Encryption and Decryption

Let us explain main concepts and definitions.

Let's assume that *A* is a finite set, named alphabet. Then *M* is a set of words or strings consisting of elements of the alphabet *A*, we'll call it a set of plain messages, and elements of *M* will be named clear messages or simply plain texts.

Let's define a set of words or lines consisting of elements of the alphabet that differs from *A* in a general case as a set of encrypted messages *C*, and elements of such set will be named ciphertexts.

Let's assume that *K* is a set of keys. Each element *e* in the key set *K* determines the encryption function E_e, that unambiguously maps the set *M* onto the set *C*, i.e. E_e is a bijection.

Each key *d* in the key set *K* determines the decryption function D_d, that maps the set *C* onto the set *M* and is also a bijection.

Let's designate application of the encryption function E_e to a plain message *m* from the set *M* as encryption, while application of the decryption function D_d to a ciphered message *c* from the set *C* as decryption.

An encryption scheme or cipher consists of the set of encryption functions $\{E_e : e \in K\}$ and their corresponding decryption functions $\{D_d : d \in K\}$ with the following feature: for each key *e* from the set *K* a key *d* exists from the same set *K*, so that $D_d = E_e^{-1}$, i.e. $D_d(E_e(m))=m$ for each plain message *m* from the set *M*.

Let's designate the pair of keys *e* and *d* as a key pair. It should be also noted that they may be identical.

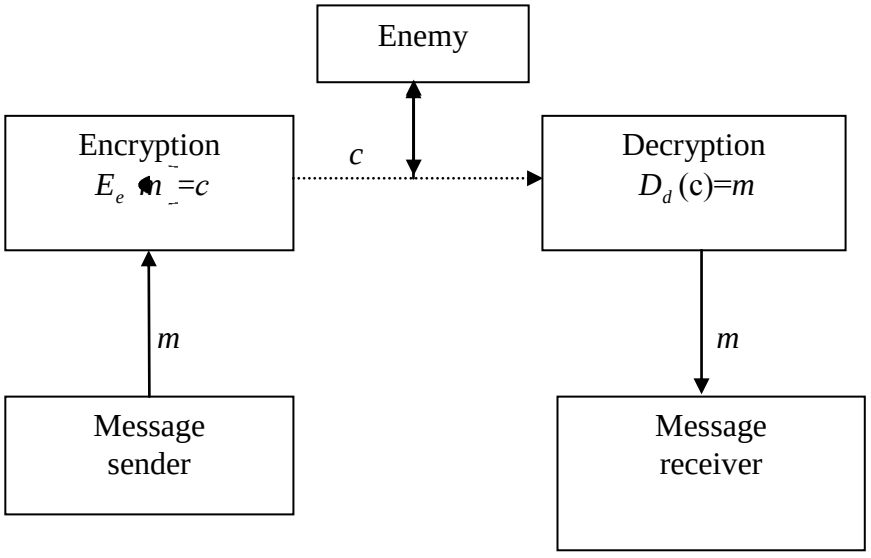

Fig. 1. Two-way communication circuit with encrypting capability.

A circuit is a means to transmit data between the parties. A circuit may be non-secure, physically protected (physically inaccessible for the third parties), and secure (no third party is able to alter or read data transmitted via the circuit).

When analyzing algorithms used and estimating communications security, it is assumed that the third parties do know the sets *M, C, K,* $\{E_e : e \in K\}$, $\{D_d : d \in K\}$. Only a specifically selected pair of keys (*e, d*) is confidential. Sometimes the encryption function is also confidential, however security of data transmission should not be based on that.

Cryptographic algorithms designed to ensure communications security are usually divided into two large classes: symmetric-key ciphers and public-key ciphers.

Symmetric-Key Encryption Algorithms

Symmetric-key cipher is a cipher with a set of encryption functions $\{E_e : e \in K\}$ and a set of decryption functions $\{D_d : d \in K\}$ in respect of which the following condition is met: for any key pair $(e, d) \in K$, d can be easily computed with known e, and vice versa. For the majority of symmetrical ciphers $e=d$. Such encryption scheme is also called a one-key scheme, or conventional encryption.

Symmetric-key ciphers are classified into two large groups – block ciphers and stream ciphers.

Symmetric-Key Stream Ciphers

Stream ciphers are sometimes called status ciphers, because encryption depends not only on the key and plain text, but on the current status as well. An important feature of stream ciphers is limited error propagation.

If K is a key space for a set of encryption functions, then sequence $e_1 e_2 \dots e_i \in K$ is called a keystream. A keystream can be randomly generated from an initial small keystream with the use of a certain algorithm, or from an initial keystream with the use of previous ciphertext characters. Such algorithm is called keystream generator.

In a general case, stream ciphers are faster than hardware block ciphers, they do not propagate errors and are easy to analyze. Despite the fact that a large amount of literature exists on analysis of stream ciphers, and their construction principles have been described, there are few complete descriptions of stream ciphers in the literature. Conversely, considerable number of block ciphers has been published, and some of them have been accepted as a national standard.

The simpliest stream cipher can be illustrated by a binary alphabet cipher, where $m_1 m_2 \dots m_t$ is a binary plain text, $c_1 c_2 \dots c_t$ is a binary cipher text, $k_1 k_2 \dots k_t$ is a key stream, and

$$c_i = m_i \oplus k_i \text{ for } i=\overline{1,t}$$

Here, $\oplus$ means modulo 2 addition or XOR.

If a keystream is generated independently and randomly, such cipher is called a one-time pad. It has been confirmed that such cipher, if properly used, can be absolutely resistant. Its main disadvantage lies in necessity to generate a key stream length equal to the length of a plain text. In addition, this requires that the key sequence be transmitted to the receiving party via an absolutely secure circuit, which is not acceptable when large amounts of data are to be transmitted.

Stream ciphers are classified into synchronous and self-synchronizing.

Synchronous stream ciphers are those ciphers that generate a key stream independently of plain and cipher texts. Encryption process for such ciphers can be described with the following formulas:

$$s_{i+1} = f(s_i, k),$$
$$z_i = g(s_i, k),$$
$$c_i = h(z_i, m_i),$$

where s_0 is an initial status determined by the key k, f is a function to compute the next status, g is a function to generate the keystream z_i, h is an output function that computes a ciphertext character c_i from a plaintext character m_i and a keystream character z_i.

Synchronous stream ciphers have the following features:

- both sender and receiver of a message must be synchronized in order to decrypt the cipher text properly (if a ciphertext character is additionally added

or lost in the communications circuit, the synchronization will be lost, and special arrangements will be required to restore it by insertion of equally spaced special markers in the cipher text, etc.);
- they do not propagate errors.

The majority of published stream ciphers are additive stream ciphers: plain and cipher texts, as well as the key stream, constitute binary sequences, and the output function *h* is a modulo 2 addition.

Self-synchronizing or asynchronous stream ciphers are defined as the stream ciphers whose key stream is a function of the key and fixed number of previous ciphertext characters. Encryption function for an asynchronous stream cipher can be described with the following formulas:

$$s_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}),$$
$$z_i = g(s_i, k),$$
$$c_i = h(z_i, m_i),$$

where $s_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$ is non-secure initial vector, *k* is a key, *g* is a function to generate the keystream z_i, *h* is an output function.

Asynchronous stream ciphers have the following features:

- when a character is additionally inserted or lost, self-synchronization takes place, since decryption of a character depends on fixed number of previous characters only (proper decryption is restored after loss of synchronization, with the fixed number of plain text characters remaining undeciphered);
- error propagation is limited;
- alteration of ciphertexts by the third parties is difficult to find out, therefore additional means are required to check message integrity and perform authentication;
- asynchronous stream ciphers are more resistant to attacks based on the use of plain text statistics.

When constructing the majority of published stream ciphers, linear and non-linear feedback shift registers are widely used. The reasons for that are as follows:

- hardware implementation of feedback shift registers is very easy;
- feedback shift registers allow to obtain sequences of greater periods with good statistical properties;
- feedback shift registers are easy to analyze with the use of algebraic methods.

Unfortunately, a sequence obtained with the use of a linear feedback register is determined completely by the initial state and polynomial of this register. Therefore, when constructing stream ciphers, no linear feedback registers are used in pure form, but non-linear functions are utilized instead, whose arguments are values of output functions of several linear feedback registers or values of stages from a single register. In addition, schemes in which one linear feedback register rules other registers are utilized. These methods allow to obtain better linear complexity of the cipher constructed, larger period and good statistical properties of the output sequence.

When using linear feedback registers, either initial state of registers or both initial status of registers and register feedback factors can be a cipher key.

The disadvantage of stream ciphers constructed with the use of linear feedback registers is that their hardware implementation is not easy. That is why steam ciphers specialy designed for software implementation were developed recently.

Symmetric-Key Block Ciphers

Symmetric-key block ciphers are widespread now and form an important part of many cryptographic systems. They are used to ensure data transmission confidentiality, provide a basis

for pseudorandom number generators, stream ciphers and hashing functions. They play an important role in the methods of authentication, data integrity and digital signature. Such ciphers as DES, FEAL, IDEA, GOST 28147-89 and RC5 belong to this class of ciphers.

A block cipher is defined as the cipher whose encryption function maps *n* bits of a plain text to *n* bits of a cipher text. Number *n* is defined as the block length, or block size. A block cipher may be considered as a substitution cipher that uses a large alphabet. The encryption function is determined by *l*-bit key *k*, selected from the key set *K* that is a subset of the set V_l of all binary *l*-bit vectors. In this case the encryption function must be a bijection. Therefore, *n*-bit block cipher is a function $E: V_n \times K \to V_n$, such that for any *k* from the set *K*, $E_k(P)$ is a reversible function (encryption function for the key *k*) that maps V_n to V_n. The inverse function, that is called the decryption function, is designated as $D_k(C)$. $C = E_k(P)$ means that the ciphertext *C* has been obtained as a result of encryption of the plaintext *P* with the use of the key *k*.

Four main modes of operation exist for the block ciphers.

In the ECB mode, a plaintext is divided into blocks of length *n*, and each block is encrypted independently with the use of the encryption function *E*, i.e. encryption equation would be $c_j = E_k(x_j)$, while decryption function would be $x_j = E_k^{-1}(c_j)$, where $1 \le j \le t$, and x_j is a block of the plaintext.

The ECB mode has the following features:

- identical blocks of a plaintext are converted into identical blocks of a ciphertext;
- blocks are encoded independently of one another, and any change in the order of blocks in the plaintext results in the same change in the order of blocks in the ciphertext;
- alteration of one or more bits in the ciphertext results in improper decryption of only one given block.

The ECB mode is not recommended for encryption of a plaintext whose length is more that the length of a block or when the key is used more than once.

The CBC mode uses *n*-bit initial vector *IV*. In this mode, encryption and decryption equations are as follows:

$$c_0 = IV$$
$$c_j = E_k(c_{j-1} \oplus x_j)$$
$$x_j = c_{j-1} \oplus E_k^{-1}(c_j).$$

The CBC mode has the following features:

- identical ciphertexts are produced when the same plaintexts are encoded using the same key and with the same initial vector;
- modification of the initial vector, key or the first block (for example, when a counter or random numbers are used in the beginning of the plaintext) results in alteration of the ciphertext;
- a ciphertext block *c* depends on *x* and all previous blocks of the plaintext, therefore, any alteration in the order of ciphertext blocks results in improper decryption.

The CFB mode is a feedback mode. In this case, a plaintext $x_1, \ldots, x_u$ is divided into blocks of length *r* ($1 \le r \le n$), the initial vector is a *n*-bit vector. The encryption function is as follows:

$$I_1 = IV,$$
$$O_j = E_K(I_j) \text{ for } 1 \le j \le u,$$

t_j = least significant bits O_j,

$c_j = x_j \oplus t_j$,

$I_{j+1} = 2^r \cdot I_j + c_j \mod 2^n$.

Decryption process can be described as follows:

$I_1 = IV$

$x_j = c_j \oplus t_j$, where t_j, O_j, I_j have been described above.

The CFB mode has the following features:

- alterations in *IV* result in different ciphertexts when identical plaintexts are encoded;
- since a ciphertext block c_j depends on plaintext blocks x_j and x_{j-1}, any alteration in the order of ciphertext blocks affects decryption;
- previous $\lceil n/r \rceil$ ciphertext blocks must not be corrupted, in order that a ciphertext block can be decrypted correctly;
- alteration of one or more bits in any r-bit block of a ciphertext c_j affects decryption of consecutive $\lceil n/r \rceil$ ciphertext blocks;
- the CFB mode is a self-synchronizable mode, however $\lceil n/r \rceil$ ciphertext blocks are required for self-synchronization.

The OFB mode is also a feedback mode. In this case, a plaintext $x_1, \ldots, x_u$ is divided into blocks of length r ($1 \le r \le n$), the initial vector is a n-bit vector.

The encryption function is as follows:

$I_1 = IV$,

$O_j = E_K(I_j)$ for $1 \le j \le u$,

$t_j = r$ left bits O_j

$c_j = x_j \oplus t_j$,

$I_{j+1} = O_j$

$I_{j+1} = 2^r I_j + t_j \mod 2^n$.

The OFB mode has the following features:

- alterations in *IV* result in different ciphertexts when identical plaintexts are enncrypted;
- the key stream does not depend on the plaintext;
- one or more erroneous bits in a ciphertext character result in improper decryption of the corresponding plaintext character;
- the OFB mode is able to restore its proper functioning unless ciphertext bits are skipped.

A strong block cipher must possess properties o confusion and diffusion.

Diffusion is defined as a cipher property that lies in the fact that one bit of a plaintext influences a few bits of a ciphertext. Therefore, encryption of two plaintext blocks that vary little one from another results in different blocks of the ciphertext. The same condition must be met in respect of ciphertext dependence on the key, i.e. one bit of the key must influence all bits of the ciphertext.

Confusion is defined as ability of a cipher to hide interrelations between plaintext characters and ciphertext characters, i.e. a ciphertext must not contain any statistical dependencies inherent to a plaintext.

Asymmetrical encryption algorithms, or public-key algorithms, form another major class of cryptographic algorithms.

Asymmetric Encryption Algorithms, Or Public-Key Algorithms

Public-key cryptography has appeared fairly recently. In 1976, American cryptologists W.Diffy and M.Hellman published a number of papers describing certain public-key ciphers. Public-key encryption schemes has gained wide grounds since then.

Assume that $\{E_e: e \in K\}$ is a set of encryption functions and $\{D_d: d \in K\}$ is a set of decryption functions, where K is a set of keys. Let's take a pair of functions (E_e, D_d) and suppose that each pair of such functions has the following propety: with a knowledge of E_e and ciphertext $c \in C$, it is impossible in acceptable period of time to find out the plaintext m, such that $E_e(m)=c$. It means that a decryption key d is hard to determine using an encryption key e. Encryption functions E_e belong to the so called trapdoor one-way functions.

Two-parties public key communication would be as follows. One side selects a pair of keys (e, d) and sends a key e, called a public key, to the other side via a non-secure circuit. As this takes place, a key d remains confidential. The other side sends a message m, encrypteded with the key e, to the first side. The first side deciphers m using its private key d. Therefore, any subscriber in the public key communications network can transmit a message to the first side, using the non-confidential public key e. And it can only be deciphered by a subscriber that possesses the private key d.

There are several types of trapdoor one-way functions that are used in various public key encryption schemes. All of them are based on number-theoretic problems, such as factorization of large numbers, taking the logarithms in finite fields, etc.

Advantages And Disadvantages Of Symmetrical Ciphers

Symmetrical ciphers have the following advantages:

- there exist hardware implementations of symmetrical ciphers with high encrypting rates (up to several hundred Mbit/sec);
- key length of symmetrical ciphers is relatively short;
- symmetrical ciphers can be used as a main component when constructing pseudorandom number generators, hashing functions and digital signature schemes;
- simple symmetrical ciphers can be used to create more resistant ciphers.

Symmetrical ciphers are characterized by the following disadvantages:

- for two-parties communications, a private key is required at both ends of the communication line;
- a large number of key pairs is required for communications networks with a large number of subscribers, which complicates seriously key management process or requires an additional third party completely trusted by the subscribers;
- the keys must be changed as frequently as possible (it is desirable that a new key be used for each communications session);
- excessively long keys are usually required for an algorithm to create a digital signature with the use of symmetric-key ciphers.

Advantages And Disadvantages Of Public key Cryptography

Public key cryptography has the following advantages:

- only a private key must be confidential;
- although a third party is required to manage keys, such third party does not need to know private keys of communicating parties;
- a key pair consisting of one private and one public key may remain unchanged for a long period of time (up to several years);

- public-key ciphers can serve as a basis for digital signature schemes, the key length in this case would be usually much less than the key length used in symmetric-key cipher schemes;
- much fewer keys are required for communications networks with a large number of subscribers than for symmetrical cipher networks.

At the same time, public key ciphers have the following disadvantages:

- encryption rate of all public-key ciphers is several orders lower than that of the best symmetric-key ciphers;
- when a public-key scheme is used, key length and signature length is much greater than in the case when a symmetric-key cipher is used;
- strength of symmetric-key ciphers used can be theoretically proved, while the strength of public-key ciphers is based today exclusively on solution intricacy of a few number theory problems.

Therefore, both symmetric-key ciphers and public-key cipher have their own advantages and disadvantages. Symmetric-key ciphers are usually used to encrypt data and ensure its integrity, while public-key ciphers are used in digital signature and key management schemes (for example, when dispatching the keys for symmetrical ciphers). Public-key ciphers are also used to ensure integrity of transmitted data, authentication and encryption of short messages (credit card numbers, PIN codes, etc.).