University of Salahaddin - Erbil
College of Engineering
Department of Software Engineering

Data Security

Academic year 2021-2022
4th Year Material
Chapter Five

Modern Stream Cipher (RC4 & A5)

Prepared By: Mr. **Zana Farhad Doghramachi, M.Tech(CSE)**
Zana.softeng@gmail.com

Stream Cipher

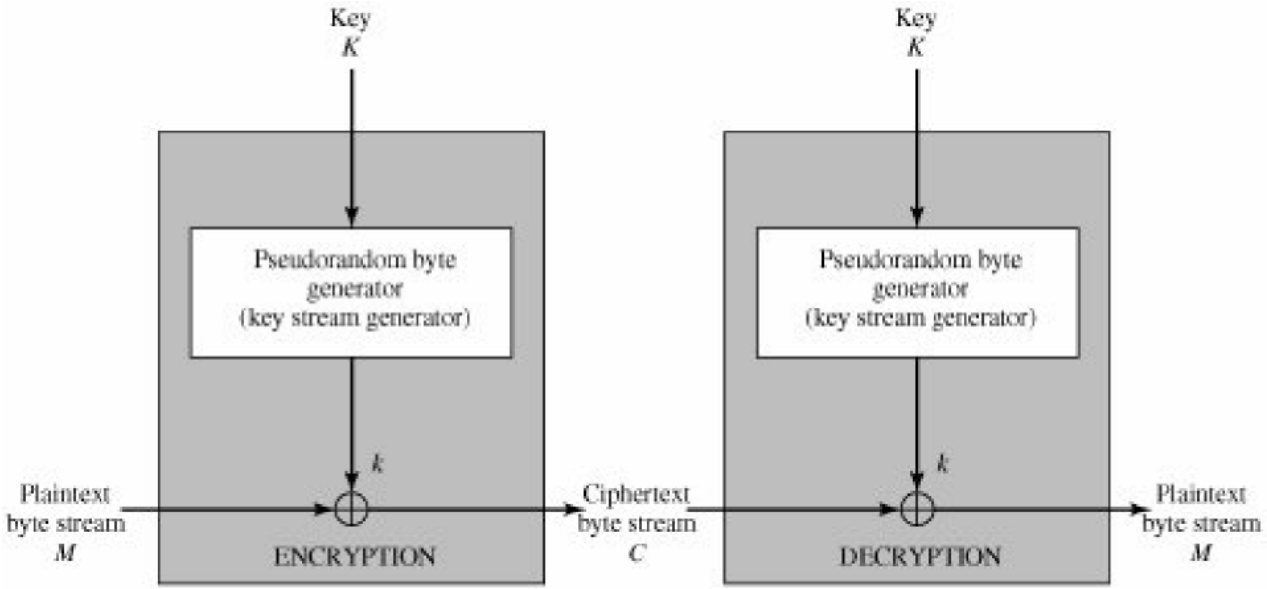
- A typical **Stream cipher** encrypts plaintext one byte at a time, although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time. In this structure a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random. Pseudorandom stream is one that is unpredictable without knowledge of the input key. The output of the generator, called a keystream, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation. For example, if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting ciphertext byte is

```
  11001100   plaintext
⊕ 01101100   key stream
  10100000   ciphertext
```

Stream Cipher Structure

- Decryption requires the use of the same pseudorandom sequence:

```
  10100000   ciphertext
⊕ 01101100   key stream
  11001100   plaintext
```



Homework

Q1 What are difference and similarity between block cipher and stream cipher?

Q2 What is pseudorandom number generator?

Stream Cipher I

- Important design considerations for a stream cipher:

1. The encryption sequence should have a large period. A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats. The longer the period of repeat the more difficult it will be to do cryptanalysis.

2. The keystream should approximate the properties of a true random number stream as close as possible. For example, there should be an approximately equal number of 1s and 0s. If the keystream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often. The more random appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult.

Stream Cipher II

3. The output of the pseudorandom number generator is conditioned on the value of the input key. To guard against brute-force attacks, the key needs to be sufficiently long. Thus, with current technology, a key length of at least 128 bits is desirable.

- The primary advantage of a stream cipher is that stream ciphers are almost always faster and use far less code than do block ciphers. The example in this section, RC4, can be implemented in just a few lines of code. The advantage of a block cipher is that you can reuse keys. However, if two plaintexts are encrypted with the same key using a stream cipher, then cryptanalysis is often quite simple.

Stream Cipher III

- Some applications that require encryption/decryption of a stream of data, such as over a data communications channel or a browser/Web link, a stream cipher might be the better alternative. For applications that deal with blocks of data, such as file transfer, e-mail, and database, block ciphers may be more appropriate. However, either type of cipher can be used in virtually any application.

RC4 Algorithm

- RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security.
- It is a variable key size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation.
- RC4 is used in the SSL/TLS (Secure Sockets Layer/Transport Layer Security) standards that have been defined for communication between web browsers and servers. It is also used in the WEP (Wired Equivalent Privacy) protocol and the newer WiFi Protected Access (WPA) protocol.
- The RC4 algorithm is remarkably simple and quite easy to explain.

RC4 Algorithm

- A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0], S[1],..., S[255].

- At all times, S contains a permutation of all 8-bit numbers from 0 through 255. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of k is generated, the entries in S are once again permuted.

Initialization of S & key bytes

- To begin, the entries of S are set equal to the values from 0 through 255 in ascending order; that is; S[0] = 0, S[1] = 1,..., S[255] = 255.

- A temporary vector, T, is also created. If the length of the key K is 256 bytes, then K is transferred to T. Otherwise, for a key of length keylen bytes, the first keylen elements of T are copied from K and then K is repeated as many times as necessary to fill out T. These preliminary operations can be summarized as follows:

```
/* Initialization */
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

Initial permutation of S

- Next we use T to produce the initial permutation of S. This involves starting with S[0] and going through to S[255], and, for each S[i], swapping S[i] with another byte in S according to a scheme dictated by T[i]:

```
/* 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]);
```

- Because the only operation on S is a swap, the only effect is a permutation. S still contains all the numbers from 0 through 255.

Key generation and Encryption

- Once the S vector is initialized, the input key is no longer used. Stream generation involves cycling through all the elements of S[i], and, for each S[i], swapping S[i] with another byte in S according to a scheme dictated by the current configuration of S. After S[255] is reached, the process continues, starting over again 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]);
```

Key generation and Encryption

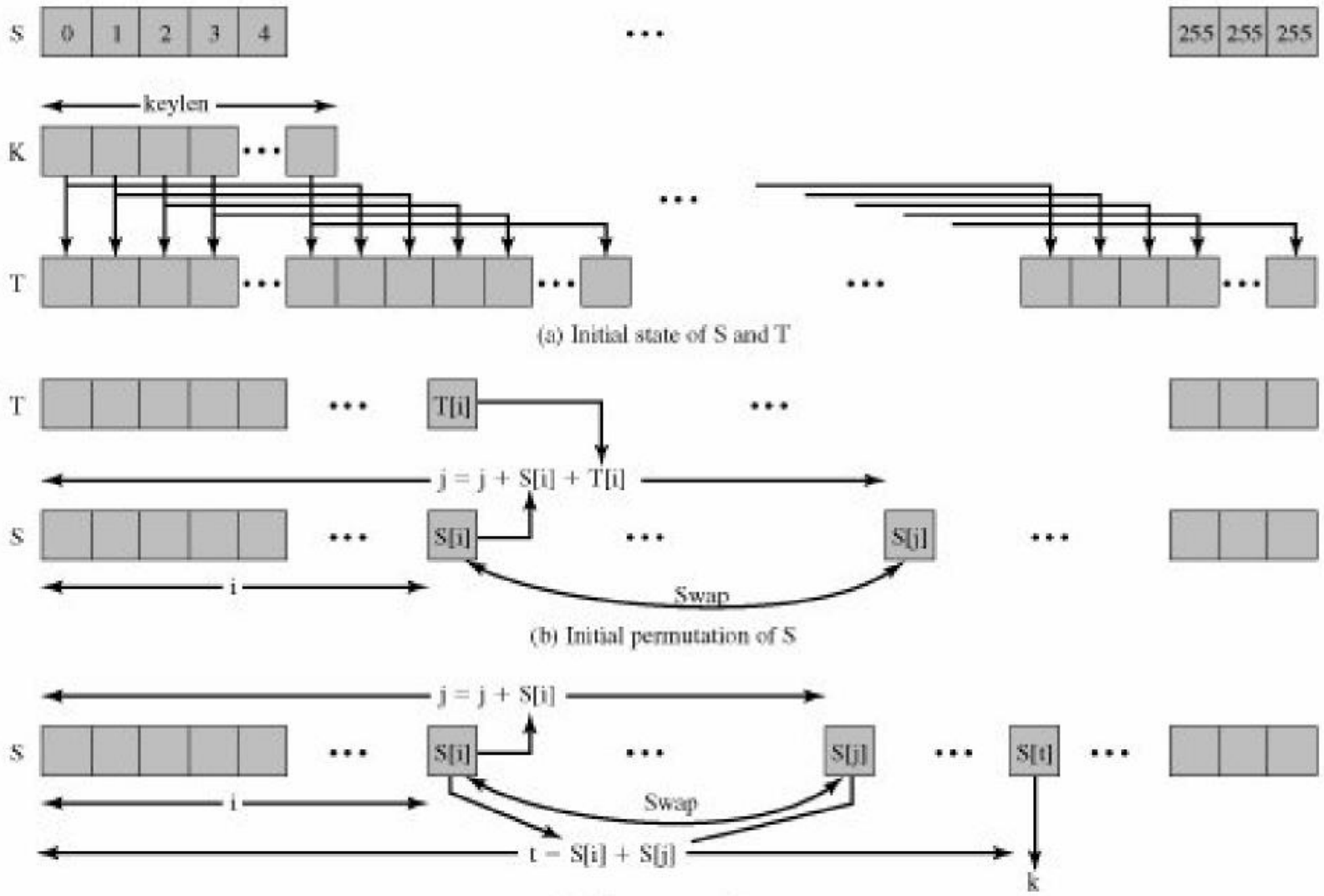
t = (S[i] + S[j]) **mod 256;**

k = S[t];

C = p $\oplus$ K

}

Stream Generation



Strength of RC4

- A number of papers have been published analyzing methods of attacking RC4 for small key sizes (less than 5 bytes). None of these approaches is practical against RC4 with a reasonable key length, such as 128 bits.

- The problem is not with RC4 itself but the way in which keys are generated for use as input to RC4.

A5 I

- A5/1 was developed by ETSI (European Telecommunications Standards Institute) for use in Eastern European states that had restrictions to certain Western technologies. in 1987, method is initially kept secret but became public knowledge through leaks and reverse engineering. The general design was leaked in 1994.

- The algorithms entirely reverse engineered by Briceno in 1999 from a mobile phone. In 2000, had around 130 million of GSM users . Nowadays, 4.8 billion wireless connections use G.S.M (A5/1 &A5/2).

- A5/1 creates a bit stream out of a 64 bit key. The bit steams are collected in a 228 bit buffer to be exclusive ORed with a 228 bit frame.

A5 II

- A5/1 is a stream cipher used to provide over-the-air communication privacy in the Global System for Mobile Communication (GSM) cellular telephone standard. It is one of seven algorithms which were specified for GSM use.

- A number of serious weaknesses in the cipher have been identified, some attacks require an expensive preprocessing stage after which the cipher can be broken in minutes or seconds. the weaknesses have been passive attacks using the known plaintext assumption. In 2003, more serious weaknesses were identified which can be exploited in the ciphertext-only scenario.

- A5/2 Algorithm used in the GSM ciphering process between a MS (Mobile Station) and the GSM network. This algorithm is simpler than A5/1

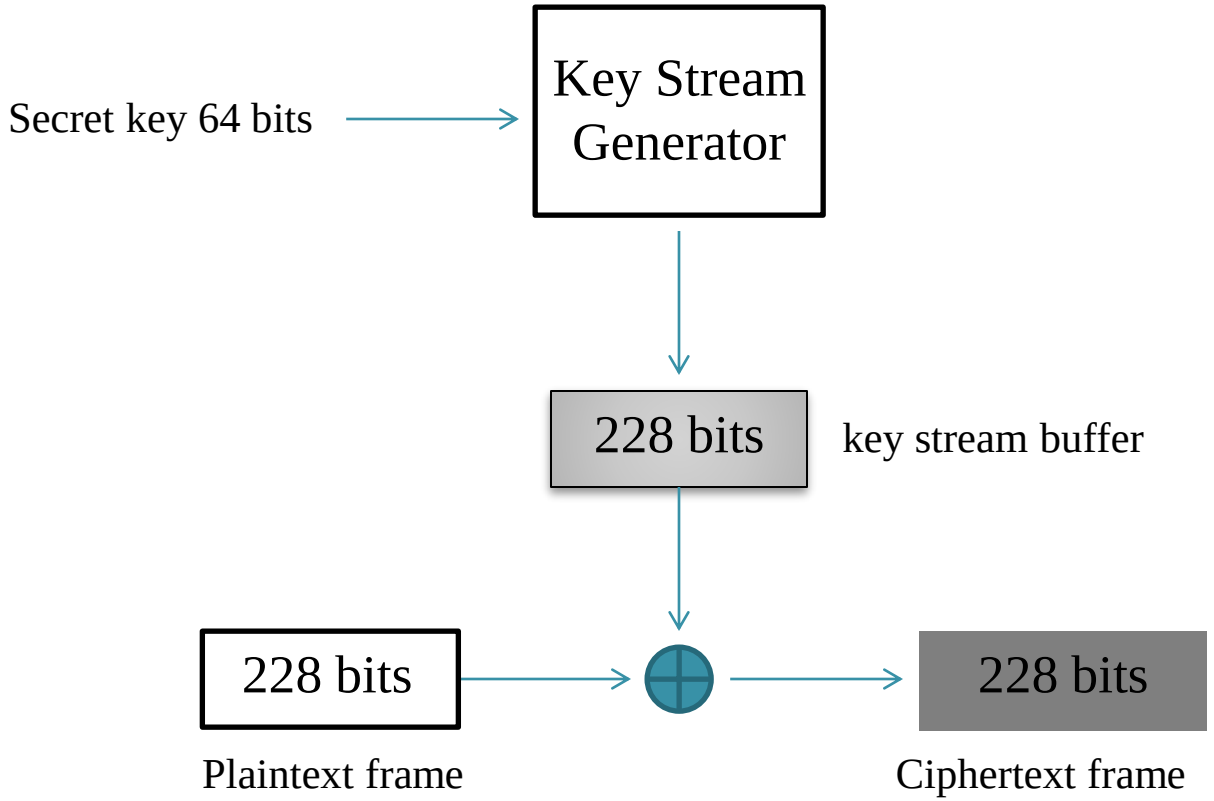
A5 III

- A5/2 is a stream cipher used to provide voice privacy in the GSM cellular telephone protocol. It was used for export instead of the relatively stronger (but still weak).

- The A5/3 encryption system – known as KASUMI - the Japanese word for "mist" - is the upgrade to A5/1 and uses a block cipher. A5/1 is designed to be used for the GSM network, whereas A5/3 is for 3GPP, and is based on the MISTY1 cipher created by Mitsubishi, but was modified to reduce processing restrictions on mobile devices.

- Key should be 32 characters 128 bits and data should be in blocks of 64-bits.

A5/1



Encryption