Contents

Introduction

Despite its age, RC4 is still one of the most widely used stream ciphers in the world. It

Extract the secret key from given inner state at the end of KSA. Select the best algorithm and implement it.

TODO co budu vyuzivat, jak se to v case vyvijelo, cemu se budu venovat, tohle resili nasledujici, ten a ten umel to a to, nasledujici clanek to vylepsil, jak, mym cilem je co(sice to udelali, ale nedodali zdrojaky, je to tam vagne popsane a tak)

1. The RC4 stream cipher

The stream cipher RC4 was designed by Ronald L. Rivest for RSA Data Security in 1987 [RS14]. It was incorporated in RSA's cryptographic library and was first commercially used in Lotus Notes. The RC4 algorithm was trade secret, but it was reverse-engineered and anonymously published in Cypherpunks mailing list [Ano94] in 1994. The code was accepted to be genuine as its output matched the output produced in software using licensed RC4. To avoid trademark issues, the RC4 cipher is often referred to as ARC4 or ARCFOUR, meaning Alleged RC4.

For its simplicity and speed in software implementation it became very popular. Among implementations in products such as Skype (in modified form), Microsoft Office XP, Lotus Notes or Oracle Secure SQL, it finds application in network protocols such as WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access), SSL (Secure Socket Layer) and formerly in TLS (Transport Layer Security). It has been said that RC4 was the most widely used stream cipher in the world.

TODO lepe napsat, snazili se to ohnout nejak spatne, vzali proudovou sifru a pouzili ji spatne — wrong mode, podivat se, jak to bylo v nejakem clanku + citace **KoreK attack** In recent years many attacks have been performed on the cipher, especially on implementations involving wrong use of the initialization vector, such as WEP. In 2015 on the basis of speculations that some state agencies have capabilities to break RC4 used in TLS has IETF published RFC 7565, which prohibited use of RC4 in TLS protocol. Similar recommendation has been issued by Microsoft and Mozilla, but RC4 is still used in many systems, mostly due to legacy reasons or ease of implementation.

TODO citovat Milana - RC4 skro vsude u viru — jednoducha na implementaci

1.1 Description of the cipher

RC4 is a stream cipher, encryption uses a pseudo-random stream of bits, which are combined with the plaintext using bit-wise exlusive-or (XOR), similarly to Vernam cipher. XOR is the involution, so decryption operates the same way. To generate the pseudo random sequence, RC4 uses secret internal state which consists of:

1. a permutation of $\mathbb{Z}_N = \{0, ..., N-1\}$, where $N = 2^n$
2. two n-bit index pointers i, j used to randomize permutation table.

The pseudo-random pointer j is secret, but i is public, its value in any stage of the stream is known. Commonly $N = 256$, so i and j are 8-bit and the cipher is byte-oriented. Key length l is defined as the number of bytes in the key and can be in range $1 \leq l \leq N$. Most applications uses the key length between $5 - 16$ bytes (corresponding to $40 - 128$ bits key).

RC4 consists of two algorithms (given bellow), Key-Scheduling Algorithm (KSA), which initializes the internal permutation involving the secret key, and Pseudo-Random Generation Algorithm (PRGA), which uses the permutation to generate pseudo-random bytes. The algorithms are described bellow. **PRGA**

is in fact KSA without key byte.?? Any addition related to RC4 will be addition modulo N, unless specified otherwise.

Algorithm 1: KSA

```
begin
    Initialization:
    $j = 0$
    Scrambling:
    for $i = 0, \cdots, N-1$ do
        $j = j + S[i] + K[i \bmod l]$
        Swap(S[i],S[j])
    return $S$
```

oba vedle sebe — neco jako minipage

Algorithm 2: PRGA

```
begin
    Initialization
    $i = 0$
    $j = 0$
    Keystream generation loop
    i = i + 1
    j = j + S[i]
    Swap(S[i],S[j])
    t = S[i] + S[j]
    return $S[t]$
```

1.2 RC4 sucessors/variants

2. Previous attacks

3. Theoretical analysis of the KSA

Hlavni myslenka, nektere klice jsou pravdepodobnejsi nez jine, z toho vnitrniho stavu!!

Nekam napsat, ze to jde snadno ukrást v paměti, když vím detaily implementace (hledam 256 prvků, co jsou permutace). - kdyztak napsat milanovi

Theorem 1. *Probability of the product of independent events is the product of probabilities, implicitly assumed it is independent.*

Notation.

- $K[a...b] \coloneqq \sum_{i=a}^{b} K[i]$
- *Let j_i be the pointer j in i-th round of the KSA.*

3.1 Roos bias

Lemma 2. *TODO prerekvizita vety 1*

Lemma 3. *TODO prerekvizita vety 1*

Theorem 4. *[PM07] Assume that during the KSA the index j takes its values uniformly at random from $\mathbb{Z}_N$. Then $\forall 0 \leq i \leq r-1, 1 \leq r \leq N$*

$$\Pr(S_r[i] = K[0...i] + \frac{i(i+1)}{2}) \geq (\frac{N-i}{N})(\frac{N-1}{1})^{\frac{i(i+1)}{2}+r} + \frac{1}{N}$$

Proof. TODO □

Corollary. TODO zobecneni na posledni kolo nebo predchozi vetu rovnou smerovat tam?

TODO tabulka s aktualnimi hodnotami
TODO to same pro InvS

Theorem 5. *After the complete KSA,*

$$\Pr(S_N[S_N[y]] = K[0...i] + \frac{i(i+1)}{2}) \approx$$

clanek 4, appendix, na zacatku graf

TODO - tabulka s aktualnimi hodnotami, dukaz
TODO zobecneni na sekvence
TODO inverzni sekvence
TODO vyyiti tohoto na ziskani klice - rovnice

3.2 Substracting equations

Let $i_1 < i_2$. If $C_{i_1} = K[0...i_1]$ and $C_{i_2} = K[0...i_2]$, then we can substract the values and get

$$C_{i_2} - C_{i_1} = K[0...i_2] - K[0...i_1] = K[i_1 + 1...i_2]$$

.

This holds with the product of the individual probabilities of C_i

3.3 Useful distributions for The Key Recovering Algorithm

Definition 1. *If $j_i = S[i]$, we call this as event 1 has occured for index i, and denote as E_1.*

Theorem 6.

$$P(S[i] = j_i) \geq (1 - \frac{1}{N})^i (1 - \frac{i-1}{N})(1 - \frac{1}{N})^{N-i-1} + \frac{1}{N}$$

Definition 2. *If $j_i = S[i]$, we call this as event 1' has occured for index i, and denote as E_1'.*

Theorem 7.

$$P(S^{-1}[i] = j_i) \geq (\frac{i}{N})(1 - \frac{1}{N})^{N-1} + \frac{1}{N}$$

Proof. $(1 - \frac{1}{N})^i (\frac{i}{N})(1 - \frac{1}{N})^{N-i-1} + \frac{1}{N}$ □

Theorem 8. *$P(S[i] = j_i \vee S^{-1}[i] = j_i)$ used for BuildKeyTable algorithm in paper 1*

TODO $P(S[S[i]] = j_i)$ - dukazy nikde nejsou...
TODO $P(S^{-1}[S^{-1}[i]] = j_i)$
TODO $P(S[S[S[i]] = j_i)$

4. Going back to permutation after KSA

- I have state, i, j, number of rounds.
- I have state, i, j, not number of rounds.
- I have state, number of rounds.

5. The Key Recovering Algorithm

Tady bych chtel popsat ten, ktery ve finale pouziju. Nebo vsechny?

Conclusion

Bibliography

[Ano94] Anonymous. RC4 source code, 1994.

[PM07] G. Paul and S. Maitra. Rc4 state information at any stage reveals the secret key. *Proceedings of SAC 2007*, 2007.

[RS14] Ronald L. Rivest and Jacob C. N. Schuldt. Spritz—a spongy RC4-like stream cipher and hash function, 2014. Presented at Charles River Crypto Day (2014-10-24).