

AN ELEMENTARY INTRODUCTION TO CRYPTOGRAPHY

Cryptography is a science of secret writing. It is probably as old as the need of a mankind to communicate, especially if the information was intended for a limited number of users only. The counterpart is **cryptoanalysis** – a science of “cipher breaking”. Even some simple ciphers were used a few thousand years ago, **cryptology** (including cryptography as well as cryptoanalysis) was theoretically established in the XIX century. The birth of an electrical telegraph and its intensive use (wire or radio) later at the first part of XX century induced the design on many mechanical and electromechanical encryption/decryption devices. The next important point is a landmark paper by Shannon (1949) when “cryptology from an art becomes a science”. The further technology advance enabled an enormous development of communications as well as of computers. Today, cryptology is not a domain of interest of state government and army only. Cryptography is used by the big companies as well as by the single persons. This fact implies a redefinition of some cryptography classical requirements as well as an introducing of a new ones (sometimes “unthinkable” from the classical cryptography point of view). The last “milestone” is **public-key** systems. In Fig. 1 block diagram of a secret system (using a secret key) is shown.

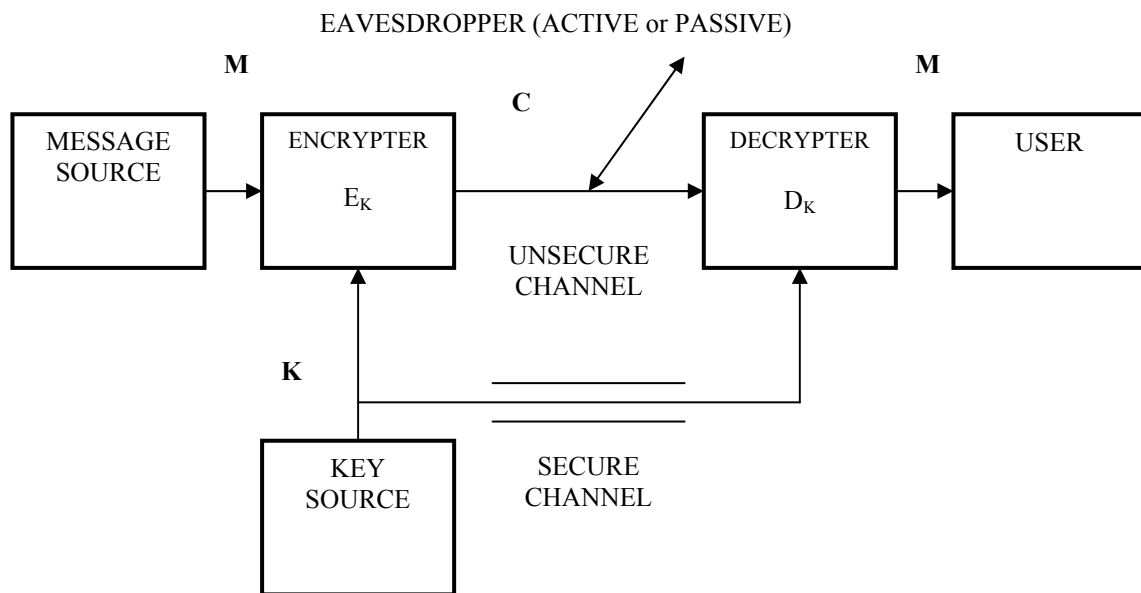


Fig. 1 Secret key cryptographic system

The message from a MESSAGE SOURCE (**M** – plain text) is transformed into a cryptogram (**C**) in the ENCRYPTER using algorithm **E_K** (on the basis of the key **K** obtained from the KEY SOURCE). The cryptogram is transmitted by the UNSECURE CHANNEL, where the EAVESDROPPER is present. He may be PASSIVE (i.e. to listen only) or ACTIVE (trying to insert some “false” cryptograms). The DECRYPTER, knowing the key **K** (obtained from the KEY SOURCE via a SECURE CHANNEL), using algorithm **D_K** reconstructs the original message **M** from the cryptogram **C**. It should be noted that the same algorithm could use more pairs of keys. The keys of one pair are practically either the same or can be easily derived one from the other.

Therefore, such a system has the following constituents:

- set of messages M ;
- set of cryptograms C ;
- set of keys K ;
- set of encrypting algorithms E (E_K uses the key $K \in K$);
- set of decrypting algorithms D (D_K uses the key $K \in K$);

Cryptoanalytical “attacks” can be classified according to the information available to cryptanalyst:

- ***ciphertext only attack***, one or more cryptograms are known, where the same key is used;
- ***known plaintext attack***, one or more pairs message-cryptogram are known, where the same key is used;
- ***chosen plaintext attack***, cryptanalyst can obtain more cryptograms for any messages according to his choice, where the same key is used.

It should be noted that some requirements for computer networks are quite different from the requirements in “classical” cryptography. The basic contemporary requirements are:

- encryption and decryption algorithms should be efficient for all the keys (key pairs);
- the encryption procedure should be “user friendly”;
- security (secrecy, authenticity, data integrity) should depend on the secrecy of the key and not on the encryption-decryption algorithm.

Secrecy and authenticity, symmetric and asymmetric cryptosystems

Secrecy requires that a cryptanalyst is not able to determine plaintext from the intercepted cryptogram. The requirements can be formalized as follows:

- It should be computationally infeasible to systematically determine the decryption transformation (D_K), for the intercepted cryptogram C , even if the corresponding message M is known;
- It should be computationally infeasible to systematically determine message M from the intercepted cryptogram C .

Authenticity requires that a cryptanalyst is not able to substitute a false cryptogram C' for an authentic cryptogram C . The requirements can be formalized as follows:

- It should be computationally infeasible to systematically determine the encryption transformation (E_K), for the intercepted cryptogram C , even if the corresponding message M is known;
- It should be computationally infeasible to find systematically cryptogram C' such that $D_K(C')$ is a valid plaintext.

Security implies the protection of the transformation D_K (i.e. the decryption key). The transformation E_K can be revealed if it does not give away D_K . Authenticity implies the protection of the transformation E_K (i.e. the encryption key). The transformation D_K can be revealed if it does not give away E_K . These requirements are symbolically illustrated in Fig. 2.

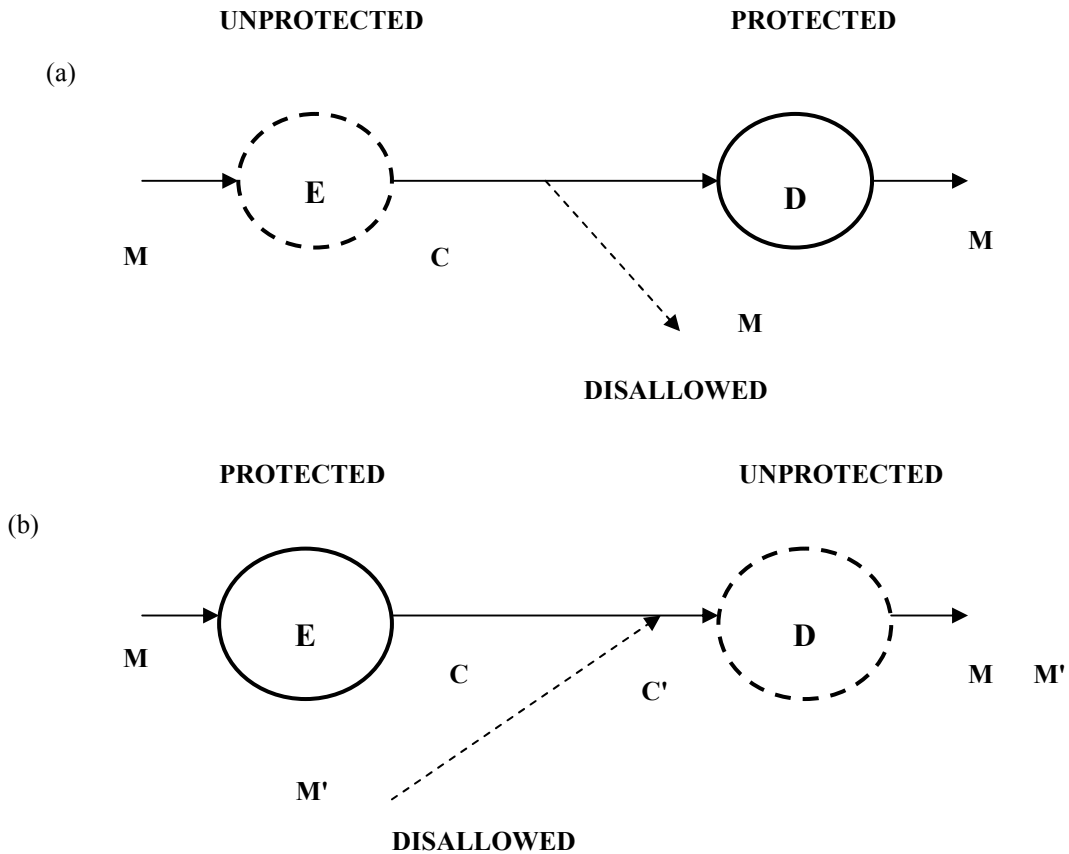


Fig. 2 Security (a) and authenticity (b)

This is the very place to comment the difference between symmetric and asymmetric systems.

In *symmetric (one-key)* cryptosystems the encrypting and decrypting keys are the same (or easily determined from each other). This means the transformations E_K and D_K are also easily derived from each other. As it is assumed that the general method of encryption is known, both E_K and D_K should be protected to achieve both secrecy and authenticity (i.e. in such a system secrecy can not be separated from authenticity. One such well-known system is DES (*Digital Encryption Standard*)).

In *asymmetric (two-key)* cryptosystems the encryption and decryption keys differ in such a way that at least one key is computationally infeasible to determine from the other. Therefore, one of the transformations E_K or D_K can be revealed without endangering the other. It is a basis for *public-key encryption systems*. In such a system each user has both a public and private key. Two users can communicate knowing only each other's public keys. The public key can be registered with a "public directory" and only that user knows a private one. The keys are reciprocal (inverse) and can be used in any order to obtain the original message. Secrecy and authenticity are provided by the separate transformations. If the user A wishes to send a message to user B, the public key E_B is used, while the user B decrypts the message using his own private key (D_B). Authenticity is not provided because any other user could substitute another message replacing the cryptogram from A with the other one. Authenticity is provided if A uses his private key for encryption, while B, taking A's public key (from "directory"), decrypts the message. In such a way B "knows" that user A only could produce the cryptogram. Secrecy and authenticity are provided simultaneously using both procedures, as shown in Fig. 3. The public keys are denoted by E, the private keys by D, while the subscript denotes the user.

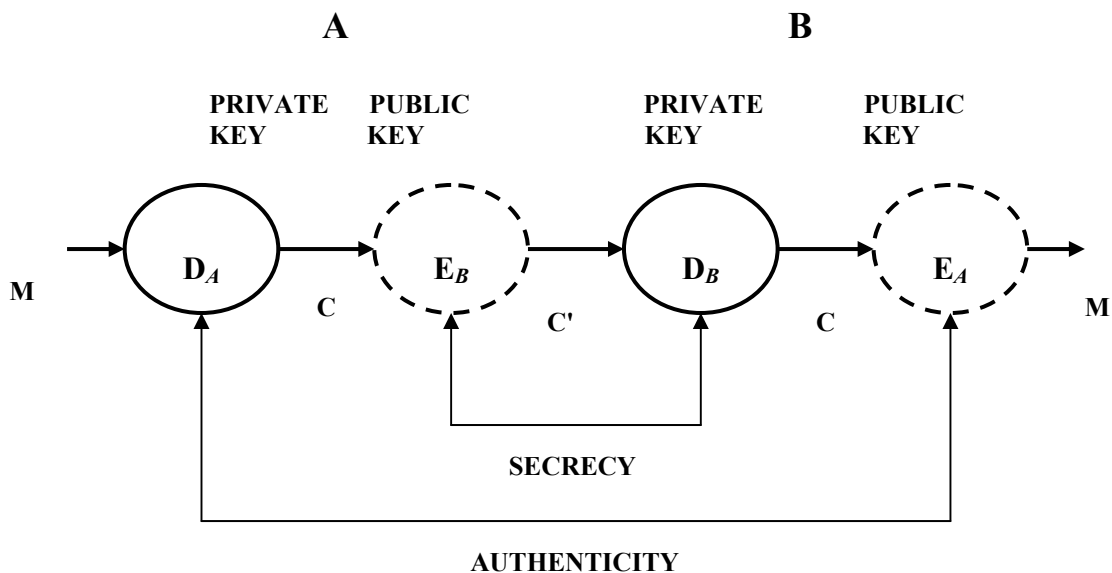


Fig. 3 Secrecy and authenticity in public-key system

Some simple encryption algorithms

In the same way as the error control codes are classified into block codes and convolutional codes, the ciphers may be classified into *block ciphers* and *stream ciphers*. When using the block cipher the message is divided into the blocks of a fixed length and every block is encrypted for itself. Stream cipher could be described as a block cipher having the block length equal one. The advantage with the stream cipher is that the encryption algorithm can be changed for every message symbol.

Transposition cipher rearranges symbols according to some scheme. The whole encryption process can be envisaged as writing the plaintext (message) into some geometric figure (often 2-dimensional array) according to some rule and reading the cryptogram according to some different rule.

Example:

- 1) word **COMMUNICATIONS** is divided into a four letter blocks according to *permutation* – for example, 3-1-4-2

COMMUNICATIONS
MCMOIUCNIAOT..

- 2) word **COMMUNICATIONS** is written into a 4×4 matrix by rows, the columns are taken off in the same order 3-1-4-2 (it is *transposition* of columns)

C O M M
U N I C
A T I O
N S . .
CUAN ONTS MIL. MCO.

The transposition (permutation) cipher can be recognized easily because the relative frequencies of the letters will be unchanged (i.e. they will be the same as in non-encrypted text). However, the conditional probabilities (letter after letter, letter after digram) will be quite different. If the original text is regarded as a Markov chain, then the cryptogram is generated by a source adjoined to the Markov source. The ciphers are broken by anagramming – by restoring a disarranged set of letters into their original positions. For such a procedure the relative frequencies of digrams and trigrams (for a corresponding language) will be helpful. For example, in English the following combinations appear often: *th, he, the*. In such a way a permutation period can be found.

The transposition of multidimensional array can be used as well, but it can be always reduced to the corresponding permutation of a one-dimensional array (vector).

The next step is a *single substitution cipher (monoalphabetic)*. The same alphabet is retained, but a simple rearrangement of the lexicographic order of letters is done.

Example:

The original alphabet: **A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

The cipher: **C F H Y P I T Q Z O J U W E A D K B R L N G M V X S**

The plaintext: **C O M M U N I C A T I O N S**

The cipher: **H A W W N E Z H C L Z A E R.**

Any possible permutation ($26!$) can be a cipher (key), the algorithm being a simple substitution. However, in “old times” it was not easy to remember the key, while it was forbidden to write it. In such a case some simple way to memorize the key was essential. One of the solutions was to denote every letter by the corresponding number ($A=0, B=1, \dots, Z=25$) and to substitute the letter denoted by i by the letter denoted by $i+n$. The addition of the indexes is carried out modulo the size of the alphabet (for the standard English mod 26). The well known is the *Caesar cipher* where $n=3$ (named after Julius Caesar).

Example:

The plaintext: **C O M M U N I C A T I O N S**
 Index (i): 2 14 12 12 20 13 8 2 0 19 8 14 13 18
 New index ($i+3$): 5 17 15 15 23 16 11 5 3 22 11 17 16 21
 Caesar cipher: **F R P P X Q L F D W L R Q V.**

The previous procedure can be interpreted as the letter **D** ($=3$) is used as a key:

The plaintext: **C O M M U N I C A T I O N S**
 Key: **D D D D D D D D D D D D D**
 Caesar cipher: **F R P P X Q L F D W L R Q V,**

where the corresponding indexes are added (modulo 26). To decrypt the key (**D**) is subtracted from the cryptogram.

A simple substitution cipher can be broken easily. To find a key, the single letter frequencies are needed only. The digram and trigram statistics can be used additionally to find the letter combinations characteristic for a language and to make a decision on the possible use of a permutation cipher (where the single letter statistics is the same, but the higher order entropies are not!). For the statistics used, it is expected to correspond to the nature of the encrypted text.

An interesting question is to find the length of the cryptogram allowing the finding of a key used – it is called the *unicity distance*. Shannon showed that for a standard English message, the unicity distance is about 27 letters – surprisingly small (the same being the case for the other languages as well).

More difficult to be broken is a *polyalphabetic substitution cipher*, meaning the successive use of different monoalphabetic ciphers for every letter of the message (it is usually periodic). Sometimes it is called *Vigener cipher*. Let the period of simple substitution be k (i.e. the simple substitutions n_1, n_2, \dots, n_k are used. They can be easily remembered as a corresponding word (we can call it “key”)

Example:

Use the Vigener cipher with the word **KEY** ($k=3$) ($K=10, E=4, Y=24$)
 The plaintext: **C O M M U N I C A T I O N S**
 Key: **K E Y K E Y K E Y K E Y K E**
 Cryptogram: **M S K W Y L S G Y D M M X W.**

To break such a cipher, the key period must be found. The cryprogram is searched for the identical trigrams (or longer blocks). Repetitions occur when a plaintext pattern repeats at a distance equal to a multiple of the key length.

And, lastly, as a key, the purely random series of letters can be used which length equals to the message length (*Vernam cipher*). If for the every new message the other key is used (*one-time pad*) then practically unbreakable cipher can be obtained. For such an approach a considerable memory resources are needed.

Such a cipher is often called *running-key*. There are various ways to generate a running-key (without the need to remember the whole key). A very simple approach is so called *autokey*. The method uses the original finite key at the beginning and thereafter uses the plaintext as a key. It should be noted that when using autokey, possible transmission errors will spread across the decrypted message, not being the case for some other encryption methods.

Example:

The plaintext: COM M U N I C A T I O N S
 Running key: KEY C O M M U N I C A T I .
 Cryptogram: M S K O I Z U W N C K O H B.

In the above it was implicitly supposed that some language and the corresponding alphabet were considered. However, in digital communications a binary alphabet (i.e. bits 0 and 1) is used. The length of a message may be very great. In this case the key is obtained using a corresponding generator (source). The “letters” (bits) of the message are added with the key modulo 2 at the transmitter (using an EXOR). The decryption is carried out at the receiver by adding (being the same as subtracting modulo 2) the key. Therefore, the hardware implementation is very simple. The corresponding procedure is shown in Fig. 4.

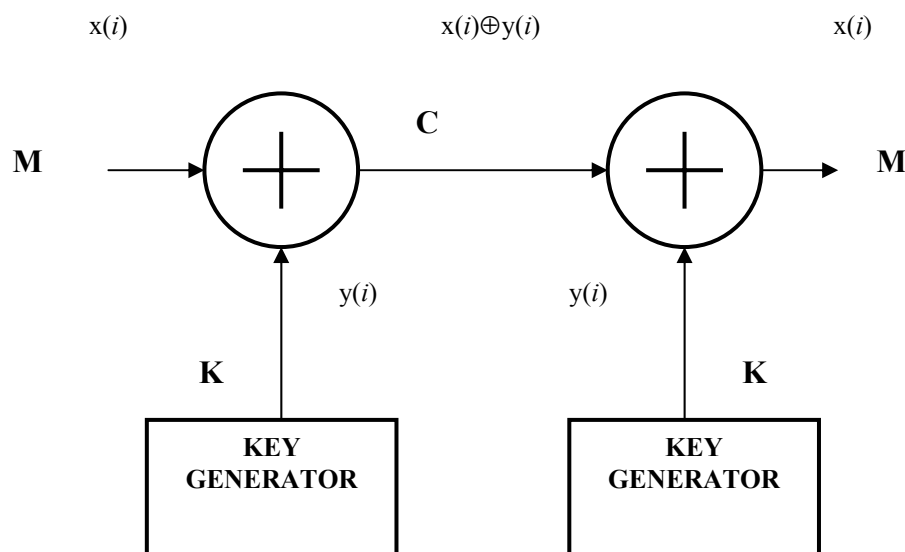


Fig. 4 Encryption and decryption in binary transmission

Such an approach is quite equivalent to the *scrambling* and *descrambling* procedures, being used very often for digital transmission. The transmitted digital signals (bits) are added modulo 2 usually with the *pseudonoise (PN) sequences*, being generated easily by linear shift registers. The scrambling allows an efficient transmission of the signals either obtained using A/D conversion from various sources (sensor, speech, image etc.) or the computer data using the same line independently of the signal statistics. The scrambling introduces the “randomness” into the transmitted signals allowing the “transparency” of the system for any original message. However, PN sequences are generated by the linear shift registers and the corresponding configurations are well known. Therefore, they cannot be used in cryptography. For encryption some other sequences, obtained by suitable decimation of PN sequences, or non-linear are used. Every such sequence is periodic (sometimes, depending on the transmission rate, period can be more years!). Obviously, the best key is a purely random binary sequence (statistically independent equally probable bits). But, this sequence cannot be obtained by a generator and must be memorized in advance.

In his fundamental paper “Communication Theory of Secrecy systems” (1949) defined the *perfect secrecy*. A necessary and sufficient condition for perfect secrecy is that the probability of receiving a particular ciphertext C given that message M was sent (encrypted under some key) is the same as the probability of receiving the same cryptogram C given that some other message M' was sent (encrypted under a different key). This means also that after receiving C , the aposterior probability that a particular message M has been sent is equal to the aprior probability that the same message M was sent. Our uncertainty (entropy) does not change with the receiving any cryptogram. In other words, any intercepted cryptogram does not change (increase) our knowledge about the key used. Further, it can be shown that Vernam cipher leads to the perfect secrecy.

Shannon also proposed the use of two encryption techniques to make the more difficult “statistical attacks” on cryptograms – confusion and diffusion. By *confusion* the substitutions are made yielding the relationship between the key and the cryptogram as complex as possible. By *diffusion* the suitable transformations spread the statistical properties of the message (plaintext) all over the cryptogram (for example, one letter “influences” all the letters of the ciphertext). He also proposed *product cipher*. While, the simple substitution and transposition does not provide a good secrecy, by their combining the good ciphers can be obtained. One of Shannon ideas led Hellman to invent public key systems. He said that the construction of a good cipher is essentially the question of finding a difficult mathematical problem, under some conditions. The cipher can be constructed in such a way that its breaking is equivalent to finding a solution of some problem known to be a difficult one.

DES (*Digital Encryption Standard*)

Among the block ciphers using the symmetrical keys DES is the most known standard. It started in 1977. in USA. It is the first contemporary algorithm (meaning the sequence of calculating – enciphering, deciphering) for a commercial use where all the details are specified and published (made known to all). It can be used for a secrecy as well as for an authenticity. Its design was carried out taking into account the corresponding technology state and anticipating the future technology development as well. Simply speaking, DES is a product cipher combining a series of successive substitutions and transpositions as well as some other simple operations (modulo 2 summation) to obtain a complex cipher. Obviously, the principles of confusion and diffusion are applied.

In Fig. 5 DES encryption algorithm is shown. The basic key consists of 64 bits, but the real length is 56 bits because 8 bits can be used as the parity checks. From the basic key auxiliary keys K_i ($i=1,2,\dots,16$) (their length being 48 bits) are calculated. The message to be encrypted is divided into 64 bit blocks. After the initial permutation (IP), these 64 bits are subdivided into two blocks consisting of 32 bits, left and right half (L_0 and R_0). Thereafter, 16 practically identical rounds of encryption follow using the keys K_i . During every round the right half is written into the left register and, simultaneously, using the corresponding key and the function f a block of 32 bits is generated, added modulo 2 with the left half and written into the right register. In fact,

$$L_i = \begin{cases} R_{i-1}, i = 1, 2, \dots, 15 \\ L_{15} \oplus f(R_{15}, K_{16}), i = 16 \end{cases}$$

and

$$R_i = \begin{cases} L_{i-1} \oplus f(R_{i-1}, K_i), i = 1, 2, \dots, 15 \\ R_{15}, i = 16 \end{cases}.$$

The unique exception is the last round, when the crosswise memorizing of the obtained 32 bit blocks is not performed. These two blocks are united, an inverse permutation (IP^{-1}) is performed resulting in a 64 bits cryptogram. The algorithm is complex (confusion) and every message bit influences on the every cryptogram bit (diffusion). The decryption is carried out in an identical manner, only the keys are used inversely (from K_{16} to K_1). All the operations are linear except when calculating the value of the function f . It is a symmetric cryptosystem (the key for encryption and decryption are identical!) and, consequently, it cannot be used as a public key system.

DES can be implemented both in software and in hardware. Besides modulo two addition (EXOR), the keys generation as well as the function f realization is very simple from the hardware's point of view and there are chips supporting the transmission rates of about 1 Gb/s. Software realization is usually based on the look-up tables, but the rates are substantially smaller.

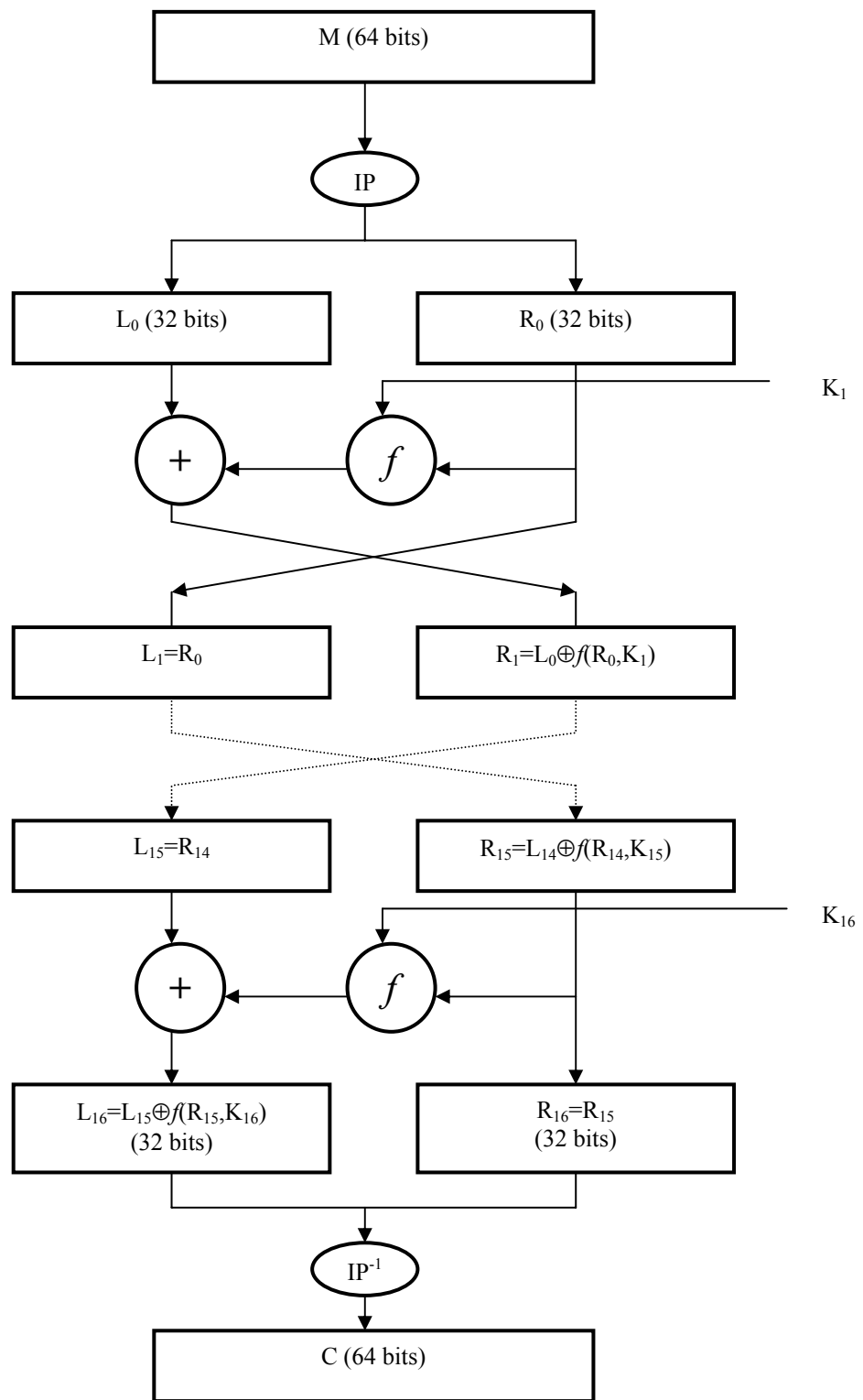


Fig. 5 DES encryption (decryption) algorithm

DES, being the “oldest” cryptosystem widely used, was widely analyzed also having in view the time needed by exhaustive search to find the key (one of 2^{56}) used. As a first remedy, the length of the key from 56 to 112 bits is used.

DES can be implemented in the various ways. First, the block code (cipher) can be generated according to Fig. 6. This procedure is denoted as **Electronic Code Book (ECB)**. Identical messages (with the identical keys) yield the identical cryptograms. Cryptograms are statistically independent and the transmission errors destroy the corresponding block only. ECB is recommended for a short messages only.

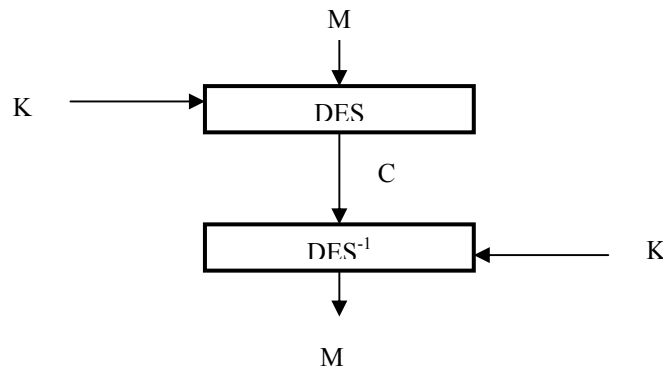


Fig. 6 Electronic Code Book

The next procedure is **Cipher Block Chaining (CBC)**, corresponding partially to the Vigenere cipher, where the key is used for the encryption of first block only, while the next keys are obtained on the autokey basis. IV (*initialization vector*) denotes the initial key, the next keys being the previous cryptograms. In this case, the transmission error spreads to the next block as well (but does not spread further). CBC is shown in Fig 7.

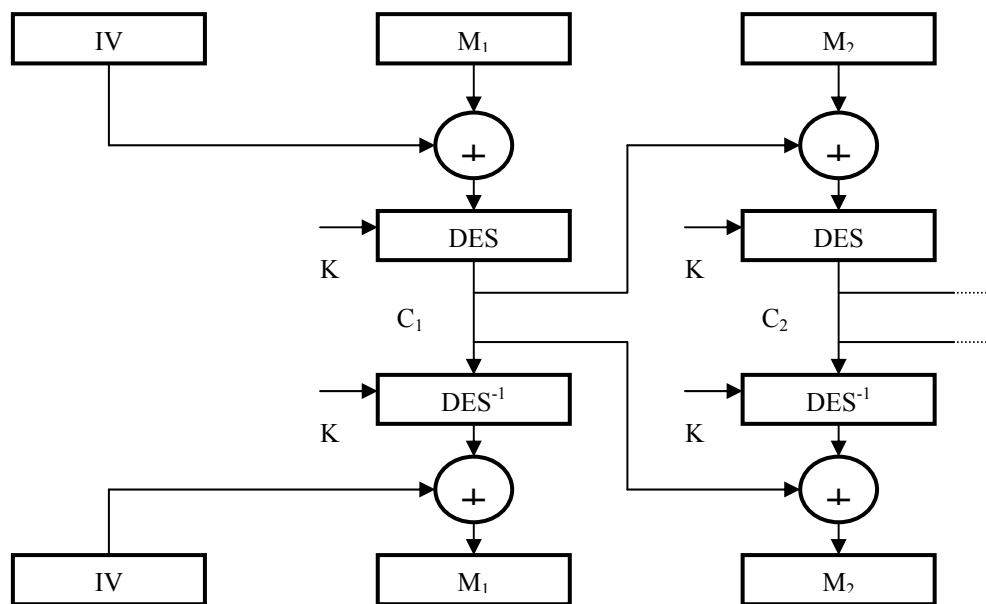


Fig. 7 Cipher Block Chaining

DES chip can be used as well to generate running key for stream ciphers. At the input some initial combination is applied, while the output (usually some part – 1 bit, 2 bits, 4 bits or 8 bits) is used as a key. Thereafter, some part of the output (or the whole output) is applied at the input, united with a part of the shifted version of the preceding input. In such a way the next part of the running key is generated. At the receiving end the running key is generated identically, starting from the same initial combination. This approach is suitable if transmission errors are to be expected, because only corresponding symbol after the decryption will be affected. Third procedure is called **Output FeedBack (OFB)**. On the other hand, if a part of the generated cryptogram at the transmitter is applied at the DES input (instead of a part of DES output) **Cipher FeedBack (CFB)** procedure is obtained. In this case, the possible transmission errors spread over the decrypted text.

For a fixed length of cryptogram, DES can be used more times successively, according to the Fig. 8. Generally, to generate short running keys, as described above, the other symmetrical system can be used as well.

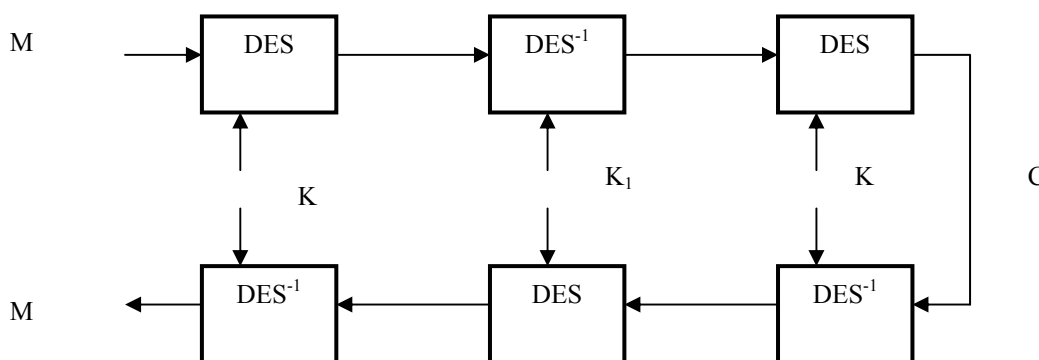


Figure 8 Multiple encryption with DES

An elementary exposition of public key encryption

The basic idea of public key system is explained previously in short. It is an asymmetric cryptosystem where the public key is known, while the private key is known to the specific user only. This idea, practically impossible in the “classical” cryptography, for the first time was introduced in 1976.

A mathematical basis for these systems is the notion of so-called **one way function**, i.e. the function whose values can be easily computed in one way, while the computation of the inverse function is very difficult. The well known example are exponentiation and logarithm computing modulo an integer. When adding or multiplying modulo some integer (n), the corresponding numbers are added or multiplied in a “standard” way, the result is divided by n and the residue (being between 0 and $n-1$) is retained. This means that the result, in the binary form, will always have the same length (the same number of bits). These algebraic structures are called *commutative rings modulo n* . When the multiplying is defined, the exponentiation is as well. Further, an inverse operation – logarithm (often called *discrete logarithm*) computing can be carried out also. However, it can be shown that in considered algebraic structures the exponentiation can be easily carried out (using

squaring and multiplying), while to find a discrete logarithm implies a great number of elementary operations. According to the fastest known algorithm, the needed number of operations – the corresponding time – the number of elementary steps, if n is a prime, is

$$T \approx e^{\sqrt{\ln n \ln(\ln n)}}.$$

It is interesting to note that the same time is needed to factor n , if it is a product of two (big) primes.

The **Euler totient function** $\phi(n)$ is the number of positive integers less than n that are relatively prime to n . If n is a prime, $\phi(n)=n-1$.

Let consider a commutative ring of integers modulo n , if two integers (e and d) are chosen satisfying

$$e \cdot d \equiv 1 \pmod{\phi(n)},$$

then the encryption can be performed by computing the exponential

$$C = M^e \pmod{n},$$

where message (M) is also an element of the ring (its numerical value is also between 0 and $n-1$). The cryptogram C will be of the same length. The decryption is performed by computing the exponential (without finding a discrete logarithm!)

$$M = C^d \pmod{n}.$$

When $\phi(n)$ is known, the pair of keys (e, d) can be found easily (by choosing one key and finding the other from the equation above (using, for example, the well-known Euclid's algorithm). The keys e and d are mutually inverse (their product equals one) and it is irrelevant which one is firstly applied (because of the commutativity). If n and e (either d , but only one from the pair) are known, it is not generally easy to find $\phi(n)$ and d (or e). If someone using the public key e performs encryption of his (known) message M

$$C = M^e \pmod{n}$$

generating the cryptogram C , to find d , the equation

$$M = C^d \pmod{n},$$

has to be solved, to obtain

$$d = \log_C M.$$

This means that the discrete logarithm has to be found.

It should be noted that if n is a big prime (the ring becoming the Galois Field), the public key system cannot be implemented because $\phi(n)=n-1$, and there is no any problem to find d , if n and e are known.

The worldwide known public key system is RSA (*Rivest, Shamir, Adleman*). The exponentiation described above is used for encryption as well as for decryption. However, the cryptogram length is a product of two big unknown primes $n = p \cdot q$, $\phi(n) = (p-1) \cdot (q-1)$ and security is based on the factoring of a known number n into two prime factors to find $\phi(n)$ and subsequently, using known value of e , to compute d . As said earlier, this problem is as difficult as the computing of discrete logarithm. RSA encryption algorithm is very often "attacked" by cryptanalysts. Taking into account the contemporary technology as well as the computing rate, it is believed that n should be represented at least with about 800

bits, while for longtime security about 1000 bits should be used.

A comparison of symmetric-key and public-key cryptography

Symmetric-key systems

Advantages:

- an easy implementation, especially using hardware attains great transmission rates;
- keys are relatively short;
- the systems can be composed to produce stronger ciphers.

Disadvantages:

- the key must remain secret at both ends;
- in a large network, many pairs of keys are needed;
- the keys should be (for security reasons) changed frequently.

Asymmetric-key systems

Advantages:

- only the private key must be kept secret;
- the keys may remain unchanged for considerable period of time;
- in a large network the number of keys may be considerably smaller than in symmetric keys systems.

Disadvantages:

- throughput rate is for several orders of magnitude slower than in best symmetric key systems;
- key sizes are typically much larger than those in symmetrical key systems;
- no public-key system has been proved to be secure.