Cryptography and Network Security

RC5

Session Meta Data

Author	**Dr T Sree Sharmila**
Reviewer	
Version Number	1.0
Release Date	10 July 2018

Revision History

Revision Date	Details	Version no.
		1.0

Agenda

- Introduction
- RC5
  - Ciphers
  - Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

Introduction

- can vary key size / input data size / #rounds
- very clean and simple design
- easy implementation on various CPUs
- yet still regarded as secure
  - Vary parameters to achieve tradeoffs

Agenda

- Introduction
- RC5
  - Ciphers
  - Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

RC5 Ciphers

- RC5 is a family of ciphers RC5-w/r/b
  - w = word size in bits (16/32/64) data=2w
  - r = number of rounds (0..255)
  - b = number of bytes in key (0..255)
- nominal version is RC5-32/12/16
  - ie 32-bit words so encrypts 64-bit data blocks
  - using 12 rounds
  - with 16 bytes (128-bit) secret key

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

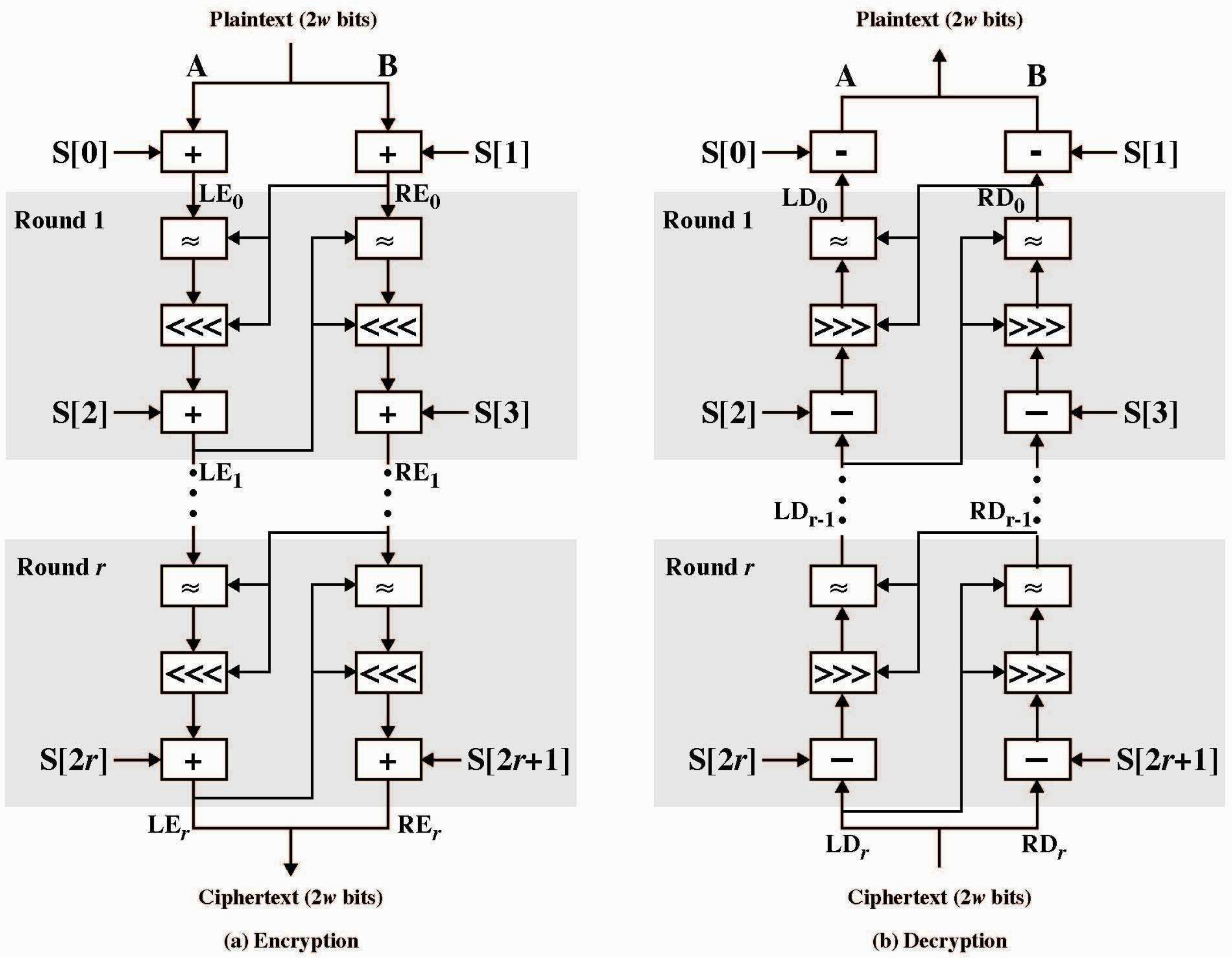
RC5 Key Expansion

- RC5 uses 2r+2 subkey words (w-bits)
  - Two subkeys for each round
  - 2 subkeys for additional operations
- subkeys are stored in array `S[i]`, i=0..t-1
- Key expansion: fill in pseudo-random bits to the original key K
- Certain amount of *one-wayness*
  - Difficult to determine K from S

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

RC5 Encryption & Decryption



(a) Encryption

(b) Decryption

Figure 6.6 RC5 Encryption and Decryption

RC5 Encryption

- split input into two halves A & B

```
$L_0 = A + S[0];$
$R_0 = B + S[1];$
for $i = 1$ to $r$ do
    $L_i = ((L_{i-1} \text{ XOR } R_{i-1}) \lll R_{i-1}) + S[2 \times i];$
    $R_i = ((R_{i-1} \text{ XOR } L_i) \lll L_i) + S[2 \times i + 1];$
```

- each round is like 2 DES rounds
- note rotation is main source of non-linearity
- need reasonable number of rounds (eg 12-16)
- Striking features: simplicity, data-dependent rotations

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

RC5 Modes

- RFC2040 defines 4 modes used by RC5
  - RC5 Block Cipher, is ECB mode
  - RC5-CBC, input length is a multiples of 2w
  - RC5-CBC-PAD, any length CBC with padding
    - Output can be longer than input
  - RC5-CTS, CBC with padding
    - Output has same length than input

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

Block Cipher Characteristics

- features seen in modern block ciphers are:
  - variable key length / block size / no rounds
  - mixed operators
    - data/key dependent rotation
    - key dependent S-boxes
  - more complex key scheduling
    - Lengthy key generation, simple encryption rounds
  - operation of full data in each round

Stream Ciphers

- process the message bit by bit (as a stream)
- typically have a (pseudo) random **key stream**
- combined (XOR) with plaintext bit by bit
- randomness of **key stream** completely destroys any statistically properties in the message
  - `C`$_i$ `= M`$_i$ `XOR StreamKey`$_i$
- what could be simpler!!!!
- but must never reuse key stream
  - otherwise can remove effect and recover messages

Block/Stream Ciphers

- Stream ciphers
  - For applications that require encryt/decryt of a stream of data
  - Examples: data communication channel, brower/web link
- Block ciphers
  - For applications dealing with blocks of data
  - Examples: file transfer, e-mail, database
- Either type can be used in virtually any application

Stream Cipher Properties

- some design considerations are:
  - long period with no repetitions
  - statistically random
  - Highly nonlinear correlation

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

RC4

- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS between browser and server, wireless WEP)
- key forms random permutation of a 8-bit string
- uses that permutation to scramble input info processed a byte at a time

RC4 Security

- claimed secure against known attacks
  - have some analyses, none practical
- result is very non-linear
- since RC4 is a stream cipher, must **never reuse a key**

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

Summary

- have considered:
  - some other modern symmetric block ciphers
  - RC5
  - RC4

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

Test your understanding

1. Explain RC5 algorithm in detail.
2. Explain RC4 algorithm in detail.
3. What are difference between RC5 & RC4 algorithm.

Agenda

- Introduction
- RC5
  - Ciphers
  - Key Expansion
  - Encryption & Decryption
  - Modes
  - Block & Stream cipher
- RC4
- Summary
- Test your understanding
- References

References

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security”, Prentice Hall of India, 2002.