Overview

• RC4 is a stream cipher using a symmetric key
• it was developed in 1987 by Ronald Rivest and kept as a trade secret by RSA Data Security
• on September 9, 1994, the RC4 algorithm was anonymously posted through Internet to the Cyperpunks’ “anonymous remailers” list
• since then RC4 is in the public domain
Introduction

• RC4 has a programmable secret key of length from 1 to 256 bytes and a state table of 256 bytes
• secret key is used to initialize the state table
• state table is used to generate a pseudo random sequence of bytes and then a pseudo random stream of bits
• such bit stream is XORed with the plaintext to produce the ciphertext
• every element (byte) in the state table is swapped at least once
Introduction

• the key of RC4 is often limited to 40 bits because of export restrictions
• currently RC4 is used with a key of 128 bits
• RC4 may have keys from 1 to 2048 bits
• some applications of RC4:
  – many commercial software packages: Lotus Notes, MS Office and Oracle Secure SQ.
  – network protocols: SSL, IPSec and WEP
  – copy protection: inside MS XBOX
RC4 Description

• RC4 consists of two parts:
  – key schedule (to initialize the state table)
  – pseudo random generation, also called even keystream generator
  – and the data path, which is simply the XOR of the data bit stream and the generated pseudo random key stream

• this is very similar to the structure of a generic symmetric key block cipher
RC4 Key Schedule

- **Initialization**
  - for \( i = 0 \) to \( 2^{n-1} \)
    - \( S[i] = i \)
  - \( j = 0 \)

- **Scrambling**
  - for \( i = 0 \) to \( 2^{n-1} \)
    - \( j = j + S[i] + K[i \mod l] \)
    - swap \((S[i], S[j])\)

\( K = \) key array
\( l = \) size of array \( K \) (bytes)
\( n = \) size (expressed as number of bits) of the words of the tables, generally is 8
\( S[ ] = \) the state table composed by \( 2^n \) words of \( n \) bits
RC4 Pseudo Random Generation (PRNG)

- initialization
  \[ i = 0 \]
  \[ j = 0 \]

- generation loop
  \[ i = i + 1 \]
  \[ j = j + S[i] \]
  swap \((S[i], S[j])\)
  output \[ z = S[S[i] + S[j]] \]

- bit stream \(z\) is XORED to the data stream
Strengths

- difficulty of knowing the state
- difficulty of knowing which location in the state table is used to select each value in the sequence
- encryption in software is about 10 times faster than DES
Weaknesses

• vulnerable to analytic attacks to the state table
• there are a few theoretical attacks performed on the RC4 algorithm; one of the best known has been developed by Fhlurer, Martin and Shamir
• there are some weak keys (one every 256 keys may be a weak key): the output encrypted bytes depend on a small number of key bits
RC4 Hygiene Rules

- the initial 512 bytes of output must be dropped in order to defeat weak key attacks and other attacks related to key schedule
- since RC4 is used as a one-time-pad one it should never use the same key twice; most protocols use a random IV (Initialization Vector) appended to the secret key; then a hash function (e.g. SHA-1) should be applied so as to defeat related key attacks
- no more than $2^{30.6}$ bytes should be encrypted with RC4 at a time (longer sequences are not enough pseudo random)
- to encrypt larger data streams one should preferably use AES in counter mode (AES-CTR), which works like a stream cipher
SW Performances

- two implementations in C differing as for the word size of the state table S
  - version 1  (word size = 8 bits)
  - version 2  (word size = 32 bits)

<table>
<thead>
<tr>
<th></th>
<th>Pentium 1GHz</th>
<th>ARM 926 270 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>version 1</td>
<td>6.6 MBytes / s</td>
<td>10 MBytes / s</td>
</tr>
<tr>
<td>version 2</td>
<td>46 MBytes / s</td>
<td>10 MBytes / s</td>
</tr>
</tbody>
</table>
HW Implementations

- there is not much bibliography about the HW implementation of the RC4 algorithm
- Most of the time SW is good enough
HW Implementations

- algorithm is simple and implementation is quite straightforward
- main drawback is the necessity of a memory to store state table $S$, which requires a memory of 256 bytes
- suppose that one memory access plus one addition takes one clock cycle, then the time latency to output an encrypted or decrypted byte is of five clock cycles

Time necessary to execute the five operations:
read $S[i]$, read $S[j]$, write $S[i]$, write $S[j]$ and read $S(S[i] + S[j])$
About RC4 on WEP

- the Wired Equivalent Privacy protocol (WEP) uses the RC4 algorithm for encryption
- three attacks exist against WEP: wordlist, brute force and statistical analysis
- statistical analysis is possible due to the clear transmission of the Initialization Vector and of a guessable first byte of plaintext
References

6. [https://www1.ietf.org/mail-archive/working-groups/cfrg/current/msg00273.html](https://www1.ietf.org/mail-archive/working-groups/cfrg/current/msg00273.html)
References

8. P. Hamalainen, M. Hannikainen, T. Hamalainen, J. Saarinen