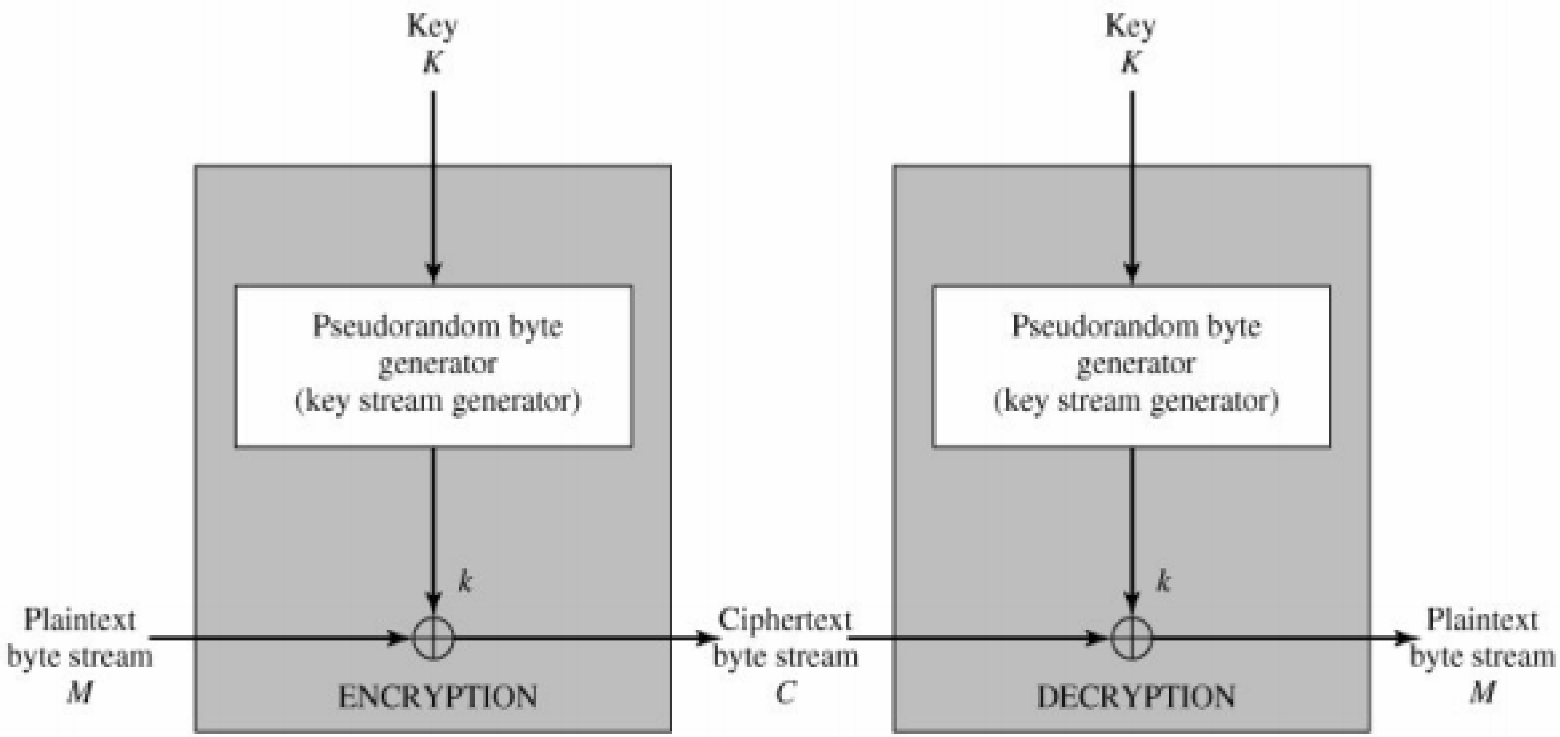
STREAM CIPHER

By : Eman Talib Jasim

Department of Information Technology 2018-2019

Stream Cipher Structure

- □ Encrypt plaintext one byte at a time
- □ Key used as input to 8-bit pseudorandom number generator to produce keystream
- □ Keystream is combined with 1 byte of plaintext at a time using bitwise XOR


Key
K
Pseudorandom byte
generator
(key stream generator)
k
Plaintext
byte stream
M
ENCRYPTION
Key
K
Pseudorandom byte
generator
(key stream generator)
k
Ciphertext
byte stream
C
DECRYPTION
Plaintext
byte stream
M

- Encryption

$$\begin{array}{ccccccccc l} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \text{plaintext} \\ \oplus & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & \text{keystream} \\ \hline & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \text{ciphertext} \end{array}$$

- Decryption

$$\begin{array}{ccccccccc l} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \text{ciphertext} \\ \oplus & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & \text{keystream} \\ \hline & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \text{plaintext} \end{array}$$

Stream Cipher Properties

- Similar to one-time pad
- Difference

  one-time pad uses genuine random generator

  stream cipher uses pseudorandom number
- Can be as secure as block cipher with comparable key length
- Faster, uses much less code
- Cannot reuse keys

Design Considerations

- Pseudorandom number must not repeats fast with small period. The longer the period of repeat the more difficult it will be to do cryptanalysis.

  large period must be used
- Pseudorandom number depends on key i.e the key must be long and number of ones equal to number of zeros
- Key needs to be sufficiently long. with current technology, a key length of at least 128 bits is desirable.

Disadvantage

- don’t encrypt more than one message in the same key because the encryption operation depends on XOR between the plaintext and the key, so the attacker can get the cipher text when he obtain the key with the ciphertext.

RC4 Algorithm

- Stream cipher designed in 1987 by Ron Rivest for RSA Security.
- Byte oriented
- Variable key size (1-256) byte
- Period greater than 10_{100}
- Run very quickly in software

- RC4 is used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards that have been defined for communication between Web browsers and servers. It is also used in the Wired Equivalent Privacy (WEP) protocol and the newer WiFi Protected Access (WPA) protocol that are part of the IEEE 802.11 wireless LAN standard.

- The next table, compares execution times of RC4 with three symmetric block ciphers. One advantage of a block cipher is that you can reuse keys. In contrast, if two plaintexts are encrypted with the same key using a stream cipher, then cryptanalysis is often quite simple.

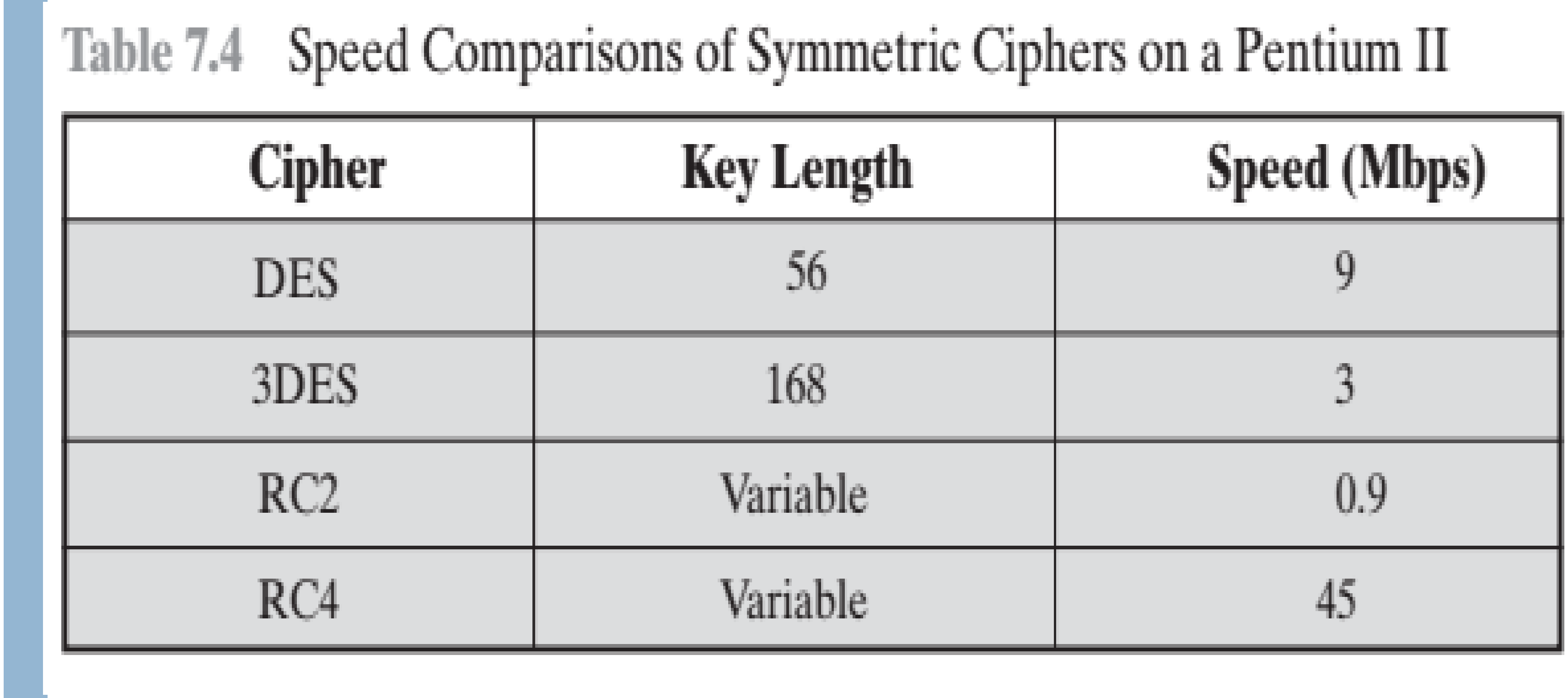


Table 7.4 Speed Comparisons of Symmetric Ciphers on a Pentium II

Cipher	Key Length	Speed (Mbps)
DES	56	9
3DES	168	3
RC2	Variable	0.9
RC4	Variable	45

RC4 Algorithm

- □ ■ Variable length key [1-256] bytes is used to initialize 256-byte state vector S
- □ ■ S = S[0] , S[1] , S[2] , ..., S[255]
- □ ■ At all times, S contains permutation of numbers between [0-255]
- □ ■ For encryption, decryption
- □ o byte k = select from S[0]-S[255] (systematic)
- □ o after generating k, S is again permuted

- **Initialization of S**
- ▪ S[0] = 0, S[1] = 1, …, S[255] = 255
- ▪ Temporary Vector T is created, K=key
- ▪ If length (K) = 256 bytes, set T=K
- ▪ Else, K is repeated until T is filled
- ▪ Use T to produce initial permutation of S

Initialization of S

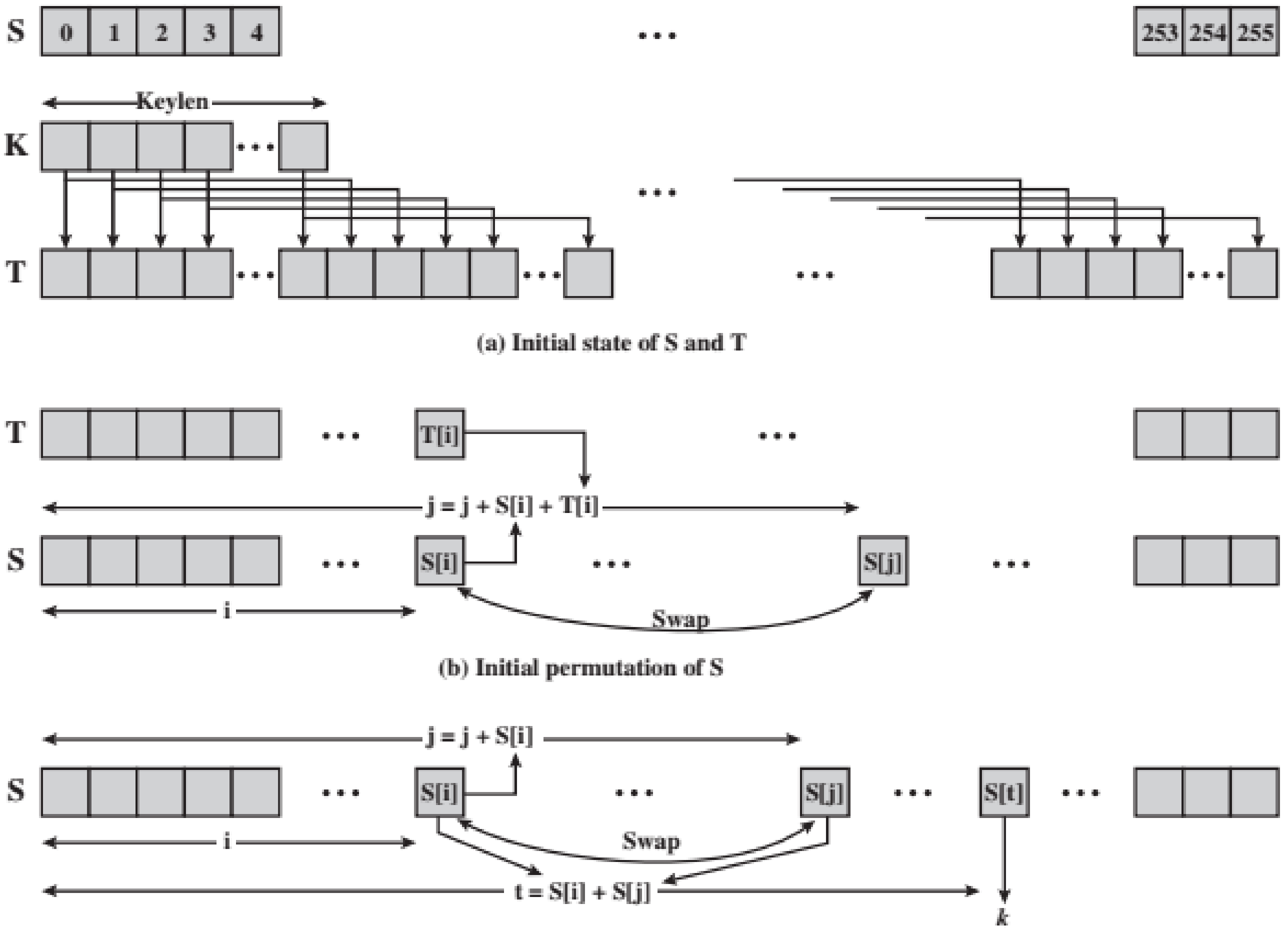
- ▪ Initialization
- for i = 0 to 255 do
- S[i] = i;
- T[i] = K[i mod keylen];
- ▪ Initial permutation of S
- j = 0;
- for i = 0 to 255 do
- j = (j + S[i] + T[i]) mod 256;
- Swap (S[i], S[j]);

- **Stream Generation**
- ▪ Once S is initialized, K is no longer used
- ▪ Cycling through all the elements of S
- ○ swap S[i] with another element
- ○ dictated by current configuration of S
- ○ when S[255] is reached, start over at S[0]

Stream Generation

- i, j = 0;
- while (true)
- i = (i + 1) mod 256;
- j = (j+ S[i]) mod 256;
- Swap (S[i], S[j]);
- t = (S[i] + S[j]) mod 256;
- k = S[t];

- Encryption
- XOR S[t] with next byte of plaintext
- Decryption
- XOR S[t] with next byte of ciphertext
- Next Figure illustrates the RC4 logic.



(a) Initial state of S and T

(b) Initial permutation of S

(c) Stream generation

Figure 7.6 RC4