$. Next, $S'_1[2]$ and $S'_1[i]$ are swapped. Finally, $z_2=S'_1[S'_1[2]+S'_1[i]]$. By the key recovery equation, we assume that $j_i=S_t^{-1}[S_t[1]-S_t[2]]$. Also, we know that $S_i[i]=S_{i-1}[j_i]=S_t[1]-S_t[2]=\alpha-\beta$. Therefore, $z_2=S'_1[\alpha-\beta+\beta]=S'_1[\alpha]=\alpha=S_t[1]$. Hence, the key recovery equation is $K[i]=S_t^{-1}[S_t[1]-S_t[2]]-\sigma_i(t)$. The condition $S_t^{-1}[S_t[1]-S_t[2]]\neq\{1,2\}$ prevents j_i from being 1 or 2, so it prevents the swap of $S_{i-1}[1]$ and $S_{i-1}[2]$ in the i-th step of the KSA.

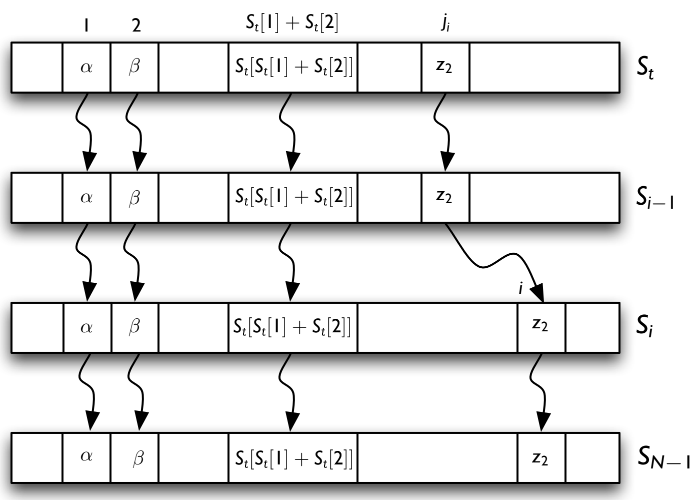

Fig. 19. RC4 state update in the A_s3 attack

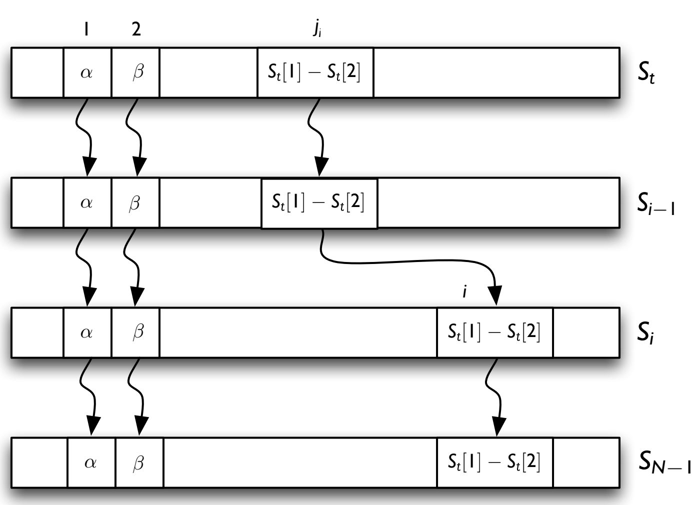

Fig. 20. RC4 state update in the A_s5_2 attack

C.11 The A_s5_3 Attack

- **Conditions:** $S_t[2]+S_t[1]=i$, $S_t^{-1}[z_2]\neq\{1,2\}$, $(S_t^{-1}[2-S_t[2]]<t+1$ or $S_t^{-1}[2-S_t[2]]>i-1)$ and $z_2=2-S_t[2]$
- **Attack path:** (see Fig.21)
  - $S_t[1]=\cdots=S_{i-1}[1]=S_i[1]=\cdots=S_{N-1}[1]$
  - $S_t[2]=\cdots=S_{i-1}[2]=S_i[2]=\cdots=S_{N-1}[2]$
  - $S_t[j_i]=\cdots=S_{i-1}[j_i]\underline{=}S_i[i]=\cdots=S_{N-1}[i]$
- **Key recovery relation:** $\overline{K}[i]=S_t^{-1}[2-S_t[2]]-\sigma_i(t)$
- **Probability of success:** $\mathrm{Kor}_2^3(i,t)$ (see Appendix D)

In the first iteration of PRGA, $i=1$ and $j_1'=S_{N-1}[1]=\alpha$. Then, $S_{N-1}[1]$ and $S_{N-1}[\alpha]$ are swapped. In the next iteration, $i=2$ and $j_2'=S_1'[2]+\alpha=\alpha+\beta=i$, where β is $S_1'[2]$ and from the conditions, we know that $\alpha+\beta=i$. Then, a swap is made between $S_1'[2]$ and $S_1'[i]$. Finally, $z_2=S_1'[S_1'[2]+S_1'[i]]$. By the key recovery equation, we assume that $j_i=S_t^{-1}[2-S_t[2]]$. Also, we know that $S_i[i]=S_{i-1}[j_i]\underline{=}2-S_t[2]=2-\beta$. Therefore, $z_2=S_1'[2-\beta+\beta]=S_1'[2]=2-S_t[2]$. Hence, the key recovery equation becomes $\overline{K}[i]=S_t^{-1}[2-S_t[2]]-\sigma_i(t)$. The condition $S_t^{-1}[z_2]\neq\{1,2\}$ prevents j_i to be 1 or 2, so it prevents the swapping of $S_{i-1}[1]$ and $S_{i-1}[2]$ in the i-th step of the KSA.

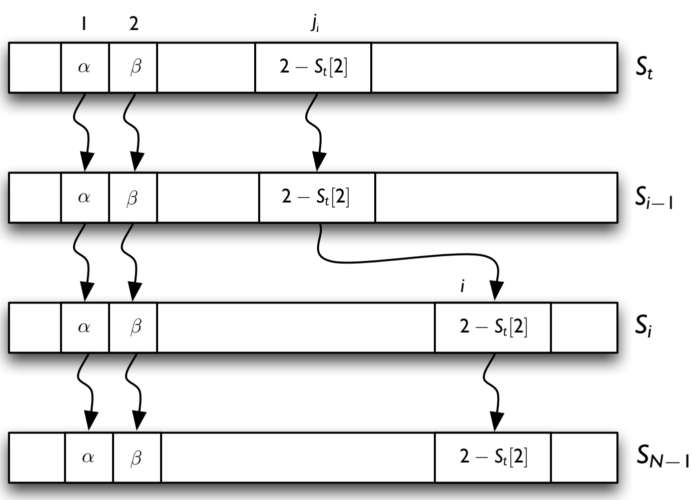

Fig. 21. RC4 state update in the A_s5_3 attack

C.12 The A_4_s13 Attack

- **Conditions:** $S_t[1]=2$, $S_t[4]\neq 0$, $(S_t^{-1}[0]<t+1$ or $S_t^{-1}[0]>i-1)$ and $z_2=0$
- **Attack path:** (see Fig. 22)
  - $S_t[1]=\cdots=S_{i-1}[1]=S_i[1]=\cdots=S_{N-1}[1]=2$
  - $S_t[j_i]=\cdots=S_{i-1}[j_i]=S_i[i]=\cdots=S_{N-1}[i]$
  - $j_4=S_t^{-1}[0]$

- $i = 4$

– **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[0] - \sigma_4(t)$

– **Probability of success:** $P^4_{\mathsf{fixed}-j}(i,t)$ (see Appendix D)

This attack only works when $i = 4$. We also assume that $j_4 = S_t^{-1}[0]$. With this assumption in mind, $S_4[4]$. In the first iteration of the PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = 2$. Then, $S_{N-1}[1]$ and $S_{N-1}[2]$ are swapped. In the next iteration, $i = 2$ and $j'_2 = S'_1[2] + 2 = 4$. Then, $S'_1[2]$ and $S'_1[4]$ are swapped. Finally, $z_2 = S'_2[S'_2[2] + S'_2[4]] = S'_2[2] = 0$. Hence, the equation for the key recovery becomes $S_t^{-1}[0] - \sigma_4(t)$. We set the condition $S_t[4] \neq 0$ to differentiate this attack from the A_u15 attack.

We classify the conditions as

$$\begin{array}{lcl} \mathsf{C}_1 : S_t[1] = 2 & \text{and} & \mathsf{C}_2 : S_t[4] \neq 0 \\ \mathsf{C}_3 : (S_t^{-1}[0] < t+1 \text{ or } S_t^{-1}[0] > i-1) & \text{and} & \mathsf{C}_4 : z_2 = 0 \end{array}$$

We also classify the attack path assumptions and the key recovery equation as

$$\begin{cases} \mathsf{S}_1 : S_t[j_4] = \cdots = S_3[j_4] \\ \mathsf{S}_2 : S_4[4] = \cdots = S_{N-1}[4] \\ \mathsf{S}_3 : S_{\underline{t}}[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_4 : \bar{K}[i] = j_i - \sigma_i(t) \\ \mathsf{E}_1 : \underline{j_i} = S_t^{-1}[0] \\ \mathsf{B} : \bar{K}[i] = S_t^{-1}[0] - \sigma_i(t) \end{cases}$$

We compute the theoretical success probability of the attack. The goal is to estimate $\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1,\mathsf{C}_2,\mathsf{C}_3,\mathsf{C}_4]$. Using a similar approach as A_u15, we end up with

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3\mathsf{C}_4] = \mathsf{Pr}(\mathsf{E}_1|\mathsf{C}) \,.\, \left(\frac{NP_B(i,t)-1}{N-1}\right) + \left(\frac{1-P_B(i,t)}{N-1}\right)$$

where

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] \approx \mathsf{Pr}(\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3|\mathsf{E}_1\mathsf{C}_4) + \frac{1}{N}\left(1 - P_A^2(i,t) \,.\, \left(\frac{N-2}{N}\right)^{N-i-1}\right)$$

$$\begin{aligned} \mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3|\mathsf{E}_1\mathsf{C}_4] &= \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{E}_1|\mathsf{C}_4]}{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_4]}\right) \\ &= \mathsf{Pr}[\mathsf{C}_4|\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{E}_1] \,.\, \left(\frac{\mathsf{Pr}[\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3\mathsf{S}_1\mathsf{S}_2\mathsf{S}_3\mathsf{E}_1]}{\mathsf{Pr}[\mathsf{C}_4] \,.\, \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_4]}\right) \\ &= \frac{1}{2}\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{N-2}{N}\right)^{N-1-i} \,.\, P_A^2(i,t) \end{aligned}$$

We know from Lemma 10 that $\mathsf{Pr}[\mathsf{C}_4] = \frac{2}{N}$ and we also have

$$\mathsf{Pr}[S_t[i] = i] = \left(\frac{N-1}{N}\right)^{t+1}$$

Therefore, overall we have

$$\begin{aligned} \mathsf{Pr}[\mathsf{B}|\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3] = &\left(\frac{NP_B(i,t)-1}{N-1}\right) \\ &\cdot \left[\frac{1}{2}\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{N-2}{N}\right)^{N-1-i} \,.\, P_A^2(i,t) + \frac{1}{N}\left(1 - P_A^2(i,t)\left(\frac{N-2}{N}\right)^{N-i-1}\right)\right] \\ &+ \left(\frac{1-P_B(i,t)}{N-1}\right) \end{aligned}$$

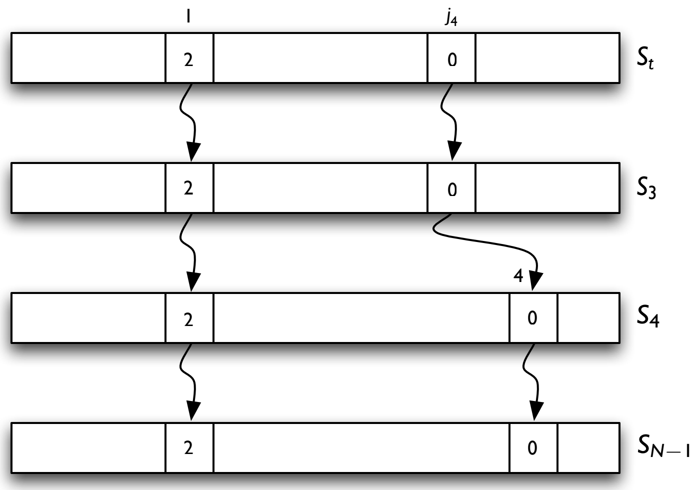

Fig. 22. RC4 state update in the A_4_s13 attack

C.13 The A_4_u5_1 Attack

- **Conditions:** $S_t[1] = 2$, $z_2 \neq 0$, $z_2 \neq N-2$, $(S_t^{-1}[N-2] < t+1$ or $S_t^{-1}[N-2] > 3)$ and $z_2 = S_t[0]$
- **Attack path:** (see Fig. 23)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = 2$
  - $S_t[0] = \cdots = S_{i-1}[0] = S_i[0] = \cdots = S_{N-1}[0] = z_2$
  - $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = \cdots = S_{N-1}[i]$
  - $i = 4$
- **Key recovery relation:** $\bar{K}[i] = S_t^{-1}[N-2] - \sigma_4(t)$
- **Probability of success:** $\mathsf{Kor}_2^3(i,t)$ (see Appendix D)

This attack only works when $i = 4$. We also know that $j_i = S_t^{-1}[N-2]$. So, $S_i[i] = S_{i-1}[j_i] = N-2$. In the first iteration of PRGA, $i = 1$ and $j_1' = S_{N-1}[1] = 2$. Then, $S_{N-1}[1]$ and $S_{N-1}[2]$ are swapped. In the next iteration, $i = 2$ and $j_2' = S_1'[2] + 2 = 4$. Next, $S_1'[2]$ and $S_1'[4]$ are swapped. Finally, $z_2 = S_2'[S_2'[2] + S_2'[4]] = S_2'[N-2+2] = S_2'[0]$. Hence, the equation for the key recovery becomes $S_t^{-1}[N-2] - \sigma_4(t)$. We set the condition $z_2 \neq 0$ to differentiate this attack from the A_4_s13 attack.

C.14 The A_4_u5_2 Attack

- **Conditions:** $S_t[1] = 2$, $z_2 \neq 0$, $(S_t^{-1}[N-1] < t+1$ or $S_t^{-1}[N-1] > 3)$ and $z_2 = S_t[2]$
- **Attack path:** (see Fig. 24)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = 2$
  - $S_t[2] = \cdots = S_{i-1}[2] = S_i[2] = \cdots = S_{N-1}[2] = z_2$
  - $S_t[j_i] = \cdots = S_{i-1}[j_i] = S_i[i] = \cdots = S_{N-1}[i]$

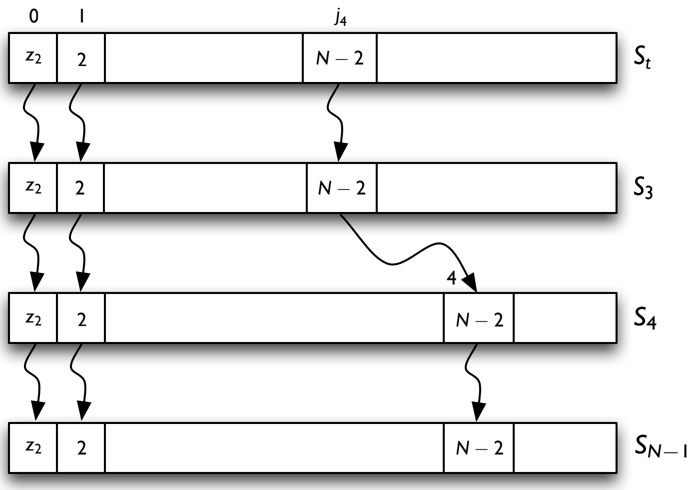

Fig. 23. RC4 state update in the A_4_u5_1 attack

- $i = 4$

– **Key recovery relation:** $\overline{K}[i] = S_t^{-1}[N-1] - \sigma_4(t)$

– **Probability of success:** $\mathsf{Kor}_2^3(i,t)$ (see Appendix D)

This attack only works when $i = 4$. We also know that $j_i = S_t^{-1}[N-1]$. So, $S_i[i] = S_{i-1}[j_i] = N-1$. In the first iteration of PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = 2$. Then, $S_{N-1}[1]$ and $S_{N-1}[2]$ are swapped. In the next iteration, $i = 2$ and $j'_2 = S'_1[2] + 2 = 4$. Then, $S'_1[2]$ and $S'_1[4]$ are swapped. Finally, $z_2 = S'_2[S'_2[2] + S'_2[4]] = S'_2[N-1+2] = S'_2[1] = S_{N-1}[2] = S_t[2]$. Hence, the equation for the key recovery becomes $S_t^{-1}[N-1] - \sigma_4(t)$. We set the condition $z_2 \neq 0$ to differentiate this attack from the A_4_s13 attack.

C.15 The A_neg_1 Attack

– **Conditions:** $S_t[2] = 0$, $S_t[1] = 2$ and $z_1 = 2$

– **Attack path:** (see Fig. 25)

- $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = 2$
- $S_t[2] = \cdots = S_{i-1}[2] = S_i[2] = \cdots = S_{N-1}[2] = 0$

– **Key recovery relation:** $\overline{K[i]} = (1 - \sigma_i(t))$ or $\overline{K[i]} = (2 - \sigma_i(t))$

– **Probability of success:** $P_{\text{neg}}(i,t)$ (see Appendix D)

In the first iteration of PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = 2$. Then, $S_{N-1}[1]$ and $S_{N-1}[2]$ are swapped. Finally, z_1 is computed as $z_1 = S'_1[S'_1[1] + S'_1[2]] = 2$. This means that $j_i \notin \{1,2\}$, otherwise it moves $S_{i-1}[1]$ or $S_{i-1}[2]$ from their current locations and so $z_1 = 2$ would not hold. Thus, we get $K[i] \neq 1 - \sigma_i(t)$ and $K[i] \neq 2 - \sigma_i(t)$.

At this stage, we compute the probability of these two negative correlations. We define the following events and conditions.

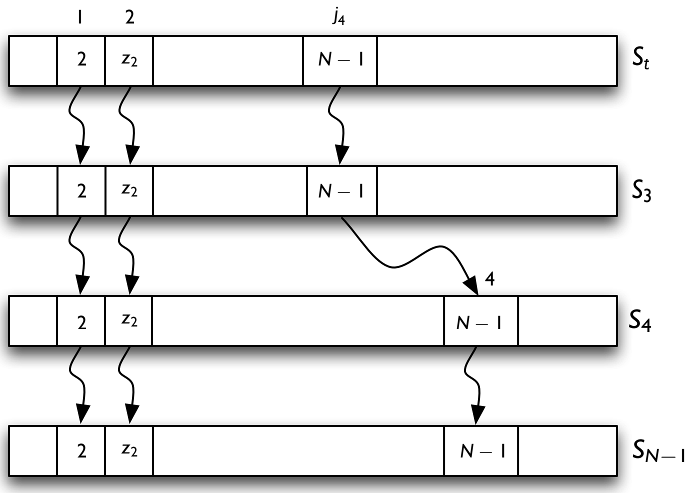

Fig. 24. RC4 state update in the A_4_u5_2 attack

$$\mathsf{E}_1 : j_i = 1 \text{ or } j_i = 2 \qquad \mathsf{B} : \bar{K}[i] = 1 - \sigma_i(t) \text{ or } \bar{K}[i] = 2 - \sigma_i(t)$$

$$\mathsf{C} : \begin{cases} \mathsf{C}_1 : S_t[2] = 0 \\ \mathsf{C}_2 : S_t[1] = 2 \\ \mathsf{C}_3 : z_1 = 2 \end{cases} \qquad \mathsf{S} : \begin{cases} \mathsf{S}_1 : S_t[1] = \cdots = S_{i-1}[1] \\ \mathsf{S}_2 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_3 : S_t[2] = \cdots = S_{i-1}[2] \\ \mathsf{S}_4 : S_i[2] = \cdots = S_{N-1}[2] \\ \mathsf{S}_5 : \bar{K}[i] = j_i - \sigma_i(t) \end{cases}$$

What we need is to compute $\mathsf{Pr}[\mathsf{B}|\mathsf{C}]$. It is computed as follows:

$$\begin{aligned} \mathsf{Pr}[\mathsf{B}|\mathsf{C}] &= \mathsf{Pr}[\mathsf{E}_1\mathsf{S}_5|\mathsf{C}] + \mathsf{Pr}[\mathsf{B}\neg\mathsf{S}_5|\mathsf{C}] \\ &= \mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}] + \mathsf{Pr}[\mathsf{B}\neg\mathsf{S}_5|\mathsf{C}] \\ &= \mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}] + \mathsf{Pr}[\mathsf{B}|\neg\mathsf{S}_5\mathsf{C}]\,(1 - \mathsf{Pr}[\mathsf{S}_5|\mathsf{C}]) \\ &\approx \mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}] + \left(\tfrac{1-\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]}{N-1}\right)(1 - \mathsf{Pr}[\mathsf{S}_5|\mathsf{C}]) \\ &= \mathsf{Pr}(\mathsf{E}_1|\mathsf{S}_5\mathsf{C})\left(\tfrac{N\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}]-1}{N-1}\right) + \left(\tfrac{1}{N-1}\right)(1 - \mathsf{Pr}[\mathsf{S}_5|\mathsf{C}]) \end{aligned}$$

We know that $\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}] \approx P_B(i,t)$, so we just need to compute $\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]$:

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}] \approx \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] = \mathsf{Pr}[\mathsf{C}_3|\mathsf{E}_1\mathsf{C}_1\mathsf{C}_2]\,.\,\left(\frac{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_1\mathsf{C}_2]}{\mathsf{Pr}[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2]}\right) \approx 0$$

So, overall, we have

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}] = \left(\frac{1 - P_B(i,t)}{N-1}\right)$$

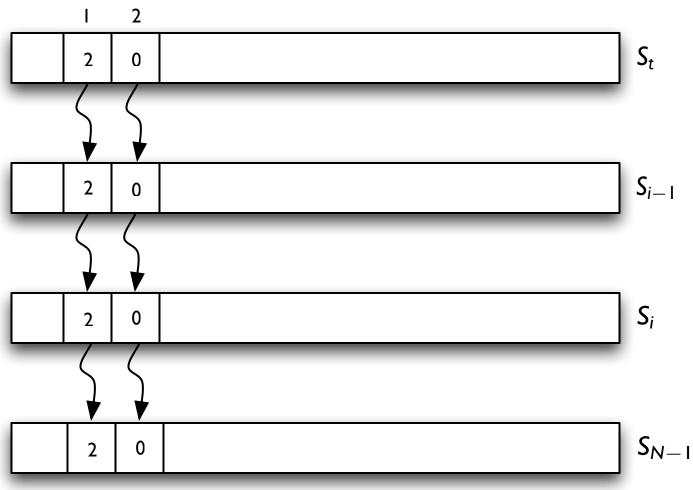

Fig. 25. RC4 state update in the A_neg_1 attack

C.16 The A_neg_2 Attack

- **Conditions:** $S_t[2] = 0$, $S_t[1] \neq 2$ and $z_2 = 0$
- **Attack path:** (see Fig. 26)
  - $S_t[2] = \cdots = S_{i-1}[2] = S_i[2] = \cdots = S_{N-1}[2] = 0$
- **Key recovery relation:** $\overline{K}[i] = (2 - \sigma_i(t))$
- **Probability of success:** $P_{\text{neg}}(i,t)$ (see Appendix D)

In the first iteration of PRGA, $i = 1$ and $j_1' = S_{N-1}[1] = \alpha$. Then, $S_{N-1}[1]$ and $S_{N-1}[\alpha]$ are swapped. In the next iteration, $i = 2$ and $j_2' = S_1'[2] + \alpha = \alpha$. Next, $S_1'[2]$ and $S_1'[\alpha]$ are swapped. Consequently, $z_2 = S_2'[S_2'[2] + S_2'[\alpha]] = S_2'[\alpha] = 0$. Similar to the previous negative attacks, if $j_i = 2$, then $S_{i-1}[2]$ will be moved in the i-th iteration of the PRGA. To differentiate between this attack and the previous one, we set $S_t[1] \neq 2$. Finally, the filtering equation for the key would be $\overline{K}[i] = (2 - \sigma_i(t))$.

We define the following events and conditions.

$$\mathsf{E}_1 : j_i = 2 \qquad \mathsf{B} : \overline{K}[i] = 2 - \sigma_i(t)$$

$$\mathsf{C} : \begin{cases} \mathsf{C}_1 : S_t[2] = 0 \\ \mathsf{C}_2 = S_t[1] \neq 2 \\ \mathsf{C}_3 : z_2 = 0 \end{cases} \qquad \mathsf{S} : \begin{cases} \mathsf{S}_1 : S_t[2] = \cdots = S_{i-1}[2] \\ \mathsf{S}_2 : S_i[2] = \cdots = S_{N-1}[2] \\ \mathsf{S}_3 : \overline{K}[i] = j_i - \sigma_i(t) \end{cases}$$

What we need is to compute $\mathsf{Pr}[\mathsf{B}|\mathsf{C}]$. It is computed as follows.

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}] \approx \mathsf{Pr}(\mathsf{E}_1|\mathsf{S}_3\mathsf{C}) \left(\frac{N\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}] - 1}{N-1} \right) + \left(\frac{1}{N-1} \right) (1 - \mathsf{Pr}[\mathsf{S}_3|\mathsf{C}])$$

We know that $\mathsf{Pr}[\mathsf{S}_3|\mathsf{C}] \approx P_B(i,t)$, so we just need to compute $\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_3\mathsf{C}]$:

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_3\mathsf{C}] \approx \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] = \mathsf{Pr}[\mathsf{C}_3|\mathsf{E}_1\mathsf{C}_1\mathsf{C}_2] \,.\, \left(\frac{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_1\mathsf{C}_2]}{\mathsf{Pr}[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2]}\right) \approx 0$$

So, overall, we have

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}] = \left(\frac{1-P_B(i,t)}{N-1}\right)$$

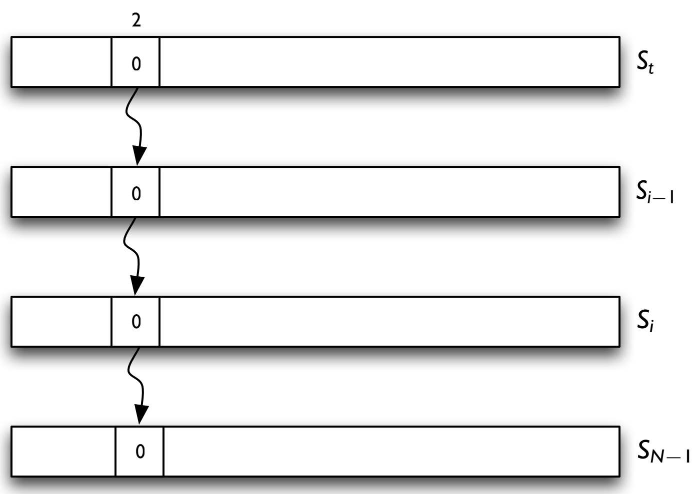

Fig. 26. RC4 state update in the A_neg_2 attack

C.17 The A_neg_3 Attack

- **Conditions:** $S_t[1] = 1$ and $z_1 = S_t[2]$
- **Attack path:** (see Fig. 27)
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = 1$
  - $S_t[2] = \cdots = S_{i-1}[2] = S_i[2] = \cdots = S_{N-1}[2] = z_1$
- **Key recovery relation:** $\overline{K[i]} = (1-\sigma_i(t))$ or $\overline{K[i]} = (2-\sigma_i(t))$
- **Probability of success:** $P_{\text{neg}}(i,t)$ (see Appendix D)

In the first iteration of the PRGA, $i = 1$ and $j_1' = S_{N-1}[1] = 1$. Consequently, $z_1 = S_1'[S_1'[1] + S_1'[1]] = S_1'[2]$. Similar to the previous negative attacks, if $j_i = 1$ or $j_i = 2$, then $S_{i-1}[1]$ or $S_{i-1}[2]$ will be relocated in the i-th iteration of the PRGA. Finally, the filtering equation for the key would be $K[i] = (1-\sigma_i(t))$ or $K[i] = (2-\sigma_i(t))$ with a very low probability.

At this stage, we compute the probability of these two negative correlations. We define the following events and conditions:

$$\mathsf{E}_1 : j_i = 1 \text{ or } j_i = 2 \qquad \mathsf{B} : \bar{K}[i] = 1 - \sigma_i(t) \text{ or } \bar{K}[i] = 1 - \sigma_i(t)$$

$$\mathsf{C} : \begin{cases} \mathsf{C}_1 : S_t[1] = 1 \\ \mathsf{C}_2 : z_1 = S_t[2] \end{cases} \qquad \mathsf{S} : \begin{cases} \mathsf{S}_1 : S_t[1] = \cdots = S_{i-1}[1] \\ \mathsf{S}_2 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_3 : S_t[2] = \cdots = S_{i-1}[2] \\ \mathsf{S}_4 : S_i[2] = \cdots = S_{N-1}[2] \\ \mathsf{S}_5 : \bar{K}[i] = j_i - \sigma_i(t) \end{cases}$$

What we need is to compute $\mathsf{Pr}[\mathsf{B}|\mathsf{C}]$. It is computed as follows:

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}] \approx \mathsf{Pr}(\mathsf{E}_1|\mathsf{S}_5\mathsf{C}) \left(\frac{N\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}] - 1}{N-1} \right) + \left(\frac{1}{N-1} \right) (1 - \mathsf{Pr}[\mathsf{S}_5|\mathsf{C}])$$

We know that $\mathsf{Pr}[\mathsf{S}_5|\mathsf{C}] \approx P_B(i,t)$, so we just need to compute $\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]$:

$$\mathsf{Pr}[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}] \approx \mathsf{Pr}[\mathsf{E}_1|\mathsf{C}] = \mathsf{Pr}[\mathsf{C}_2|\mathsf{E}_1\mathsf{C}_1] \, . \, \left(\frac{\mathsf{Pr}[\mathsf{E}_1|\mathsf{C}_1]}{\mathsf{Pr}[\mathsf{C}_2|\mathsf{C}_1]} \right) \approx 0$$

So, overall, we have

$$\mathsf{Pr}[\mathsf{B}|\mathsf{C}] = \left(\frac{1 - P_B(i,t)}{N-1} \right)$$

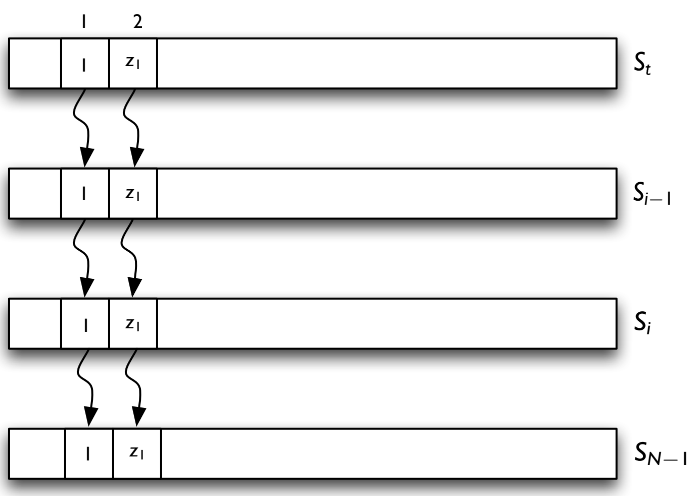

Fig. 27. RC4 state update in the A_neg_3 attack

C.18 The A_neg_4 Attack

- **Conditions:** $S_t[1] = 0$, $S_t[0] = 1$ and $z_1 = 1$
- **Attack path:** (see Fig. 28)

  - $S_t[0] = \cdots = S_{i-1}[0] = S_i[0] = \cdots = S_{N-1}[0] = 1$
  - $S_t[1] = \cdots = S_{i-1}[1] = S_i[1] = \cdots = S_{N-1}[1] = 0$
- **Key recovery relation:** $\bar{K}[i] = (-\sigma_i(t))$ or $\bar{K}[i] = (1 - \sigma_i(t))$
- **Probability of success:** $P_{\text{neg}}(i,t)$ (see Appendix D)

In the first iteration of the PRGA, $i = 1$ and $j'_1 = S_{N-1}[1] = 0$. Then, $S_{N-1}[1]$ and $S_{N-1}[0]$ are swapped. Consequently, $z_1 = S'_1[S'_1[1] + S'_1[0]] = 1$. Similar to the previous negative attacks, if $j_i = 0$ or $j_i = 1$, then $S_{i-1}[0]$ or $S_{i-1}[1]$ would be moved at the i-th step of the PRGA. Finally, the filtering equation for the key would be $\bar{K}[i] = (-\sigma_i(t))$ or $\bar{K}[i] = (1 - \sigma_i(t))$ which occurs with a low probability.

We compute the probability of these two negative biases. We define the following events and conditions:

$$\mathsf{E}_1 : j_i = 0 \text{ or } j_i = 1 \qquad \mathsf{B} : \bar{K}[i] = -\sigma_i(t) \text{ or } \bar{K}[i] = 1 - \sigma_i(t)$$

$$C : \begin{cases} \mathsf{C}_1 : S_t[0] = 1 \\ \mathsf{C}_2 : S_t[1] = 0 \\ \mathsf{C}_3 : z_1 = 1 \end{cases} \qquad \mathsf{S} : \begin{cases} \mathsf{S}_1 : S_t[0] = \cdots = S_{i-1}[0] \\ \mathsf{S}_2 : S_i[0] = \cdots = S_{N-1}[0] \\ \mathsf{S}_3 : S_t[1] = \cdots = S_{i-1}[1] \\ \mathsf{S}_4 : S_i[1] = \cdots = S_{N-1}[1] \\ \mathsf{S}_5 : \bar{K}[i] = j_i - \sigma_i(t) \end{cases}$$

What we need is to compute $\Pr[\mathsf{B}|\mathsf{C}]$. It is computed as follows:

$$\Pr[\mathsf{B}|\mathsf{C}] \approx \Pr(\mathsf{E}_1|\mathsf{S}_5\mathsf{C}) \left(\frac{N\Pr[\mathsf{S}_5|\mathsf{C}] - 1}{N-1} \right) + \left(\frac{1}{N-1} \right) (1 - \Pr[\mathsf{S}_5|\mathsf{C}])$$

We know that $\Pr[\mathsf{S}_5|\mathsf{C}] \approx P_B(i,t)$, so we just need to compute $\Pr[\mathsf{E}_1|\mathsf{S}_5\mathsf{C}]$:

$$\Pr[\mathsf{E}_1|\mathsf{S}_3\mathsf{C}] \approx \Pr[\mathsf{E}_1|\mathsf{C}] = \Pr[\mathsf{C}_3|\mathsf{E}_1\mathsf{C}_1\mathsf{C}_2] \,.\, \left(\frac{\Pr[\mathsf{E}_1|\mathsf{C}_1\mathsf{C}_2]}{\Pr[\mathsf{C}_3|\mathsf{C}_1\mathsf{C}_2]} \right) \approx 0$$

So, overall, we have

$$\Pr[\mathsf{B}|\mathsf{C}] = \left(\frac{1 - P_B(i,t)}{N-1} \right)$$

C.19 The Sepehrdad-Vaudenay-Vuagnoux Correlation

- **Conditions:** $S_t^{-1}[0] < t+1$ or $S_t^{-1}[0] > 15$, $z_{16} = -16$ and $j_2 \notin \{t+1, \ldots, 15\}$ (Cond)
- **Attack path:** (see Fig. 30)
  - $S_t[j_{16}] = \cdots = S_{15}[j_{16}] = S_{16}[16] = 0$
  - $i = 16$
- **Key recovery relation:** $\bar{K}[16] = (S_t^{-1}[0] - \sigma_{16}(t))$
- **Probability of success:** $P_{\mathsf{SVV10}}(t)$ (see Appendix D)

Sepehrdad, Vaudenay and Vuagnoux showed in [65] that $\Pr[S'_{16}[j'_{16}] = 0 | z_{16} = -16]$ is not $1/N$ and it holds with probability $P_{\text{db}} = 0.038488$. This probability was derived empirically. This bias was further analyzed in [62,63] and was proved in [63]. We revisit this proof for completeness and we modify it slightly to derive a more precise proof with our notations (see Fig. 29 for the bias path). We first find the probability $\Pr[z_{16} = -16, S'_{16}[j'_{16}] = 0]$ and then using $\Pr[z_{16} = -16]$, we compute the probability above.

In the first round of the KSA, when $i = 0$ and $j_0 = K[0]$, the value 0 is swapped into $S_0[K[0]]$. The index $j_0 = K[0] \notin \{16, -16, x\}$, so that the values 16, -16 and x at these indices respectively are not swapped out in the first round of the KSA, where $16 < x < N$ and $x \neq 240$. The role of x will be clarified later. We also require that $K[0] \notin$

Fig. 28. RC4 state update in the A_neg_4 attack

$\{1,\ldots,15\}$, so that the value 0 at index $K[0]$ is not touched by the values of i during S_1 to S_{15} state updates. Thus, $K[0] \notin \{1,2,\ldots,15,16,-16,x\}$. This happens with probability $\left(1-\frac{18}{N}\right)$.

From the KSA rounds S_0 to S_{14}, none of the indices $j_1,\ldots,j_{14}$ touches the four indices $16,-16,K[0],x$. This happens with probability $\left(1-\frac{4}{N}\right)^{14}$. When $i=15$, the value of $j_{15}=-16$ with probability $\left(\frac{1}{N}\right)$. This moves -16 to index 15 in S_{15}. When $i=16$, we have

$$j_{16} = j_{15} + S_{15}[16] + K[16] = -16 + 16 + K[0] = K[0]$$

where $S_{15}[K[0]] = 0$. Hence, after the swap, we have $S_{16}[16] = 0$. Since $K[0] \neq 15$, we have $S_{16}[15] = -16$.

From S_{16} to S_{x-1}, the index j_i does not touch the indices 15, 16 and x with probability $\left(1-\frac{3}{N}\right)^{x-17}$. When $i=x$, the value of $j_x = 15$ with probability $\left(\frac{1}{N}\right)$. Due to the swap, the value x moves to $S_x[15]$ and the value -16 moves to $S_x[x] = S_x[S_x[15]]$. For the remaining $N-x-1$ rounds of the KSA and for the first 14 rounds of the PRGA, none of the j_i or j'_i values should touch the indices $15,16,x$. This happens with probability $\left(1-\frac{3}{N}\right)^{N-x+13}$. In the next state update, i.e., S'_{15}, the value x is moved to $S'_{15}[j'_{15}]$. We need to have $j'_{15} \notin \{16,x\}$, otherwise 0 and -16 are relocated. This happens with probability $\left(1-\frac{2}{N}\right)$. We need to end up with $S'_{16}[j'_{16}] = 0$. This is exactly the case, because $S'_{16}[j'_{16}] = S'_{15}[16]$ and $S'_{15}[16] = 0$. Since $j'_{16} = j'_{15} + S'_{15}[16]$, we have $j'_{16} = j'_{15}$. Hence, in the next state update, i.e., S'_{16}, the value x is moved to index 16 and zero is moved to index j'_{16}. The last probability we need to consider is the probability that -16 is not moved in the S'_{16} state update, meaning $j'_{16} \neq -16$. This is correct with probability $\left(1-\frac{1}{N}\right)$. Finally,

$$Z_{16} = S'_{16}[S'_{16}[16] + S'_{16}[j'_{16}]] = S'_{16}[S'_{16}[16]] = S'_{16}[x] = -16$$

This is the exactly the path we were searching for.

Considering another case where both events $S'_{16}[j'_{16}] = 0$ and $z_{16} = -16$ are happening with complete random association, the overall probability is computed as:

$$\mathsf{Pr}[S'_{16}[j'_{16}] = 0, z_{16} = -16] = \frac{1}{N^2} + \left(1 - \frac{1}{N^2}\right)\gamma$$

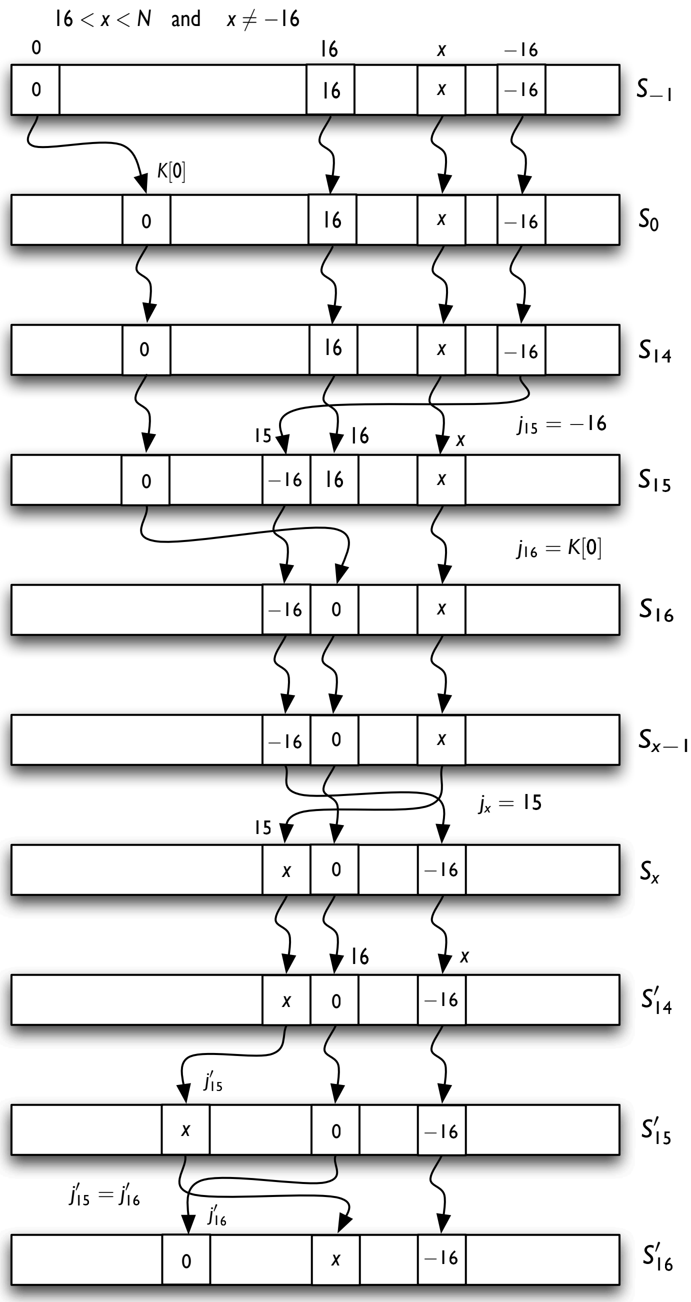

Fig. 29. RC4 state update in the SVV_10 correlation

where γ is the probability that the bias path is correct and is computed as:

$$\begin{aligned}\gamma &= \left(1-\tfrac{18}{N}\right)\left(1-\tfrac{4}{N}\right)^{14}\left(\tfrac{1}{N^2}\right)\left(1-\tfrac{2}{N}\right)\left(1-\tfrac{1}{N}\right)\sum_{\substack{x=17\\x\neq 240}}^{N-1}\left(1-\frac{3}{N}\right)^{x-17+N-x+13}\\ &= \left(\tfrac{N-18}{N^2}\right)\left(1-\tfrac{4}{N}\right)^{14}\left(1-\tfrac{3}{N}\right)^{N-4}\left(1-\tfrac{18}{N}\right)\left(1-\tfrac{2}{N}\right)\left(1-\tfrac{1}{N}\right)\end{aligned}$$

To compute $\mathsf{Pr}[S'_{16}[j'_{16}]=0|z_{16}=-16]$, we need to find $\mathsf{Pr}[z_{16}=-16]$. Recalling the different steps of computing this probability is pretty involved in [63], therefore we refer the interested reader to [63] for the proof of $\mathsf{Pr}[z_{16}=-16]=1.0355/N$. Consequently, the overall probability is:

$$\mathsf{Pr}[S'_{16}[j'_{16}]=0|z_{16}=-16]=\frac{1}{1.0355}\left[\frac{1}{N}+\left(N-\frac{1}{N}\right)\gamma\right]$$

Using the $\mathsf{SVV\_10}$ bias, the overall probability of the bias between the keystream bytes and the key bytes are not easily computable. Therefore, we refined this bias to derive a new one $\mathsf{Pr}[S_{16}[16]=0|\mathsf{Cond1}]=P_{\text{db2}}=0.03689$, where $\mathsf{Cond1}$ denotes $z_{16}=-16$. In the following, we also recall the proof of this bias from [63]:

$$\begin{aligned}\mathsf{Pr}[S_{16}[15]=-16] &= \mathsf{Pr}[S_{16}[15]=-16,S_{16}[16]=0]+\mathsf{Pr}[S_{16}[15]=-16,S_{16}[16]\neq 0]\\ &= \tfrac{1}{N^2}+\left(1-\tfrac{1}{N^2}\right)\alpha_{16}+\mathsf{Pr}[S_{16}[16]\neq 0]\cdot\mathsf{Pr}[S_{16}[15]=-16|S_{16}[16]\neq 0]\\ &\approx \tfrac{1}{N^2}+\left(1-\tfrac{1}{N^2}\right)\alpha_{16}+\left(1-\tfrac{1}{N}\right)\tfrac{1}{N}\\ &= \tfrac{1}{N}+\left(1-\tfrac{1}{N^2}\right)\alpha_{16}\end{aligned}$$

Now, we compute the main probability $\mathsf{Pr}[z_{16}=-16,S_{16}[16]=0]$ as follows:

$$\begin{aligned}\mathsf{Pr}[z_{16}=-16,S_{16}[16]=0] &= \mathsf{Pr}[z_{16}=-16,S_{16}[16]=0,S_{16}[15]=-16]+\mathsf{Pr}[z_{16}=-16,S_{16}[16]=0,S_{16}[15]\neq -16]\\ &= \mathsf{Pr}[S_{16}[16]=0,S_{16}[15]=-16]\cdot\mathsf{Pr}[z_{16}=-16|S_{16}[16]=0,S_{16}[15]=-16]\\ &\quad+\mathsf{Pr}[S_{16}[15]\neq -16]\cdot\mathsf{Pr}[z_{16}=-16,S_{16}[16]=0|S_{16}[15]\neq -16]\end{aligned}$$

Hence, merging this bias with the weaknesses of the KSA, we obtain

$$0 \underset{\mathsf{Cond1}}{\overset{P_{\text{db2}}}{=}} S_{16}[16]=S_{15}[j_{16}] \underset{\mathsf{Cond}'}{\overset{P_A^1(16,t)}{=}} S_t[j_{16}] \qquad \text{and} \qquad j_{16} \overset{P_B(16,t)}{=} \bar{K}[16]+\sigma_{16}(t)$$

where $j_{16}\notin\{t+1,\ldots,15\}$ (Cond') due to Lemma 6. We should set $S_t^{-1}[0]<t+1$ or $S_t^{-1}[0]>15$ ($\mathsf{Cond2}$) to make sure that the index of zero is not trivially picked at the next iterations. Using Lemma 6, we obtain

$$\bar{K}[16] \underset{\mathsf{Cond}}{\overset{P_{\mathsf{SVV10}}(t)}{=}} S_t^{-1}[0]-\sigma_{16}(t)$$

which holds with the overall probability of

$$P_{\mathsf{SVV10}}(t)=P_{db2}\otimes P_A^1(16,t)\otimes P_B(16,t)$$

We found out that by adding $j_2\notin\{t+1,\ldots,15\}$ condition to the attack, we can derive a much better success rate in practice. Currently, we do not have any justification for this new condition.

Table 3. The biases for RC4, exploitable against WEP and WPA

row	reference	f	g	p
i	Klein – Improved	$S_t^{-1}[-z_i+i]-\sigma_i(t)$	$(i-z_i) \notin \{S_t[t+1],\ldots,S_t[i-1]\}$	$P_{\mathrm{KI}}(i,t)$
i	A_u15	$2-\sigma_i(t)$	$S_t[i]=0$, $z_2=0$	$P_u^1(i,t)$
i	A_s13	$S_t^{-1}[0]-\sigma_i(t)$	$S_t[1]=i$, ($S_t^{-1}[0]<t+1$ or $S_t^{-1}[0]>i-1$), $z_1=i$	$\mathsf{Kor}_1^2(i,t)$
i	A_u13_1	$S_t^{-1}[z_1]-\sigma_i(t)$	$S_t[1]=i$, ($S_t^{-1}[z_1]<t+1$ or $S_t^{-1}[z_1]>i-1$), $z_1=1-i$	$\mathsf{Kor}_1^2(i,t)$
i	A_u13_2	$1-\sigma_i(t)$	$S_t[i]=i$, $S_t[1]=0$, $z_1=i$	$P_u^3(i,t)$
i	A_u13_3	$1-\sigma_i(t)$	$S_t[i]=i$, $S_t[1]=1-i$, $z_1=1-i$	$P_u^3(i,t)$
i	A_s5_1	$S_t^{-1}[z_1]-\sigma_i(t)$	$S_t[1]<t+1$, $S_t[1]+S_t[S_t[1]]=i$, $z_1 \neq \{S_t[1],S_t[S_t[1]]\}$, ($S_t^{-1}[z_1]<t+1$ or $S_t^{-1}[z_1]>i-1$)	$\mathsf{Kor}_2^3(i,t)$
i	A_s5_2	$S_t^{-1}[S_t[1]-S_t[2]]-\sigma_i(t)$	$S_t[2]+S_t[1]=i$, $S_t^{-1}[S_t[1]-S_t[2]]\neq\{1,2\}$, ($S_t^{-1}[S_t[1]-S_t[2]]<t+1$ or $S_t^{-1}[S_t[1]-S_t[2]]>i-1$), $z_2=S_t[1]$	$\mathsf{Kor}_2^3(i,t)$
i	A_s5_3	$S_t^{-1}[z_2]-\sigma_i(t)$	$S_t[2]+S_t[1]=i$, $S_t^{-1}[z_2]\neq\{1,2\}$, ($S_t^{-1}[z_2]<t+1$ or $S_t^{-1}[z_2]>i-1$), $z_2=2-S_t[2]$	$\mathsf{Kor}_2^3(i,t)$
i	A_u5_1	$S_t^{-1}[S_t^{-1}[z_1]-i]-\sigma_i(t)$	$S_t[1]=i$, $S_t^{-1}[z_1]<t+1$, $S_t^{-1}[S_t^{-1}[z_1]-i]\neq 1$, ($S_t^{-1}[S_t^{-1}[z_1]-i]<t+1$ or $S_t^{-1}[S_t^{-1}[z_1]-i]>i-1$), $z_1\neq\{i,1-i,S_t^{-1}[z_1]-i\}$, $S_t^{-1}[z_1]\neq 2i$	$\mathsf{Kor}_2^3(i,t)$
i	A_u5_2	$1-\sigma_i(t)$	$S_t[i]=1$, $z_1=S_t[2]$	$P_u^2(i,t)$
i	A_u5_3	$1-\sigma_i(t)$	$S_t[i]=i$, $S_t^{-1}[z_1]\neq 1$, $S_t^{-1}[z_1]<t+1$, $z_1=S_t[S_t[1]+i]$	$P_u^5(i,t)$
i	A_s3	$S_t^{-1}[z_2]-\sigma_i(t)$	$S_t[1]\neq 2$, $S_t[2]\neq 0$, $S_t[2]+S_t[1]<t+1$, $S_t[2]+S_t[S_t[2]+S_t[1]]=i$, $S_t^{-1}[z_2]\neq\{1,2,S_t[1]+S_t[2]\}$, $S_t[1]+S_t[2]\neq\{1,2\}$, ($S_t^{-1}[z_2]<t+1$ or $S_t^{-1}[z_2]>i-1$)	$\mathsf{Kor}_3^4(i,t)$
4	A_4_s13	$S_t^{-1}[0]-\sigma_4(t)$	$S_t[1]=2$, $S_t[4]\neq 0$, ($S_t^{-1}[0]<t+1$ or $S_t^{-1}[0]>i-1$), $z_2=0$	$P_u^4(i,t)$
4	A_4_u5_1	$S_t^{-1}[N-2]-\sigma_4(t)$	$S_t[1]=2$, $z_2\neq 0$, $z_2=S_t[0]$, $z_2\neq N-2$, ($S_t^{-1}[N-2]<t+1$ or $S_t^{-1}[N-2]>3$)	$\mathsf{Kor}_2^3(i,t)$
4	A_4_u5_2	$S_t^{-1}[N-1]-\sigma_4(t)$	$S_t[1]=2$, $z_2\neq 0$, ($S_t^{-1}[N-1]<t+1$ or $S_t^{-1}[N-1]>3$), $z_2=S_t[2]$	$\mathsf{Kor}_2^3(i,t)$
i	A_neg_1	$1-\sigma_i(t)$ or $2-\sigma_i(t)$	$S_t[2]=0$, $S_t[1]=2$, $z_1=2$	$P_{\mathrm{neg}}(i,t)$
i	A_neg_2	$2-\sigma_i(t)$	$S_t[2]=0$, $S_t[1]\neq 2$, $z_2=0$	$P_{\mathrm{neg}}(i,t)$
i	A_neg_3	$1-\sigma_i(t)$ or $2-\sigma_i(t)$	$S_t[1]=1$, $z_1=S_t[2]$	$P_{\mathrm{neg}}(i,t)$
i	A_neg_4	$-\sigma_i(t)$ or $1-\sigma_i(t)$	$S_t[1]=0$, $S_t[0]=1$, $z_1=1$	$P_{\mathrm{neg}}(i,t)$
16	SVV_10	$S_t^{-1}[0]-\sigma_{16}(t)$	$S_t^{-1}[0]<t+1$ or $S_t^{-1}[0]>15$, $z_{16}=-16$, $j_2\notin\{t+1,\ldots,15\}$	$P_{\mathsf{SVV10}}(t)$

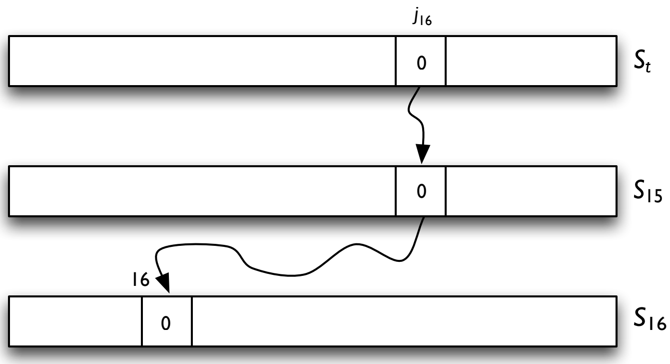

Fig. 30. RC4 state update in the SVV_10 full attack

D Correlations Probabilities Computation

Biases were computed using the following formulas:

$$
\begin{aligned}
P_{\mathrm{KI}}(i,t) &= P_J \otimes P_0 \otimes P_A^1(i,t) \otimes P_B(i,t) \\
\mathsf{Kor}_c^b(i,t) &= R_c^b(i,t) \otimes P_B(i,t) \\
P_{\mathrm{neg}}(i,t) &= \left(\tfrac{1-P_B(i,t)}{N-1}\right) \\
P_{\mathsf{SVV10}}(t) &= P_{db2} \otimes P_A^1(16,t) \otimes P_B(16,t) \\
P_u^1(i,t) &= \left(\tfrac{NP_B(i,t)-1}{N-1}\right) . \left[\tfrac{1}{2} P_A^1(i,t) \left(\tfrac{N-1}{N}\right)^{N-i} + \tfrac{1}{N}\left(1 - P_A^1(i,t)\left(\tfrac{N-1}{N}\right)^{N-i}\right)\right] + \left(\tfrac{1-P_B(i,t)}{N-1}\right) \\
P_u^2(i,t) &= \tfrac{1}{N}\left(\tfrac{NP_B(i,t)-1}{N-1}\right) . \left[\tfrac{1}{\xi} P_A^2(i,t) \left(\tfrac{N}{N-1}\right)^{t-2} \left(\tfrac{N-2}{N}\right)^{N-1-i} + \left(1 - P_A^2(i,t)\left(\tfrac{N-2}{N}\right)^{N-i-1}\right)\right] + \left(\tfrac{1-P_B(i,t)}{N-1}\right) \\
P_u^3(i,t) &= \left(\tfrac{NP_B(i,t)-1}{N-1}\right) . \left[\left(\tfrac{N-1}{N}\right)^{t+1} \left(\tfrac{N-2}{N}\right)^{N-1-i} . P_A^2(i,t) + \tfrac{1}{N}\left(1 - P_A^2(i,t)\left(\tfrac{N-2}{N}\right)^{N-i-1}\right)\right] + \left(\tfrac{1-P_B(i,t)}{N-1}\right) \\
P_u^4(i,t) &= \left(\tfrac{NP_B(i,t)-1}{N-1}\right) . \left[\tfrac{1}{2}\left(\tfrac{N-1}{N}\right)^{t+1} \left(\tfrac{N-2}{N}\right)^{N-1-i} . P_A^2(i,t) + \tfrac{1}{N}\left(1 - P_A^2(i,t)\left(\tfrac{N-2}{N}\right)^{N-i-1}\right)\right] + \left(\tfrac{1-P_B(i,t)}{N-1}\right) \\
P_u^5(i,t) &= \left(\tfrac{NP_B(i,t)-1}{N-1}\right) . \left[\frac{\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{t}{N}\right)\left(\frac{N-3}{N}\right)^{N-1-i}}{\left(1-\frac{1}{N}\right)\left(\frac{N-1}{N}\right)^{t+1}\left(\frac{t}{N}\right)+\frac{1}{N}} . P_A^3(i,t) + \tfrac{1}{N}\left(1 - P_A^3(i,t)\left(\tfrac{N-3}{N}\right)^{N-i-1}\right)\right] + \left(\tfrac{1-P_B(i,t)}{N-1}\right)
\end{aligned}
$$

where $P_J = \frac{2}{N}$, $P_0 = \left(\frac{N-1}{N}\right)^{N-2}$, $P_{\mathrm{db2}} = \frac{9.444}{N}$ and $\xi = \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^N\left(1-\frac{1}{N}+\frac{1}{N^2}\right)+\frac{1}{N^2}+1\right]$.

$$
\begin{aligned}
P_A^b(i,t) &= \left(\tfrac{N-b}{N}\right)^{i-t-1} \\
P_B(i,t) &= \prod_{k=0}^{i-t-1}\left(\tfrac{N-k}{N}\right)+\tfrac{1}{N}\left(1-\prod_{k=0}^{i-t-1}\left(\tfrac{N-k}{N}\right)\right) \\
R_c^b(i,t) &= r_c(i)P_A^b(i,t)+\tfrac{1}{N}(1-r_c(i)P_A^b(i,t)) \\
r_1(i) &= \left(\tfrac{N-2}{N}\right)^{N-i-1} \\
r_2(i) &= \left(\tfrac{N-3}{N}\right)^{N-i-1} \\
r_3(i) &= \left(\tfrac{N-4}{N}\right)^{N-i-1}
\end{aligned}
$$

These formulas are new. Biases were originally provided with probabilities for $t=-1$.