Team RC5

Zhenkuang He
Daniyal Syed
Brandon Mauer
Team Members

• Zhenkuang He
  1st year Graduate
  Computer Science

• Daniyal Syed
  1st year Graduate
  Computer Security and Information Assurance

• Brandon Mauer
  2nd year Graduate
  Computer Security and Information Assurance
RC5 At a Glance

• Rivest Cipher 5
• Symmetric block cipher
• Designed to be simple and fast
• Uses Feistel-like structure
• Variable size block (32/64/128 bit)
• Variable size key (0-2048 bit)
• Variable number of rounds (0-255)
• Uses "data-dependent rotations"
The RC5 Algorithm

Consists of three components:

1. a key expansion algorithm
2. an encryption algorithm
3. a decryption algorithm
Notation of RC5 Parameters

Plaintext input to RC5 consists of two \( w \)-bit words which we denote \( A \) and \( B \).

RC5 uses an expanded key table \( S[0...t-1] \), consisting of \( t = 2(r-1) \) \( w \)-bit words, where \( r \) is the number of rounds.

The key expansion algorithm initializes \( S \) from the users given secret key parameter \( K \).
The RC5 Algorithm

1. Key Expansion

The key expansion routine expands the user's secret key $K$ to fill the expanded key array $S$. Note that this does not refer to an S-box.

The key expansion algorithm uses two magic constants and consists of three simple algorithmic parts.
The RC5 Algorithm

a. Definition of the Magic Constants

The key expansion algorithm uses two word-sized binary constants $P_w$ and $Q_w$.

For $w = 32$, these constants are given below in binary and in hexadecimal.

$$P_w = P_{16} = 1011\ 0111\ 1110\ 0001 = B7E1$$
$$Q_w = Q_{16} = 1001\ 1110\ 0011\ 0111 = 9E37$$
The RC5 Algorithm

b. Converting the Secret Key from Bytes to Words

Copy the secret key $K[0...b-1]$ into an array $L[0...c-1]$ of $c = [b / u]$ words, where $u = w / 8$ is the number of bytes / word. Any unfilled byte positions of $L$ are zeroed.

Pseudo-code:

```
c = \left\lfloor \max(b, 1) / u \right\rfloor
for i = b - 1 downto 0 do
    L[i / u] = (L[i / u] <<< 8) + K[i]
```
The RC5 Algorithm

c. Initializing the Array S

Pseudo-code:

\[ S[0] = Pw; \]
\[ \text{for } i = 1 \text{ to } t - 1 \text{ do} \]
\[ \quad S[i] = S[i - 1] + Qw; \]
The RC5 Algorithm

d. Mixing in the Secret Key

Mix in the user's secret key in three passes over the arrays $S$ and $L$.

More precisely due to the potentially different sizes of $S$ and $L$. The larger array will be processed three times and the other may be handled more times.
The RC5 Algorithm

d. Mixing in the Secret Key

Pseudo-code:

\[
\begin{align*}
  & i = j = 0; \\
  & a = b = 0; \\
  & \text{do 3} \times \text{max}(t, c) \text{ times:} \\
  & \quad a = S[i] = (S[i] + a + b) \ll 3; \\
  & \quad b = L[i] = (L[j] + a + b) \ll (a + b); \\
  & \quad i = (i + 1) \mod (t); \\
  & \quad j = (j + 1) \mod (c); 
\end{align*}
\]
The RC5 Algorithm

1. Key Expansion

The key expansion function has a certain amount of "one-wayness": it is not so easy to determine K from S.
The RC5 Algorithm

2. Encryption

Pseudo-code:

\[
\begin{align*}
A &= A + S[0]; \\
B &= B + S[1]; \\
\text{for } i = 1 \text{ to } r \text{ do} \\
A &= ((A \oplus B) \ll B) + S[2 \times i] \\
B &= ((B \oplus A) \ll A) + S[2 \times i + 1]
\end{align*}
\]
The RC5 Algorithm

3. Decryption

Pseudo-code:

for i = r downto 1 do
    B = ((B - S[2 * i + 1] >>> A) Xor A;
    A = ((A - S[2 * i] >>> B) Xor B;
B = B - S[1];
A = A - S[0];
Additional Information

- RC5 mainly relies on the strength of data-dependent rotations.
- Shifts are performed mod w. If w is a power of 2, $\log_2(w)$ low-order bits determine how much shifting is performed.
- In order to maximize the variance of the shifts over each round, the round key from S is added to the intermediate ciphertext that was just computed.
- XOR operations between A and B should provide avalanche effect.
References

• Ron Rivest - "The RC5 Encryption Algorithm"

• RC5 One Round Diagram
Questions?
Thank you!