The RC4 Stream Cipher Algorithm

- As mentioned earlier, a key component of a stream cipher is the pseudorandom byte sequence generator.
- We will now go through the **pseudorandom byte sequence generator** in the RC4 algorithm.
- RC4 is a **variable key length** stream cipher with byte-oriented operations. It produces a pseudorandom **byte** stream.
- Fundamental to the RC4 algorithm is a 256 element array of 8-bit integers. It is called the **state vector** and denoted S .
- Theoretical analysis shows that for a 128 bit key length, the **period** of the pseudorandom sequence of bytes is likely to be greater than 10^{100}.

Some Applications of RC4

- RC4 is used in the **SSL/TLS** (Secure Socket Layer / Transport Layer Security) standard for secure communications between web browsers and web servers.

- RC4 is also used in the **WEP** (Wired Equivalent Privacy) protocol and the newer WiFi Protected Access (**WPA**) protocol that are part of the IEEE 802.11 wireless LAN standard.

RC4 state vector initialization (1)

- The state vector is initialized with the encryption key. The exact initialization steps are as follows:
- The state vector S is initialized with entries from 0 to 255 in the ascending order. That is

$$
\begin{aligned}
S[0] &= 0x00 = 0 \\
S[1] &= 0x01 = 1 \\
S[2] &= 0x02 = 2 \\
S[3] &= 0x03 = 3 \\
&\ldots\ldots \\
&\ldots\ldots \\
S[255] &= 0xFF = 255
\end{aligned}
$$

RC4 state vector initialization (2)

- The state vector *S* is further initialized with the help of another temporary 256-element vector denoted *T*. This vector also holds 256 integers. The vector *T* is initialized as follows:

1. Let's denote the encryption key by the vector *K* of 8-bit integers. Suppose we have a 128-bit key. Then *K* will consist of 16 non-negative integers whose values will be between 0 and 255.

RC4 state vector initialization (3)

2. We now initialize the 256-element vector *T* by placing in it as many repetitions of the key as necessary until *T* is full. Formally,

$$T[i] = K[i \bmod keylen] \qquad for\ 0 \leq i < 255$$

where *keylen* is the number of bytes in the encryption key. In other words, *keylen* is the size of the key vector *K* when viewed as a sequence of non-negative 8-bit integers.

RC4 state vector initialization (4)

- Now we use the 256-element vector *T* to produce the initial permutation of *S* . This permutation is according to the following formula that first calculates an index denoted *j* and then swaps the values *S* [*i*] and *S* [*j*] .This algorithm is generally known as the Key Scheduling Algorithm (KSA) :

```
j   =   0
for i = 0 to 255
     j   =   ( j + S[i] + T[i] )   mod   256
     SWAP   S[i], S[j]
```

RC4 state vector initialization (5)

- There is no further use for the temporary vector *T* after the state vector *S* is initialized as described above.
- Note that the encryption key is used only for the initialization of the state vector *S* . It has no further use in the operation of the stream cipher.
- Note also that initialization procedure for the state *S* is just a permutation of the integers from 0 through 255. Each integer in this range will be in one of the elements of *S* after initialization. This happens because all that the initialization does is to swap the elements of *S* according to the secret key.

RC4 keystream Generation (1)

- After the state vector *S* is initialized, the pseudorandom byte stream is generated from the state vector.
- The following procedure generates the pseudorandom byte stream from the state vector

```
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]
     output k
```

RC4 keystream Generation (2)

- Note how the state vector *S* changes continuously by the swapping action at each pass through the while loop.
- The above procedure spits out values of the variable k for the pseudorandom byte stream. The plaintext byte is XORed with this byte to produce an encrypted byte that is transmitted to the destination.
- The pseudorandom sequence of bytes generated by the above algorithm is also known as the keystream.

Strengths and Weaknesses of the RC4 Stream Cipher

- Its simplicity is its greatest asset.
- Because all operations are at the byte level, the cipher possesses fast software implementation. For that reason, RC4 remains the mostly widely used software stream cipher.
- Although it is still used extensively, RC4 is not considered to be sufficiently secure any longer. It was shown that RC4 possesses a large number of weak keys. With these keys, a knowledge of just a few key bits can be used to make strong inferences about the state vector used for encryption.

Finally . . .

- Acknowledgment: These lecture notes are based on the textbook by William Stallings and notes prepared by Avinash Kak, Purdue University. My sincere thanks are devoted to them and to all other people who offered the material on the web.

- Students are advised to study and solve the problems and answer the questions in **Assignment-A1**.