Bull. Nov. Comp. Center, Comp. Science, 43 (2019), 53-60

Analysis of RC4 encryption algorithm

Kurbonov Elyorjon Shokirovich

Abstract: The most important and cryptographically significant goal of a stream cipher is to produce a pseudorandom sequence of bits or words using a fixed length secret key, often paired with a fixed length public initialization vector. Over the last three decades of research and development in stream ciphers, a number of designs have been proposed and analyzed by the cryptology community. We briefly discuss a few major stream ciphers, relevant in terms of practical application and cutting-edge design.

Keywords: RC4 encryption algorithm, cryptograph, privacy, security, information security, encryption algorithm.

1. Introduction

RC4, also known as Alleged RC4 or ARC4, is the most widely deployed commercial stream cipher, having applications in network protocols such as SSL, WEP, WPA and in Microsoft Windows, Apple OCE, Secure SQL, etc. It was designed in 1987 by Ron Rivest for the RSA Data Security (now RSA Security). The design was a trade secret until it was anonymously posted on the web in 1994. Later, the public description was verified by comparing the outputs of the leaked design with those of the licensed systems using proprietary versions of the original cipher. Although the public design has never been officially approved or claimed by the RSA Security to be the original cipher, this note confirms that the leaked code is indeed RC4. The cipher has gained immense popularity for its intriguing simplicity, which has also made it widely accepted for numerous software and web applications.

1.1 Bluetooth™ stream cipher

Bluetooth is one of the major modern technologies for wireless communication, prevalent in an array of practical devices. The technology was developed by the Bluetooth Special Interest Group (SIG), formed in 1998. The technology has been

embraced by all companies in the communication business ever since. For confidentiality in Bluetooth transmission, the E0 stream cipher is used as the pseudorandom keystream generator. The cipher follows the standard design model of a combiner generator using linear feedback shift registers, where the keylength is typically 128 bits. Several attacks have been mounted on E0 since 1999, resulting in practical and near-practical breaches.

1.2. GSM stream ciphers

A5/1 and A5/2 stream ciphers, designed around late 1980s, are used to provide privacy in the GSM cellular network. A5/2 is a (deliberately) weakened version of A5/1, created for certain export regions. Both the ciphers A5/1 and A5/2, initially kept secret, became public in 1994 through leaks and reverse engineering. After several minor and major attacks on A5/1 and A5/2 published during 1994–2006, the GSM Association mandated that the GSM phones will not support A5/2 anymore, and usage of A5/1 was mandated by the 3GPP association. Later in the 3G cellular systems, the keystream generation algorithm for privacy was modified to A5/3, which uses the block cipher KASUMI.

1.3. 4G stream ciphers

In the race towards 4G mobile technology, 3GPP LTE Advanced [3] is the leading contender. For LTE Advanced technology, the chosen security algorithms for encryption and authentication employ two different stream ciphers – SNOW 3G and ZUC. While SNOW 3G [2] has already been deployed in the earlier 3G platform, along with KASUMI, the other cipher ZUC is a brand new design. Both ciphers are based on similar design principles using word-oriented linear feedback shift registers and are used in the LTE Advanced technology within a portfolio, along with the block cipher AES-128.

1.4. eSTREAM portfolio ciphers

The eSTREAM project, coordinated under ECRYPT framework from 2004 to 2008, was dedicated towards stream cipher research, with an aim to "identify new stream ciphers suitable for widespread adoption". Following the call for primitives, thirty-four stream ciphers were submitted to eSTREAM and an overall evaluation was done in three phases. The project came to an end in 2008 and a portfolio of seven stream ciphers was announced. The current portfolio contains

HC-128, Rabbit, Salsa20/12 and SOSEMANUK under the Software (SW) profile, and Grain v1, MICKEY v2 and Trivium under the Hardware (HW) profile. These ciphers follow cutting-edge design principles, and are projected as the stream ciphers of the future.

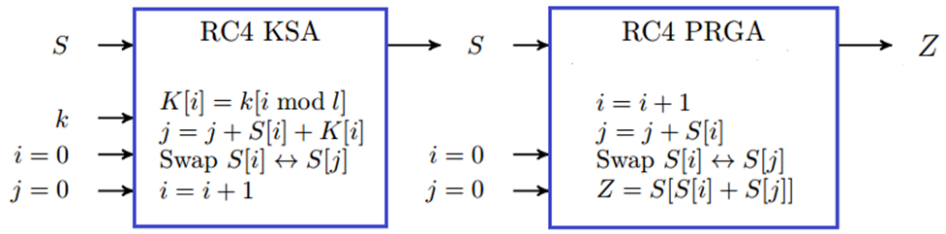
So far we have discussed the generic description of stream ciphers, and have listed some of the most prominent designs in practice: RC4, E0, A5/1, A5/2, SNOW 3G, ZUC, HC-128, Rabbit, Salsa20/12, SOSEMANUK, Grain v1, MICKEY v2, and Trivium.

Given this array of major practical stream ciphers in the literature, choice of the specific cipher RC4 for analysis and implementation deserves an explanation. The main motivation of this thesis, focused on RC4 analysis and implementation, will be summarized after a short description of the cipher.

2. Two significant components of RC4 encryption algorithm

One of the main ideas for building a stream cipher relies on constructing a pseudorandom permutation and thereafter extracting a pseudorandom sequence of words from this permutation. RC4 follows this design principle to extract pseudorandom bytes from pseudorandom permutations. The cipher consists of two major components, the Key Scheduling Algorithm (KSA) and the Pseudorandom Generation Algorithm (PRGA). The internal state of RC4 contains a permutation of all 8-bit words, a permutation of $N=2^8=256$ bytes, and the KSA produces the initial pseudorandom permutation of RC4 by scrambling an identity permutation using the secret key *k*.

The secret key k of RC4 is typically between 5 to 32 bytes long, which generates the expanded key *K* of length N = 256 bytes by simple repetition. If the length of the secret key k is l byte (typically $5 \leq l \leq 32$), then the expanded key K is constructed as $K[i] = k[i \bmod l]$ for $0 \leq i \leq N - 1$. The initial permutation produced by the KSA acts as an input to the next procedure PRGA that generates the keystream. *The RC4 algorithms KSA and PRGA are depicted in the Figure* .



Description of RC4 stream cipher

For round $r = 1, 2, \ldots$ of the RC4 PRGA, we denote the indices by i_r, j_r, the keystream output byte by Z_r, the output byte-extraction index as $t_r = S_r[i_r] + S_r[j_r]$, and the permutations before and after the swap by S_r and S_{r-1}, respectively. After r rounds of the KSA, we denote the state variables by adding a superscript K to each variable. By S_0^K and S_0 we denote the initial permutations before the KSA and PRGA, respectively. Note that $S_0 = S_N^K$ is the permutation obtained right after the completion of the KSA. Throughout this thesis, all additions (subtractions) in context of RC4 are to be considered as addition (subtraction) modulo N, and all equalities in context of RC4 are to be considered as 'congruent modulo N.

2.1. Choice of RC4 for analysis and implementation

A closer look at RC4, as described in the Figure and again presented as a formal algorithm in the Table , leaves one wondering whether these four lines of the core code, too simple even for a toy cipher, can generate a pseudorandom keystream as demanded of a stream cipher. That is the beauty of RC4! Since its public reveal through the Internet leakage in 1994, the sheer elegance and enigmatic appeal of the cipher has roots in its simple design, which is undoubtedly the simplest for any practical cryptographic algorithm to date.

The RC4 Algorithm: KSA and PRGA

Key Scheduling (KSA)	**Pseudorandom Generation (PRGA)**
Input: Secret Key k. **Output:** S-Box S_0 generated by k *Initialize* $S_0^K = \{0, 1, 2, \ldots, N-1\}$	**Input:** S Box S_0, output of KSA **Output:** Random stream Z

$K[i] = k[i \bmod l] and\ i[0][K] = j[0][K] = 0$ for (r=1;i<N; i++) $j[r][K] = j[r-1][K] + S_{r-1}^{K}[i_r^K] + K[i_r^K]$ $Swap(S_{r-1}^{K}[i_r^K] \leftrightarrow S_{r-1}^{K}[j_r^K])$ $i_r^K = i_{r-1}^K + 1$	*Initialize the counters:i[0]=j[0]=0* for (r=1;i<N; i++) $i_r = i_{r-1} + 1$ $j_r = j_{r-1} + S_{r-1}[i_r]$ $Swap(S_{r-1}^{K}[i_r^K] \leftrightarrow S_{r-1}^{K}[j_r^K])$ **Output:** $Z_r = S_r[S_r[i_r] + S_r[j_r]]$

The simplicity in design has attracted everyone towards this cipher. It has been a hit in the software industry for decades, and has been adopted as the core cipher for numerous web and software applications like Microsoft Windows, Apple OCE, Secure SQL, to name a few. The most pervasive application of RC4 however, has been in standardized web and network security protocols.

2.2. RC4 in security protocols

The IEEE 802.11 standard protocol for WiFi security used to be Wired Equivalent Privacy (WEP), which has now been replaced by Wi-Fi Protected Access (WPA). Both WEP and WPA use RC4 as their core module. In case of WEP, the protocol uses RC4 with a pre-shared key appended to a public initialization vector (nonce) for self-synchronization. Using the technique of related key attacks on RC4, this scheme has been broken through passive full-key recovery attacks, and thus WEP is considered insecure in practice. To mitigate this problem, WEP has been replaced by WPA. The goal of WPA was to resolve all security threats of WEP. However, the original WEP protocol was extensively adopted by the industry, and it was already implemented in several commercial products, both in software and hardware. This rendered a design of WPA from scratch quite impractical and costly. The work-around was to fix the full-key recovery problems of WEP using a patch, as minimal as possible, on top of the original protocol. WPA accomplished this by introducing a key fixing function to feed the RC4 core with different unrelated keys for each packet. In addition to this, WPA incorporated a packet integrity protection scheme to prevent replay and alteration of the initialization vector, which is a main tool in active attacks. It is nowadays recommended by the Wi-Fi alliance to use WPA2, which uses AES block cipher as the core instead of RC4. However, for hardware based applications and products using WEP, and later WPA, it is neither cost-effective nor easy to migrate completely away from the RC4 core. Even with the proven weaknesses in WEP, a large number of applications still have an active option for the protocol,

and users quite frequently opt for the simplicity of WEP over WPA or WPA2. Thus, RC4 continues to dominate the domain of network security to date, through the most widely used IEEE 802.11 security protocols WEP and WPA. Another prominent use of RC4 in web security is through the Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), which provide communication security over the Internet. The RC4 suite is considered to be one of the best choices for use in SSL/TLS, as it can prevent popular attacks on the protocol that primarily target the CBC cipher suite. Even though a new attack [5] on the RC4 suite of TLS has been proposed in 2013, it still remains the most popular cipher for the protocol. It is even being debated whether one should fix the recently discovered problems with a patch. In short, the users and the industry have been obsessed with RC4 for all possible web and network security solutions for more than two decades, and the cipher still remains the most popular, most studied and most debated symmetric key algorithm in practice.

3. Conclusion

So far we have discussed the generic description of stream ciphers, and have listed some of the most prominent designs in practice: RC4, E0, A5/1, A5/2, SNOW 3G, ZUC, HC-128, Rabbit, Salsa20/12, SOSEMANUK, Grain v1, MICKEY v2, Trivium. Given this array of major practical stream ciphers in the literature, choice of the specific cipher RC4 for analysis and implementation deserves an explanation. In this article we will examine the RC4 encryption algorithm.

4. References

[1] 3rd Generation Partnership Project. Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. ETSI/SAGE Specification – Document 2: ZUC Specification. – 2011. - Vol. 1.6.

[2] 3rd Generation Partnership Project. Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2. ETSI/SAGE Specification – Document 2: SNOW 3G Specification. – 2006.- Vol. 1.1.

[3] 3rd Generation Partnership Project. Long term evaluation release 10 and beyond (LTE-Advanced). Proposed to ITU at 3GPP TSG RAN Meeting. Spain. - 2009.

[4] Akgün M. , Kavak P., and Hüseyin Demirci. New results on the key scheduling algorithm of RC4. INDOCRYPT.– 2008.. – P. 40 – 52. – (Lect. Notes Comp. Sci.;5365).

[5] AlFardan N., Bernstein D., Paterson K., Poettering B., and. Schuldt J.C.N.. On the security of RC4 in TLS. // USENIX Security Symposium, 2013. online at : http://www.isg.rhul.ac.uk/tls/.

[6] Schneier B. Applied Cryptography. Protocols, Algorithms, Source Texts in C language. – M .: TRIUMPH., 2003.

[7] Paul G., Maitra S. RC4 Stream Cipher and its Variants. – CRC Press, 2019.

[8] Akbarov D. E. Cryptographic Methods of Information Security and their Application. – Tashkent : Uzbekistan, 2009.