Lecture Three

Stream Cipher and Shift Register

1. Introduction

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. The main properties of stream ciphers separating them from block ciphers are that the encryption function works on individual symbols (letters) of the underlying alphabet and that the encryption function is time-varying.

Stream ciphers have extensive applications; many of them are in the area of wireless communication. As an example, they are part of the security framework in GSM networks, Bluetooth or WLANs.

Shift register sequences are used in both cryptography and coding theory. There is a wealth of theory about them; stream ciphers based on shift registers have been the workhorse of military cryptography since the beginnings of electronics.

In this chapter we attempt to give an importance of LFSR and how its derived from finite state machine, then introduce the LFSR concepts described by Golomb and Massey.

2. Modern Cryptosystems

There are essentially two different types of cryptographic systems (cryptosystems), these cryptosystems are described in the next two subsections.

2.1 Public Key Cryptosystems

First let us redefined some important notations:

- **Key space K**: a set of strings (keys) over some alphabet, which includes the encryption key e_k and the decryption key d_k.
- The **Encryption** process (algorithm) E: E_{ek} (M) = C.
- The **Decryption** process (algorithm) D: D_{dk}(C) = M.
- The algorithms E and D must have the property that:

  $Dd_k(C) = Dd_k(Ee_k(M)) = M$.

It's also called **asymmetric cryptosystems**. In a public key (**non-secret key**) cryptosystem (see figure (1)), the encryption key e_k and decryption key d_k are different, that is $e_k \neq d_k$.

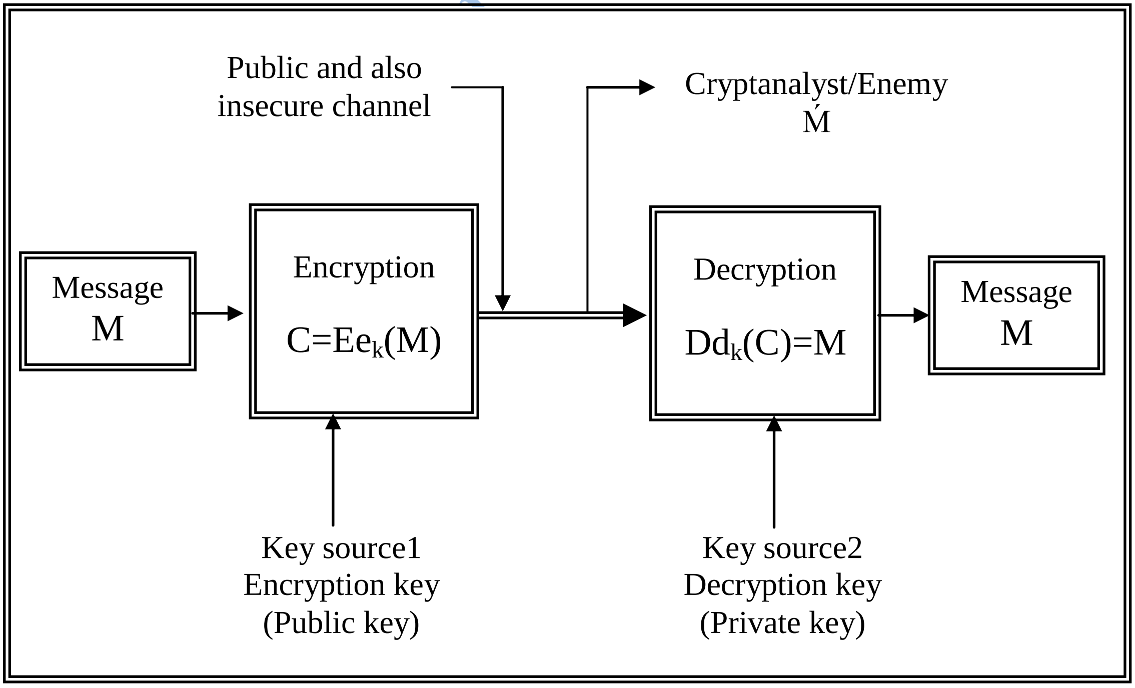

Figure (1) Modern Public-key Cryptosystem $e_k \neq d_k$.

2.2 Secret Key Cryptosystems

It's also called **symmetric cryptosystems**. In a conventional secret-key cryptosystem (see figure (2)), the same key ($e_k=d_k=k\in K$), called **secret key**, used in both encryption and decryption; we are interest in this type of cryptosystems.

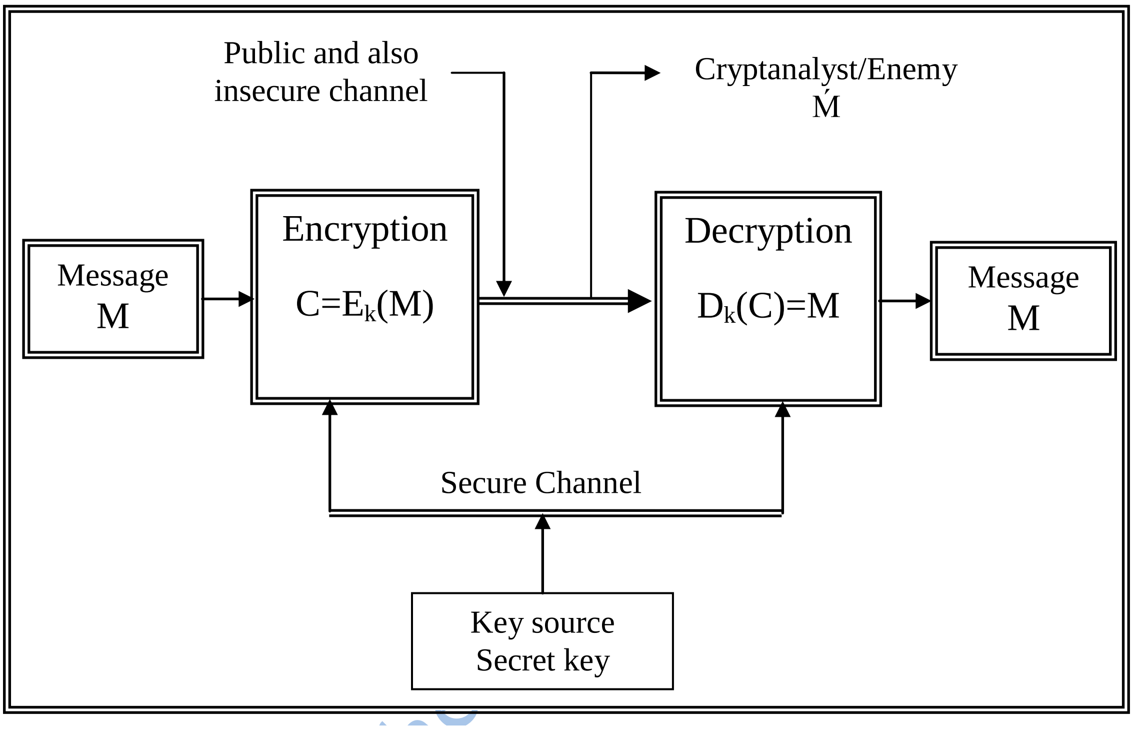

Figure (2) Conventional Secret-key Cryptosystems $e_k \neq d_k$.

The sender uses an invertible transformation f defined by:

$$f: M \xrightarrow{k} C$$

So produce the ciphertext:

$C = (E_k(m))$, $m\in M$ and $c\in C$.

and transmits it over the public insecure channel to the receiver. The key k should also be transmitted to the legitimate receiver for decryption but via a secure channel since the legitimate receiver knows the key k, he can decrypt c by transformation f^{-1} defined by:

$$f^{-1}: C \xrightarrow{k} M$$

and obtain:

$D_k(c) = D_k(E_k(m)) = m$, $c \in C$ and $m \in M$,

and it’s the original plaintext message.

There are many different types of secret key cryptosystems, like monographic (character) ciphers, polygraphic (block) ciphers, exponentiation ciphers and stream (bit) ciphers in which we shall focus.

The Advantages of symmetric-key cryptography are:

1. Symmetric-key ciphers can be designed to have high rates of data throughput. Some hardware implementations achieve encrypt rates of hundreds of megabytes per second, while software implementations may attain throughput rates in the megabytes per second range.
2. Keys for symmetric-key ciphers are relatively short.
3. Symmetric-key ciphers can be employed as primitives to construct various cryptographic mechanisms including pseudorandom number generators, hash functions, and computationally efficient digital signature schemes, to name just a few.
4. Symmetric-key ciphers can be composed to produce stronger ciphers. Simple transformations which are easy to analyze, but on their own weak, can be used to construct strong product ciphers.