This is draft translation from Russian the book **Harmed Texts and Multi-channel Cryptography** by V. Michtchenko and Y. Vilanski, «Enciclopedix» 2007.

Authors have granted the following specific permission for this electronic version:

> Authors are granted to retrieve, print and store a single copy of this book for personal use only. The permission does not extend to photocopying or producing copies for other than personal use of the person creating the copy, or making electronic copies available for retrieval by others without prior permission in writing from authors.

Except where overridden by the specific permission above the copyright notice from authors applies to this electronic version:

For further information mail to: mva@open.by.

Valentin Michtchenko and Yury Vilanski

Harmed Texts and Multi-channel Cryptography

Edited by V. Michtchenko

Minsk
«Enciclopedix»
2007

УДК 004.056.55

Мищенко, В.А. Ущербные тексты и многоканальная криптография / В.А. Мищенко, Ю.В. Виланский; под общ. ред. В.А. Мищенко. – Минск: Энциклопедикс, 2007. – 292 с. – ISBN 978-985-6742-46-3

A new science direction in cryptography is considered in the monograph - a multichannel cryptography based on harmed texts. The mechanisms of obtaining harmed texts having no meaning either in a plaintext alphabet or in a ciphertext alphabet are studied. Such an approach allows synthesizing new multi-channel encryption algorithms, and modernizing classical cryptographic protocols from this point of view.

The book is designed for specialists in cryptography and for everyone interested in problems of information security and applications of this direction in mass technologies.

Figures: 56. Tables: 14. Bibliography: 101.

Р е ц е н з е н т ы :
кафедра ЭВМ Белорусского государственного университета информатики и радиоэлектроники;
кандидат физико-математических наук Демиденко В.М.

ISBN 978-985-6742-46-3

Elevated art consists in hiding the art.

A Japanese proverb

Introduction

In his famous work [93] Claude Shannon defined three types of secret systems:

1) systems of cloaking which include using such methods as invisible ink, presentation of a message as an innocuous text or cryptogram masking and other methods with the help of which the fact of message presence is concealed from an adversary;

2) secret systems (for instance, speech inversion) where special equipment is needed to decrypt a message;

3) "properly" secret systems where sense of a message is concealed with the help of a cipher, a code, etc., but the existence of a message itself is not concealed, and an adversary is supposed to have special equipment which he needs to intercept and record signals that are transmitted.

In his work Claude Shannon dwelled only on the third type of the systems. In this book we shall be interested in the combination of the first and the third types of the systems.

Nowadays cryptography possesses a powerful mathematical apparatus for synthesizing practically resistant encryption systems. But it is the knowledge of such an apparatus together with modern methods of cryptanalysis and availability of

computing engines of high performance that makes the existing practically resistant encryption systems unreliable at quantums of time that are large enough. A modern cryptanalysis has various types of attacks which in the end make it possible to define either a secret key or to read a text without this key at all. In all cases the necessary component for a cryptanalysis is an intercepted ciphertext of a secret message. There is no cryptanalysis without a ciphertext. But what should we do if a ciphertext is harmed in a way, that doesn't allow us drawing correct conclusions during a cryptanalysis? Is it possible to deliberately harm a ciphertext in such a way that it would be impossible in principle to successfully carry out an analysis and, at the same time, paradoxically as it may seem, to read it to a legitimate addressee? In this book we made an attempt to define the notion of a text harm from these very positions. We suggested common enough approaches to synthesize ciphers with harmed texts. Such an approach goes back to the first type methods of secret systems in Shannon's classification in combination with the secret systems of the third type.

The idea of harm came during developing multichannel cryptographic systems on the basis of secret splitting, when you need to know all the components to restore a plain text. Evidently, all the separate components or their incomplete totalities may be considered as harmed texts, as they do not allow restoring a plain text without the corresponding long-continued exhaustive search of a missing part. Practical application of these ideas is based on the cryptographic basis of splitting, when special methods and some secret key information are used for these purposes.

Harm of any text is tightly bound with the notion of "meaning".

A ciphertext is a sensible text in a ciphertext alphabet

if a plaintext made sense, but it doesn't make sense in the alphabet of the plaintext. An encryption system allows only transforming one alphabet into another, completely preserving the meaning of a message. Thus, we may harm a ciphertext in two ways: to harm a plaintext by making it meaningless (in this case a ciphertext will not make sense in its alphabet either), or to harm directly a ciphertext by making it meaningless in its alphabet (in this case it will be meaningful, but the meaning will not correspond to the meaning of a plaintext, or will be meaningless at all).

A meaningless casual text can be considered as a noise and, therefore, consciousness analysis gives a negative result. It's impossible to find something that the text doesn't contain. But if it has some meaning expressed with words with the help of an alphabet of a language, then this text can be read, i.e. a source meaning can be restored. From this point of view the task of harming a conscious text is a reverse one: how can we harm a conscious text in such a way that it would be taken as a noise in its alphabet? In other words, how can we transform a conscious text into a text which doesn't have any meaning? This book is devoted to consideration of these questions together with multichannel cryptographic methods of information transformation using steganographic effects.

Introduction, conclusion, chapters 1, 2, 4 and 5 are written by V.A. Michtchenko, chapter 3 – by V.A. Michtchenko and Y.V. Vilanski.

The authors would like to express their sincere thanks to their colleagues in "Creative Laboratory" for discussing the information and helping at getting up the book; doctor, professor Suchindran S. Chatterjee; candidate of physical and mathematical sciences, associate professor V.M. Demidenko; Mr. Michael Ellenby; Mr. Carl A. Erickson; doctor of technical sciences, professor E.A. Golubev; Mr. Gareth James; candi-

date of physical and mathematical sciences, associate professor V.V. Lepin; doctor of technical sciences, professor R.H. Sadyhov; doctor of technical sciences, professor M.L. Seleznev; doctor David J. Soldera; doctor of technical sciences, professor Y.A. Khetagurov for being interested in this topic, positive criticism and support of any kind, for popularization of new approaches, for desire and patience to understand the new written in this book.

Valentin Michtchenko,
Yury Vilanski

Letter contracts and designations

We shall start our book with definitions of some universally accepted terms and notions, as sometimes they are interpreted in some other meanings and can mislead an inexperienced reader. The reader can find the terminology and basic definitions in the appendix A.

Definition of some commonly accepted terms and concepts

Information – data transmitted by people or devices with the help of prearranged symbols.

Language - a system of discrete symbols designed for communication purposes.

Information language – a special artificial language used in various information-processing systems.

Meaning - a notion describing a global sense of a statement which is not limited by the meaning of its components and elements but which defines these meanings itself.

Text – a set of language symbols having a meaning.

Randomized text – a text in which certain language symbols take random meanings from the finite set of meanings.

Semantic information – a characteristic of a content transmitted in a message.

Algorithm [15] – a notion similar to the notions of "cookie", "process", "procedure". It's not just a set of finite number of rules setting an execution sequence of some operation. In addition to this an algorithm has five basic peculiarities:

- the number of algorithm's steps should be finite;
- every step of the algorithm should be defined explicitly;
- the algorithm works with some source data or data obtained during algorithm functioning;
- some output data are obtained in the result of algorithm functioning;
- the algorithm must be effective, i.e., all its steps must be executed in a finite time interval.

Computing method – a notion equivalent to the notion of the algorithm without efficiency properties.

Programming language – a language to describe algorithms.

Program – a computing method written in a programming language.

System – a set of elements interrelated and interconnected with each other, which form a particular integrity, unity in a certain sense.

Code – a system of conventional signs to perform information signals with the purpose of its transmission, processing and storing.

Designations frequently used throughout the book

E – an encryption operator

E^{-1} – a decryption operator

K – a secret encryption key in symmetric systems

k_{pb} – a public key in an asymmetric system with a public key

k_{pr} – a secret key in an asymmetric system with a public key

Y – a cryptogram (ciphertext)

M – a plaintext (message) liable to encryption

L – a message length

H – indeterminacy (entropy)

B – redundancy of language

r – entropy of language per symbol

Y_{D} – a harm

Y_{DT} – a harmed ciphertext

C – a data-pump - a harmed ciphertext in the MV2 algorithm

F – flags - a harm in the MV2 algorithm

$K_{\mathrm{D}}^{(i)}$ – a key of the harm at i-th round

CHT – a ciphertext of a harmed text

CH – a ciphertext of a harm

HCT – a harmed ciphertext

HC – a harm of a ciphertext

$\binom{n}{m}$ – a number of combinations from n to m

$\|$ – an operation of concatenation

$\{0,1\}^n$ – a set of all n-digit binary strings

log – a binary logarithm by the base 2 ($\log_2$)

$\oplus$ – XOR (exclusive OR)

$\#\{x\}$ – capacity(a number of elements) in a set $\{x\}$

$GF(2)$ – 2.

$(GF(2))^n$ – n-dimension vector space above the finite field

$\delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$, – a saltus function

$\mathcal{F}_r^n$ – a set of MV2-type transformations

$\mathcal{U}_{rk}$ – a set of binary strings containing no less than r and no more than k digits: $\mathcal{U}_{rk} = \bigcup_{i=r}^{k} \{0,1\}^i$

Z_p – a group of module deductions p

Chapter 1
Harmed texts

1.1 Meaning of texts and information theory

The French physicist L. Brillouin [4] interrelated information and physical entropy. This interrelation was initially put in the very basis of the information theory, as Shannon suggested to use the probabilistic entropy function, that was borrowed from statistical thermodynamics, to calculate amount of information.

With the help of the entropy function one can also analyze a written conscious text, because symbols of a conscious text have different probability of their appearance, and don't happen chaotically in a conscious text, but have some order determined by the rules of word-formation and use of words in an utterance. But any text has not only entropic characteristics, but also characteristics of meaning and value of the information they contain. C. Shannon deliberately simplified his model: information theory doesn't consider these properties of transferable information. These properties is a concern of transmission and receiving parts. Shannon's information theory only gives a quantitative measure of

transferable information not worrying about its properties of meaning and value.

Hereinafter we shall use the notion of "meaning", therefore we shall give an encyclopedic definition to this term.

In the model "meaning – text" a meaning is a notion that describes a global content of an utterance. [3].

Definition 1.1 *The term "meaning" can signify the entire content of an utterance that is not reduced to the sense of its components, but that defines these senses itself.*

Any utterance contains notational words that are defined by meaning, and auxiliary words. Depending on the meaning of an utterance practically any part of a sentence (a subject, predicate, adverbial modifier of place and time, object, rarely – adjectives, and sometimes even prepositions) can be notational words. To understand this definition better we shall consider an example.

Example 1.1 *The phrase "I went to the cinema" might have several meanings and its key words for every meaning. If the sense of the phrase is who went to the cinema, then the key word is "I" and the phrase "went to the cinema" is meaningless.*

If the meaning is where I went, then the key word will be the combination of words "to the cinema", and the phrase "I went" is meaningless, etc.

The information theory doesn't study estimation of meaning and value of the transmitted information as they (meaning and value) are subjective. The information theory only allows to estimate the degree of order of a text or degree of its deviation from the state of the complete chaos when all the letters would have equal probability and the text would become a meaningless set of letters.

The more the difference of probabilities and the more the probability of the following letters depends on the probability of the previous letters, the more text order is. At that quantity of information that evaluates this order will be equal to decrease of a text entropy in comparison with the maximally possible entropy value that corresponds to absence of order in a text at all, i.e. corresponds to equiprobable appearance of any letter after any previous letter. Techniques of information calculation that were suggested by C. Shannon, allow displaying ratio of quantity of predictable (i.e. the one that is formed according to certain rules) information and quantity of unexpected information that can not be predicted in advance. Shannon defined the information contained in the rules as redundant, because knowing rules of message building allows probabilistically predicting letter appearance before they are transmitted.

If in a language an alphabet that contains N number of symbols is used, then *the absolute entropy* of the language at equiprobable use of all the symbols is $R = \log N$.

For example, for the English language which has the number of letters that is equal to $N = 26$ the value is $R = \log 26 = 4,7$ bits per alphabet letter. It is the maximal entropy of particular symbols. But as probabilities of use of particular symbols are different, in reality *language entropy per one symbol* of a message M is $r = H(M)/L$, where $H(M)$ is indetermination of a message, and L is a length of a message in symbols of the alphabet. In a number of researches [40] the value of entropy per symbol is defined for long messages of the English language. This value is 1,3 bits per letter.

The value $B = R - r$ is called *redundancy of language*. Redundancy of language for English is about 3,4 bits per letter.

In a simplified model of the English language, where all

the punctuation marks, spaces and numbers are omitted, for the 8-bit letter image in the ASCII table when $r = 1,3$ redundancy will be 6,7 bits per letter!

Encryption system allows transforming an alphabet of a plaintext into an alphabet of a ciphertext and vice versa without changing the meaning of a message. Such a system is destined to conceal meaning in an alphabet of a source language. But redundancy of language enables to keep certain information about a plaintext in a ciphertext. This fact together with a number of statistical regularities allow a cryptanalyst reading ciphertexts without knowing a key, or even defining the key itself. C. Shannon showed that if an encryption system eliminates completely redundancy of a plaintext, it becomes impossible in principle to restore a plaintext according to a ciphertext.

1.2 Cipher attacks. The concept of harmed texts

Let a cipher be defined as (E, E^{-1}, M, Y, K). Here M, Y are a plaintext and a ciphertext correspondingly, E, E^{-1} are encryption and decryption transformations, and K is a private key. The main situations during a cryptanalysis can be brought to the following cases [13]:

1. One or several ciphertexts Y are known. The aim of a cryptanalyst is to define E (a kind of a cipher), to find E, E^{-1}, M.

2. One or several pairs of (M, Y) are known. To define a kind of the cipher E or E^{-1} and to find K.

3. The kind of the cipher E or E^{-1} and one or several ciphertexts Y are known. To find M or K, M.

4. The kind of the cipher E or E^{-1} and one or several pairs of (M, Y) are known. The task is to find K.

5. E, E^{-1}, a ciphertext Y or pairs (M, Y), some transformation form $E(., K)$ are known, but K and $E^{-1}(., K)$ are unknown. Such a formulation is typical for systems with a public key. The task is to find K.

In all the above listed cases you need to know at least a ciphertext Y. Let's assume that a cryptanalyst doesn't get a real ciphertext Y, but some other ciphertext Y', which is not a regular result of encryption, but it is deliberately corrupted in such a way that it would be impossible to restore a plaintext M from it. Is it possible? Yes, it is possible. For instance, let encryption be carried out in the way that

$$Y' = E_1(M, K)$$
$$Y'' = E_2(M, K),$$

where the mappings E_1 and E_2 are not injective, but are interconnected in the following way: there's such a mapping E_{12}^{-1}, that for any plaintext M and a key K the following equation is carried out:

$$M = E_{12}^{-1}(Y', Y'', K).$$

Let Y'' be concealed from a cryptanalyst thanks to transmitting via another channel or with the help of concealment methods that are unknown to a cryptanalyst, or it is encrypted by a perfect secrecy system due to a small size of Y''. Apart from enlarged field of keys at the expense of cryptographic splitting a cryptanalyst faces the task of brute

force to get the missing part Y''. It might be a very time-consuming task as to computing relation due to the fact that a length Y'' may be greater than a length K. We can develop this idea by introducing m number of ciphertext and assuming that:

$$\begin{aligned} &Y' = E_1(M, K) \\ &Y'' = E_2(M, K) \\ &\dots \\ &Y'^{(m)} = E_m(M, K); \\ &M = E^{-1}_{1\dots m}(Y', Y'', \dots Y'^{(m)}, K). \end{aligned}$$

In such a formulation we shall call ciphertexts $Y'^{(i)}$ *harmed texts*, if each of them is meaningless in an alphabet of a ciphertext.

1.3 The concept of harmed texts

Let us have a text M the length of which is L_0 and the meaning is $S(M)$. Let this text be written in some language in an alphabet A with redundancy of language equal to B_A and, correspondingly with redundancy of text

$$B(M) = B_A L_0 = \left(\log N - \frac{H(M)}{L_0} \right) \cdot L_0.$$

Let us have at our disposal an ideal compression technique which allows eliminating all the redundancy and getting M' – a text of the minimal length L_{min} at preserving the meaning, i.e.:

$$S(M') = S(M).$$

Let's note that any data compression technique makes a text length smaller, but keeps the meaning of a message unchanged by converting an alphabet of a plaintext into an alphabet of an archiver, as an ordinary information encoder.

In reality any archiver works worse than an ideal one, that is why a text M'' created with its help has a greater length L_{min} and preserves the meaning of a plaintext:

$$S(M'') = S(M). \tag{1.1}$$

Evidently, further attempts to lessen the length of a text will lead to distortion of a message meaning and, therefore, for any text M^* of a length $L < L_{min}$ the equality (1.1) won't be observed:

$$S(M^*) \neq S(M).$$

We get this effect because the subsequent reduction of a text occurs at the expense of "deformation" of letters, which are represented by some code and are already irredundant, not at the expense of eliminating redundancy of letters. At that under "deformation" of letters we understand subsequent length reduction of letter codes *beyond its information irredundancy.*

Definition 1.2 *We shall call a text that was obtained with the help of a subsequent deformation of letters with reduction of their length after elimination of redundancy as a harmed text.*

Thus, the necessary and sufficient condition of redundancy of text with loss of a meaning is a reduction of a code length of text symbols outside their irredundancy. As a result of this, a harmed text has a smaller length than a length of a plaintext and doesn't have the meaning of a plaintext.

It follows from this definition that the whole text set, occurred from some plaintext and a set of transformations, consists of two disjoint subsets: harmed texts and texts that have a meaning of a plaintext. Pay attention to the following fact: all ciphertexts have a meaning of a plaintext in an alphabet of a compressing alphabet, all elementarily transformed plaintexts can preserve the meaning of a plaintext

by throwing out particular words or symbols. Consequently, all these transformed texts are not harmed.

One may suggest many enough methods for checking texts for consciousness. [2]. The disadvantage of all well-known methods is detailed knowledge of a statistical language structure. In our argumentation we shall be guided by some other positions: position of redundancy of language. In contrast to attempts to measure a meaning we come to the idea of only fixing presence or absence of a meaning, possibly with probability measure, at that

$$P(S(M)) + Q(S(M)) = 1,$$

where P and Q are probability of presence and probability of absence of a meaning correspondingly. It results in necessity to create a mechanism that would destroy a meaning of plaintexts or a ciphertext (a mechanism of harming), and that would have $P(S(M))$ close to 0. In this way we won't worry about the problems of measuring a meaning or defining features of consciousness; we shall be interested in *a mechanism of meaning destruction* with a probabilistic measure of presence or absence of a meaning.

Such a mechanism might have its key which additionally increases key space.

Definition 1.3 *We shall call a cyclic algorithm of obtaining harmed texts that consists in a random substitution of bit representation of every symbol of a plaintext by a tuple of a smaller or equal number of bits with their further concatenation as an universal mechanism of harming and shall indicate C_m, where m is a number of rounds.*

It follows from the definition that during harming codes of text symbols are replaced by tuples of various length. The advantage of such an approach is its universality. Irrespective of the nature of a plaintext (texts in natural languages,

ciphertexts, program file texts and so on) it allows destroying a meaning and verify its absence after some number of rounds m of mechanism of harming C_m are executed.

As the further reduction of a text length outside irredundancy leads to distortion of a message meaning, additional information is needed to restore a text on the basis of a harmed text. We shall call this additional information *a harm*.

A harm restores broken injectiveness of a transformation at irregular substitutions for a given harmed text. We shall be interested in such rules of harming that doesn't allow restoring a plaintext (may be except attempts of brute force) if one has only harmed texts or harms. The object of interest for us will be such rules of harming that require knowing all harmed texts, all harms and the rule of harming itself to restore a plaintext. This idea allows pretty flexible implementation of the algorithms, because the process of harming can be cyclic, and one can change the rule of harming and manage the length of a final harmed text by changing the number of steps at every stage. This mechanism of obtaining harmed texts destroys a meaning of a plaintext and generates information that allows restoring a plaintext and its meaning.

A harmed text is always random, as it is defined only by random tuples of variable length. A harm only characterizes a length of random substitutions during execution of C_m and doesn't bear any semantic stress, being in fact a harmed text.

Thus, in the result we have two ciphertexts (a harm and a harmed text), none of them has a meaning neither in an alphabet of a plaintext nor in an alphabet of a ciphertext. Actually we presented a ciphertext of a plaintext in form of a set of two harmed ciphertexts, each of which separately can't restore a plaintext. Here we have a realized cryptographic idea of secret splitting [30], when information is split according to a key into two or more parts and one should know all the

parts to restore it.

Peculiarity of this process is that there's no need to know intermediate harmed sequences to restore the original sequence. One should know only the last harmed sequence (the last harm after all the rounds are over) and all the harms together with the rules of their use.

Theorem 1.1 *Let M be a sensible text of a length L_0, M_a be a text with a length $L_a < L_0$, that was received from M with the help of "an ideal" archiver, Y_{DT} is a text derived from a text M after executing m rounds of mechanism of harming C_m and its length $L(Y_{\mathrm{DT}}) < L_a$. Then an obtained text Y_{DT} is harmed.*

Proof. Let Y_{DT} be not a harmed text with a length $L_{\mathrm{DT}} < L_a$. Then a transformation $C_m(M)$ is an archiver that gives a sensible text. It contradicts the hypothesis, because data compression can not give a sensible text with a length $L < L_a$. Consequently, the text Y_{DT} is harmed. □

We shall consider an English text as an example. Let a plaintext have a length L_0 bites. At 8-bit symbol (letter) presentation in accordance with the ASCII table for a message with all punctuation marks and numbers this text contains $B_A = 3,4$ bits of redundant information per letter. Therefore an ideal archiver can lessen a length of a plaintext till the value:

$$\frac{8-3,4}{8}L_0 = 0,575L_0,$$

and preserve a meaning of a plaintext. Further lessening of letter lengths in a bit dimension will lead to a distortion or loss of a meaning. Let a length of a previous text become η times smaller at every round of a universal mechanism of harming. Then a number of rounds m, necessary to destroy a

meaning can be defined by the inequality:

$$\frac{L_0}{\eta^m} < 0,575L_0,$$

from which it follows that:

$$m > \frac{\log 1,739}{\log \eta}.$$

In order destroying of a meaning in a harmed text to occur with a high probability the number m should be selected taking into account certain applications and a method of immersion of a mechanism of meaning distortion into a computing environment. If, for example, someone uses a mechanism as an encryption system with an observable harmed text and a harm, then m should be no less than 16.

One can draw an analogy between ciphertexts and harmed texts.

A ciphertext contains the whole meaning of a plaintext in an alphabet of a ciphertext and changes its image without changing a meaning with a key change at a constant encryption algorithm.

All harmed texts together with harms contain a meaning of a plaintext and change their presentation without changing a meaning at a change of a rule of harming. Here a ciphertext and a set of harmed texts are equivalent to each other. But not a single particular harmed text or incomplete set of harmed texts contains a meaning of a plaintext.

This statement is based on the definition of a meaning, that requires certain words of a thesaurus that are not present in a harmed text any more, because its length was reduced, and these statements result from impossibility to restore a compressed in some way text more than redundant information allows doing it.

A quantitative measure of effectiveness of a harm is a degree of a meaning distortion that is equal to difference of entropies of a harmed text and a plaintext at various length segments of a harmed text. The quantity of such segments equals

$$s = \left[\frac{L_0 - L_{\mathrm{DT}}}{L_{\mathrm{DT}}} \right],$$

where L_0 and L_{DT} are lengths of a plaintext and a harmed text correspondingly.

Therefore, the degree of meaning distortion of a plaintext can be evaluated by the value

$$d = H(Y_{\mathrm{DT}}) - \sum_{i=1}^{s} H(M_i)p_i, \quad \sum_{i=1}^{s} p_i = 1,$$

where M_i is a part of a plaintext that corresponds to ith segment, p_i is its probability and a length of every M_i equals a length Y_{DT}.

If a text is generated by an ergodic source, then

$$d = H(Y_{\mathrm{DT}}) - H(M_i).$$

A value d characterizes the degree of symbol disorder of a harmed text in comparison with the order of a plaintext. At equiprobable distribution of symbols in a harmed text (that corresponds to the maximal harm) a value d has the maximal value

$$d_{max} = \log L_{\mathrm{DT}} - \sum_{i=1}^{s} H(M_i)p_i; \quad \sum_{i=1}^{s} p_i = 1$$

or for an ergodic symbol source of a plaintext:

$$d_{max} = \log L_{\mathrm{DT}} - H(M_i).$$

Let's consider an example of a universal mechanism of harming.

Example 1.2 *We shall consider a text consisting of a single word : " Аргонавт ". Harm this text with loss of a meaning.*

In accordance with the ASCII table (Application D) we have:

$A \longrightarrow$ 11000000; $p \longrightarrow$ 11100000; $г \longrightarrow$ 11100011;
$о \longrightarrow$ 11101110; $н \longrightarrow$ 11101101; $а \longrightarrow$ 11100000;
$в \longrightarrow$ 11100010; $т \longrightarrow$ 11110010.

To make the mechanism visual we shall arrange these bites in accordance with a word structure and mark borders of every bite:
The text (Аргонавт) $\longrightarrow$ *11000000‖ 11110000‖ 11100011‖ 11101110‖ 11101101‖ 11100000‖ 11100010‖ 11110010.*

Let the following table of irregular substitutions be randomly chosen at the first step:

11000000$\longrightarrow$*011;* *11110000*$\longrightarrow$*10101;* *11100011*$\longrightarrow$*0101;*
11101110$\longrightarrow$*1111;* *11101101*$\longrightarrow$*000;* *11100000*$\longrightarrow$*11;*
11100010$\longrightarrow$*100001;* *11110010*$\longrightarrow$*00110.*

At that we shall get:

A harmed text of the first round:
01110101‖ 01011111‖ 00011100‖ 00100110 $\longrightarrow$ *u|_|FS|&*

A harm of the first round:
00100001‖ 00010001‖ 00101000‖ 00100001 $\longrightarrow$*!|DC1|(| !*

At the second round the table of irregular substitutions can be different:

01110101$\longrightarrow$ *110100;* *01011111*$\longrightarrow$ *001;*
00011100$\longrightarrow$ *01110;* *00100110*$\longrightarrow$ *11.*

A harmed text of the second round:
11010000‖10111011 $\longrightarrow$*P(Кир.)|"*

A harm of the second round:
00000100‖10000101 $\longrightarrow$ *EOT|...*

A harm indicates a length of a substitution at reverse restoring of a plaintext.

Here the rule of harming (the table of irregular substitutions) plays a part of a key, and depending on it, as at encryption during a key change, a harm and a harmed text will be changed. Thus, for example, if the rule of harming lies in a different table of substitutions, we shall get different results at the first round:

11000000$\longrightarrow$*01;* *11110000*$\longrightarrow$*101101;*
11100011$\longrightarrow$*01101;* *11101110*$\longrightarrow$*111;*
11101101$\longrightarrow$*010;* *11100000*$\longrightarrow$*10;*
11100010$\longrightarrow$*100011;* *11110010*$\longrightarrow$*01110.*

A harmed text of the first round:

01101101‖01101111‖01010100‖01101110 ⟶ m|o|T|n

A harm of the first round:

01000001‖00001001‖00101000‖00100001 ⟶A|HT|(|!

Peculiarity of this process is that there's no need to know intermediate harmed sequences to restore an original sequence. One should know only the last harmed sequence and all the harms together with the rules of their use. For our example we finally have:

A harmed text: 11010000‖10111011 ⟶P(Кир.)|"

A harm of the second round: 00000100‖10000101 ⟶EOT|...

A harm of the first round:

00100001‖00010001‖00101000‖00100001 ⟶!|DC1|(|!

Thus, we can analytically write the process of getting harmed texts in the following way:

$$\begin{array}{l} Y_{\mathrm{DT}}^{(i)} = D_1^{(i)}(Y_{\mathrm{DT}}^{(i-1)}, K_{\mathrm{D}}^{(i)}) \\ Y_{\mathrm{D}}^{(i)} = D_2^{(i)}(Y_{\mathrm{DT}}^{(i-1)}, K_{\mathrm{D}}^{(i)}) \end{array}, \quad i = 1, 2, ...m,$$

where $Y_{\mathrm{DT}}^{(i)}$ is a harned text and $Y_{\mathrm{D}}^{(i)}$ is a harm, obtained at ith round, $K_{\mathrm{D}}^{(i)}$ is a key of a harm at ith round, $Y_{\mathrm{DT}}^{(0)} = M$ is a plaintext, m is a number of rounds.

The process of restoring a plaintext looks in the following way:

$$Y_{\mathrm{DT}}^{(i-1)} = D_i^{-1}(Y_{\mathrm{DT}}^{(i)}, Y_{\mathrm{D}}^{(i)}, K_{\mathrm{D}}^{(i)}), \quad i = m, m-1, \ldots, 1.$$

In the Fig. 1.1 a general scheme of a round of a universal mechanism of harming C_m is represented. A text $Y_{\mathrm{DT}}^{(i-1)}$ that goes to the input is obtained at the previous round of the mechanism C_m. At the first round it coincides with a plaintext: $Y_{\mathrm{DT}}^{(0)} = M$. This scheme presupposes executing a special splitting transformation under an input text $Y_{\mathrm{DT}}^{(i-1)}$ with the parameters (a key) $K_{\mathrm{D}}^{(i)}$; in the result we get two output texts $Y^{(i)}$ and $Y_{\mathrm{D}}^{(i)}$. In a general case it is conceded that a text $Y^{(i)}$ obtained in the result of using a splitting

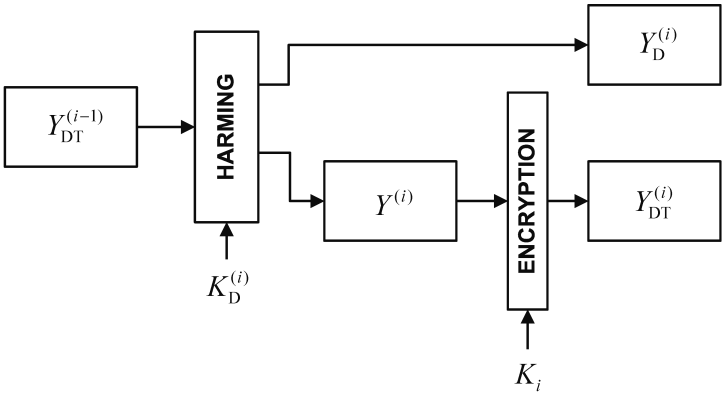

Fig. 1.1: *A round of universal mechanism of harming*

transformation, may be additionally encrypted with the help of a key K_i, in the result the output of every round is a text $Y_{\mathrm{DT}}^{(i)}$.

Then a universal mechanism of harming C_m can be described as

$$\begin{array}{l} Y_{\mathrm{DT}} = E_1(M, K) \\ Y_{\mathrm{D}} = E_2(M, K) \\ M = E_{12}^{-1}(Y_{\mathrm{DT}}, Y_{\mathrm{D}}, K) \end{array}, \qquad (1.2)$$

where

$$\begin{array}{l} Y_{\mathrm{DT}} = Y_{\mathrm{DT}}^{(m)}, \\ K = \phi(K_{\mathrm{D}}^{(1)}, \ldots, K_{\mathrm{D}}^{(m)}, K_1, \ldots, K_m), \\ Y_{\mathrm{DT}} = \psi(Y_{\mathrm{D}}^{(1)}, \ldots, Y_{\mathrm{D}}^{(m)}) \end{array} \qquad (1.3)$$

and ϕ is some presentation for a key $K_{\mathrm{D}}^{(1)}$, …, $K_{\mathrm{D}}^{(m)}$, K_1, …, K_m and ψ is some presentation for all harms $Y_{\mathrm{D}}^{(1)}$, …, $Y_{\mathrm{D}}^{(m)}$, that depends on specific application.

The mechanism of harming in point affects all the symbols of a text; in our example it affects all the bytes. In a general case all the letter symbols of an original sequence will be changed because each symbol of an arbitrary alphabet is

represented by a block of bits, and, consequently, the words will be also changed at the expense of letter deformation outside their irredundancy, not due to procedure of standard substitutions of a fixed length. Therefore the meaning of the last original sequence which defines these words is destroyed. At that a harmed sequence decreases. It doesn't happen because of a compression, but due to a deformation, decrease of a letter bit length beyond irredundancy which results in a loss of meaning. It's obvious that such a process can be carried out over and over again with obtained harmed sequences, getting the latter and harms of the second, third and so on levels that are connected with them.

1.4 Cryptographically harmed texts and multichannel cryptography

In cryptography we can harm texts with the help of the following methods:

1) harming a plaintext and further encryption of a harmed text and a harm (Fig. 1.2). At that in the output we shall have *a ciphertext of a harmed text* CHT *and a ciphertext of a harm* CH ;

2) harming a ciphertext and obtaining a *harmed ciphertext* HCT *and a harm of a ciphertext* HC (Fig. 1.3);

3) a combined method of harming a plaintext with further encryption of a harmed text CHT and a harm CH and an additional harming a text CHT and/or a text CH. At that in the output we have *harmed texts* HCT(CHT)

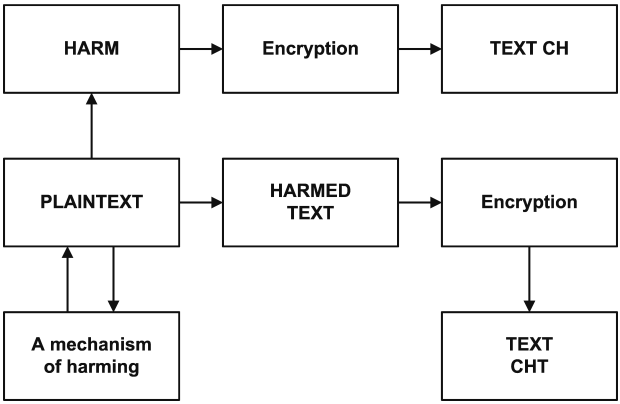

Fig. 1.2: *Scheme of obtaining harmed texts of the types* CHT *and* CH

and HC(CHT) in case of harming repeatedly the text CHT (Fig. 1.4); *harmed texts* HCT(CH) *and*HC(CH) in case of harming repeatedly the text CH (Fig. 1.5); *harmed texts* HCT(CHT) *and* HC(CHT), HCT(CH) *and* HC(CH) in case of harming repeatedly the text CHT and CH (Fig. 1.6).

Definition 1.4 *We shall call ciphertexts obtained with the help of the following methods as cryptographically harmed texts:*

- *by harming a plaintext with further encryption of a harmed text and/or its harms;*
- *by harming a ciphertext;*
- *by harming a ciphertext of a plaintext and/or a ciphertext of harms.*

A cryptographically harmed text doesn't have any meaning in an alphabet of a ciphertext, as it is either obtained by

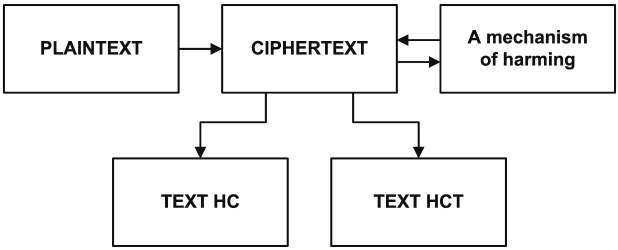

Fig. 1.3: *Scheme of obtaining harmed texts of the types* HCT *and* HC

encrypting meaningless texts in the result of harming a plaintext, or a ciphertext was harmed and it resulted in loss of meaning in an alphabet of a ciphertext. This important feature allows to synthesize a new class of cryptographically secure systems.

From the point of view of a cryptanalyst a process of encryption by an algorithm has two degrees of freedom: a choice of a message from the whole set of possible messages, and a choice of a key from the possible key field. These degrees of freedom are correspondingly characterized by entropy of a message M and that one of a key K :

$$\begin{aligned} H(M) &= -\sum_{x\in\mathcal{M}} p(x)\log p(x); \\ H(K) &= -\sum_{k\in\mathcal{K}} p(k)\log p(k), \end{aligned} \tag{1.4}$$

where $\mathcal{M}$ – the set of plaintexts, $\mathcal{K}$ – a set of keys, $p(x)$ – probability of appearance of a certain text $x \in \mathcal{M}$ and $p(k)$ – probability of a choice of a specific key $k \in \mathcal{K}$.

After a cryptanalyst observes a ciphertext Y the characteristics (1.4) will be changed and will possess the values:

$$\begin{aligned} H(M|Y) &= -\sum_{(x,y)\in\mathcal{M}\times\mathcal{Y}} p(x,y)\log p(x|y); \\ H(K|Y) &= -\sum_{(x,y)\in\mathcal{K}\times\mathcal{Y}} p(k,y)\log p(k|y), \end{aligned} \tag{1.5}$$

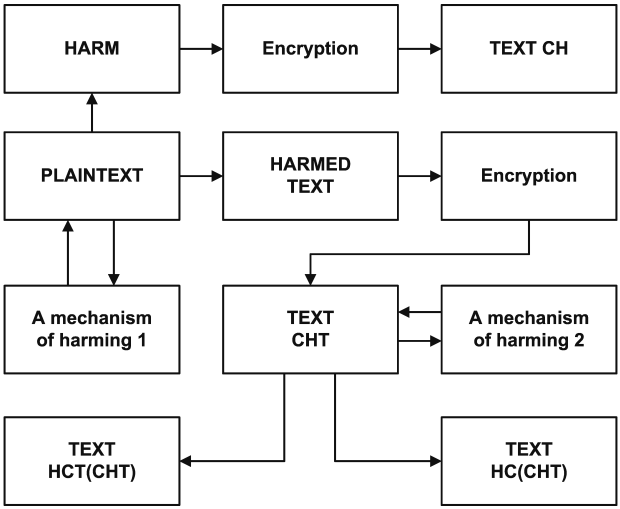

Fig. 1.4: *Obtaining harmed texts* HCT(CHT) *and* HC(CHT) *in case of harming repeatedly the text* CHT

where $\mathcal{Y}$ – a set of ciphertexts, $p(x, y)$ and $p(k, y)$ – joint probabilities of appearance of a message x and a ciphertext y accordingly, a choice of a key k and appearance of the ciphertext y, a $p(x|y)$ and $p(k|y)$ – conditional probabilities of encryption of the message x and of use of the key k accordingly, on conditions that a cryptanalyst observes the ciphertext y.

C. Shannon called the estimations (1.5) accordingly insecurity of a plaintext and key which characterize *theoretical secrecy measure* [29].

For an initial segment M_L of the plaintext M and Y_L – ciphertext Y of the length L

$$\begin{aligned} H(M_L|Y_L) &= -\sum_{(x_L,y_L)\in\mathcal{M}_L\times\mathcal{Y}_L} p(x_L, y_L)\log p(x_L|y_L); \\ H(K|Y_L) &= -\sum_{(k,y_L)\in\mathcal{K}\times\mathcal{Y}_L} p(k, y_L)\log p(k|y_L). \end{aligned} \tag{1.6}$$

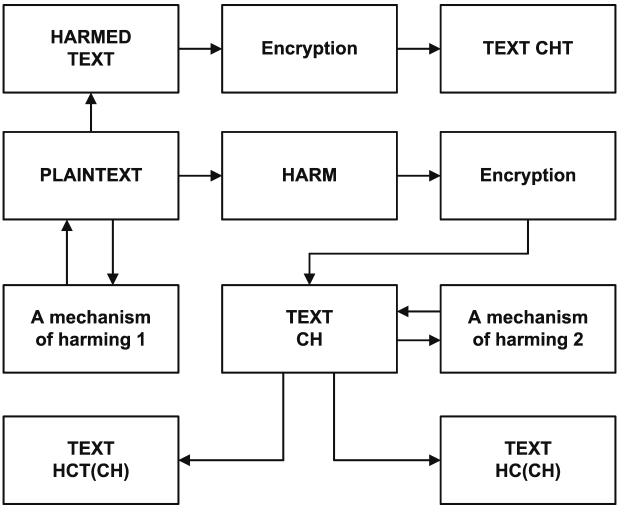

Fig. 1.5: *Obtaining harmed texts* HCT(CH) *and* HC(CH) *in case of harming repeatedly the text* CH

According to the C. Shannon's theorem about insecurity of a key and message [29]:

1. Insecurity of a key $H(K|Y_L)$ is a nonincreasing function from L, i.e.:

$$H(K|Y_{L_1}) \geq H(K|Y_{L_2}) \text{ for any } L_2 > L_1.$$

2. Insecurity of the first z letters of a message is a nonincreasing function from L, i.e.:

$$H(M_L|Y_{L_1}) \geq H(M_L|Y_{L_2}) \text{ for any } L_2 > L_1.$$

3. At any L we have the following inequation:

$$H(K|Y_L) \geq H(M_L|Y_L).$$

We shall further use a Shannon's notion "unicity distance" [29], which can be defined in the following way.

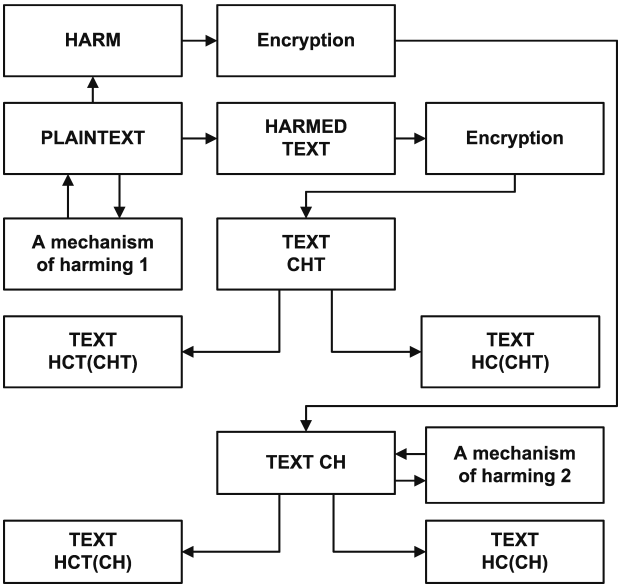

Fig. 1.6: *Obtaining harmed texts* HCT(CHT) *and* HC(CHT), HCT(CH) *and* HC(CH) *in case of harming repeatedly the text* CHT *and* CH

Definition 1.5 *Under unicity distance of a cipher regarding an open text we shall understand such a minimal natural number L, at which according to the known ciphertext Y_L an open message M_L corresponding to it is restored explicitly.*

Definition 1.6 *Under unicity distance of a cipher regarding a key we shall understand such a minimal natural number L, at which according to the known ciphertext Y_L the encryption key K is defined explicitly.*

C. Shannon defined that the average number of texts $\overline{s}_L$, which can be encrypted into a set ciphertext Y_L, satisfies the equation:

$$\log \overline{s}_L = H(M_L|Y_L), \tag{1.7}$$

and the average number of keys $\overline{k}_L$, for which we can obtain a segment of the ciphertext Y_L, satisfies the equation:

$$\log \overline{k}_L = H(K|Y_L). \tag{1.8}$$

If for a specific cipher there's a solution of the equations (1.7) and (1.8) relative to L at $\overline{s}_L = \overline{k}_L = 1$, then, the found $L = U_0$ is unicity distance.

We shall find unitiсy distance for a model of a random cipher [2], for which there is a probability of obtaining a conscious text at a random and equiprobable choice of the key K and at an attempt of encrypting with it the ciphertext Y_L. This probability equals :

$$P_s = \frac{2^{H \cdot L}}{|I|^L},$$

where H – entropy per letter of a conscious text in an input alphabet I, $|I| > 2$, $2^{H \cdot L}$ – an approximate value of the number of conscious texts. During encryption of a ciphertext of the length L on the whole key field K we shall get in average

$$H(K)\frac{2^{H \cdot L}}{|I|^L} = N_s \tag{1.9}$$

conscious texts. Assuming $N_s = 1$, we shall find unicity distance:

$$L = U_0 = \frac{H(K)}{\log |I| - H} = \frac{H(K)}{B \log |I|}, \tag{1.10}$$

where $B = 1 - \dfrac{H}{\log |I|}$ – redundancy of a plaintext.

As the expression (1.9) is at one time the average number of keys for obtaining conscious texts, then the expression (1.10) is at one time is unicity distance for the key, i.e. for the whole considered random cipher. This feature of ciphers

significantly facilitates cryptanalysis, especially if we take into account that the value U_0 for natural languages and modern key fields is not large. In the Table 1.1 there are values U_0 for the English language at different values $H(K)$ [44].

Tabl. 1.1: *Values U_0(of text symbols) for the English language*

H(K)	40	56	64	80	128	256
U_0	5,9	8,2	9,4	11,8	18,8	37,6

For the first method of harming when a universal mechanism C_m is represented by the expressions (1.2), (1.3), the expression (1.10) will be transformed:

$$U_0 = \frac{\sum_{i=1}^{m}(H(K_{\mathrm{D}}^{(i)}) + H(K_i))}{B \log |I|}. \qquad (1.11)$$

If a plaintext had a meaning, then, for such a system at the exhaustive search of the whole encryption key field and a key of the harm the harmed texts have the only conscious text which equals a plaintext provided that the length of a ciphertext is larger than unicity distance. Effectively conscious texts give a fidelity criterion of the found keys.

But if the length of a ciphertext is smaller than unicity distance, then, such keys can be found which can give several conscious texts for the given ciphertext. Their number is defined as $2^{H(K)-BL_0}$ [51, 35].

In the Table 1.2 there are data of the values U_0 for

$$H(K_{\mathrm{D}}^{(i)}) = H(K_i) = 128 \text{ and } H(K_{\mathrm{D}}^{(i)}) = H(K_i) = 256$$

at $m = 10 \ldots 15$ steps of the first method of harming (harming a plaintext with further encryption).

Tabl. 1.2: *Values U_0 (of texts symbols) for the first method of harming*

Length of key	Number of steps						
	0	10	11	12	13	14	15
128 (1.11)	18,8	329	361,9	394,8	427,7	460,6	493,5
256 (1.11)	37,6	376	413,6	451,2	488,8	526,4	564

In case of concealing a the harmed ciphertext Y_{DT} all its possible values determine an additional key field and

$$U_0 = \frac{H(Y_{\mathrm{DT}}) + \sum_{i=1}^{m}(H(K_{\mathrm{D}}^{(i)}) + H(K_i))}{B \log |I|}. \qquad (1.12)$$

In case of additional concealment of the last ciphertext of the harm $Y_{\mathrm{D}}^{(m)}$ due to its small size and commensurability with a ciphertext of the harmed Y_{DT} unicity distance can be additionally enlarged:

$$U_0 = \frac{H(Y_{\mathrm{D}}^{(m)}) + H(Y_{\mathrm{DT}}) + \sum\limits_{i=1}^{m}(H(K_{\mathrm{D}}^{(i)}) + H(K_i))}{B \log |I|}. \qquad (1.13)$$

In the Table 1.3 there are values U_0, computed according to the formulae (1.12) and(1.13), for different number of steps provided that $H(Y_{\mathrm{D}}^{(m)}) = H(Y_{\mathrm{DT}}) = 1024$ and $H(K_i) = H(K_{\mathrm{D}}^{(i)})$, for the values $H(K_i)$, which are equal to 128 and 256.

We shall consider the second method of harming – harming a ciphertext. In this case we have a set of a harmed ciphertext and m harms of this ciphertext, at that all of them separately don't correspond to the conscious plaintext. At the totality of the harmed ciphertext and all its harms increasing of unicity distance takes place at the expense of an additional key of harming a ciphertext.

For this method we can obtain an increased in comparison with the first method unicity distance due to additional

Tabl. 1.3: *Values U_0 (of text symbols) for the first method of harming in case of concealment of Y_{DT} (1.12) and in case of additional concealment of $Y_{\mathrm{D}}^{(m)}$ (1.13)*

Length of key	Number of steps						
	0	10	11	12	13	14	15
128 (1.12)	18,8	359,6	392,5	425,4	458,3	491	524
256 (1.12)	37,6	406,5	444	481,8	519,4	557	992,1
128 (1.13)	18,8	390,1	423	456	488,8	521,7	554,6
256 (1.13)	37,6	437,1	474,7	512,3	549,9	587,5	625,1

encryption of the plaintext by the key K_e at other equal conditions:

$$U_0 = \frac{H(K_e) + \sum_{i=1}^{m}(H(K_{\mathrm{D}}^{(i)}) + H(K_i))}{B \log |I|}; \tag{1.14}$$

$$U_0 = \frac{H(K_e) + H(Y_{\mathrm{DT}}) + \sum\limits_{i=1}^{m}(H(K_{\mathrm{D}}^{(i)}) + H(K_i))}{B \log |I|}; \tag{1.15}$$

$$U_0 = \frac{H(K_e) + H(Y_{\mathrm{DT}}) + H(Y_D^{(m)}) + \sum\limits_{i=1}^{m}(H(K_{\mathrm{D}}^{(i)}) + H(K_i))}{B \log |I|}. \tag{1.16}$$

Values U_0, calculated according to the expressions (1.14) – (1.16), provided that $H(K_e) = H(K_i) = H(K_{\mathrm{D}}^{(i)})$, for $H(K_e)$, which are equal to 128 and 256, are presented in the Table 1.4.

We shall consider the third method of harming – harming a text CHT. This method gives an even larger unicicty distance of the totality of harmed texts in comparison with the first and the second method due to the harm of a plaintext and the text CHT. For this method the following expression is true (1.17).

Values U_0, calculated according to the formula (1.17), provided that $H(K_e) = H(K_i) = H(K_{Di}) = 256$, are presented in the table 1.5.

$$U_0 = \frac{H(K_e) + H(Y_{\mathrm{D}}^{(m_1)}) + H(Y_{\mathrm{DT}})}{B \log |I|} + \\ + \frac{\sum_{i=1}^{m_1} (H(K_{1\mathrm{D}}^{(i)}) + H(K_{1i}))}{B \log |I|} + \\ + \frac{\sum_{j=1}^{m_2} (H(K_{2\mathrm{D}}^{(j)}) + H(K_{2j}))}{B \log |I|}. \tag{1.17}$$

Tabl. 1.4: *Values U_0 (of text symbols) for the second method of harming*

Length of key	Number of steps						
	0	10	11	12	13	14	15
128 (1.14)	18,8	345,5	378,4	411,2	444,1	477	510
256 (1.15)	37,6	418,3	455,9	493,5	531,1	568,7	606,3
256 (1.16)	37,6	441,8	479,4	517	554,6	592,2	629,8

Tabl. 1.5: *Values U_0 (of text symbols) for the third method of harming*

Length of key	Number of rounds ($m_1 = m_2$)						
	0	10	11	12	13	14	15
256 (1.17)	37,6	817,8	893	968,3	1043,5	1118,7	1193,9

The last result in the Table 1.5 corresponds to a hypothetical encryption system with a key of 2805 bits length!

Thus, we come to the idea of multichannel cryptography based on splitting a ciphertext into harmed texts. Such a

splitting doesn't enable a cryptanalyst to obtain an initial ciphertext to estimate unicity distance and get a conscious text by manipulating keys in case of concealing a harmed text or at least one of the harms.

Definition 1.7 *We shall call a transformation of plaintexts at which two or more harmed ciphertexts possessing a cryptographic feature of secret splitting are formed as a multichannel cryptographic transformation:*

- *any incomplete set of harmed texts doesn't give the possibility (maybe except for the brute force attack of the missing harmed texts) to decrypt a received message;*
- *a complete set of harmed texts transforms a task of decryption into a classical task of system breaking at a known ciphertext with as large as possible key field at the expense of the key field of the mechanism of harming.*

1.5 Concealment and encryption of harmed texts

In Greek steganography means "cryptographic writing", when a message itself is hidden from eyes of strangers. The message is built into a thing or a message (container) available for everyone and is transmitted openly to an addressee. The addressee knows how to extract the concealed information from the container.

Unlike steganography cryptography doesn't conceal the fact of a secret message. It makes the message meaningless or difficult to understand for strangers by using an unknown language of a ciphertext.

It's quite natural to combine these two styles to increase security of the concealed information, for instance, first we encrypt information and then conceal its presence in a container which we send. But whereas cryptographic methods can quickly process large amounts of information and transmit it via open channels, steganography mostly operates with small amounts. This limitation doesn't allow using steganographic methods of concealment in many applications.

At the same time the stated in this book approach to multichannel cryptographic transformations with secret splitting and a data pump with manageable length allows to have a different look at interaction between these two directions. Indeed, if in the result of a cryptographic transformation we can get cryptographically harmed ciphertext of a small length, then there's a real possibility of concealing them in conscious text messages which are containers themselves.

In other words an open conscious text plays the role of a ciphertext! This task, which is a reverse one as regards encryption, is probably unsolvable for modern cryptanalysis, as at corresponding rules of key choice it goes back to perfect secrecy encryption systems.

As harmed texts have a manageable size then, it's advisable to make their size in such a way that would allow manipulating them to reach a goal of a specific application. For instance, to conceal or affirm them with a digital signature, store on carriers with a small memory capacity and so on. Cryptographically harmed texts of small sizes is an unconscious random sequence. Further it's advisable to encrypt this sequence in such a way that it would satisfy cryptographic avalanche criteria [82], that improves its statistical properties after encryption, and conceal.

Steganography deals with questions of information concealment. We shall not consider classical methods connected with

musical and video containers here. We would like to hide harmed texts of a small size in a text in such a way that the text itself wouldn't arouse suspicion of containing other information besides the observed one.

Definition 1.8 *A conscious text which conceals a ciphertext is called a text-container.*

Definition 1.9 *We shall call a text-container which has the same size as a concealed ciphertext as an ideal text-container.*

We shall further consider that a binary random sequence which has no meaning goes to the input of such a steganographic system. In the output of such a system we should obtain a conscious, prepared in advance text which plays a role of an ideal container. It's obvious that the length of this text should be equal to the length of the harmed text. But as the length of the harmed text is much smaller than the plaintext, this requirement is not difficult to perform. Let us have some information presented in the form of tuple of zeros and ones. We shall assume that it is measured by the integer number of bytes. If it's not true, then it's always possible to augment it with zeros or ones till the integer number of bytes and add one more auxiliary byte with an indication of the number of added bits.

The general scheme of such a scheme is presented in Fig. 1.7.

At the stage of preparation for concealment of a harmed text Y_{DT} a desired ideal text-container is formed M_s and the key of concealment is defined:

$$K_S = Y_{\mathrm{DT}} \oplus M_s.$$

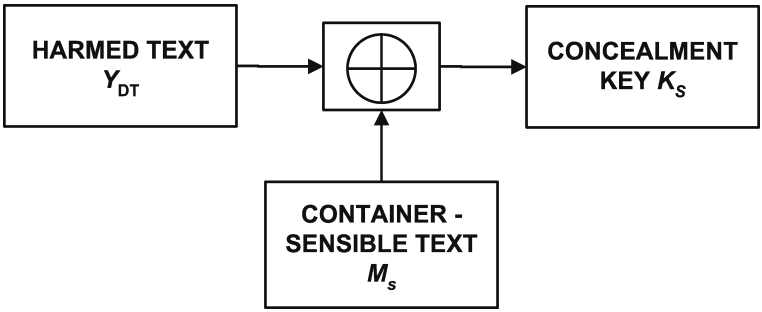

Fig. 1.7: *Key generation*

With the help of this text the harmed text Y_{DT} :

$$Y_{\mathrm{DT}} \oplus K_S = M_s.$$

is processed.

Thus, in the output we have a conscious text M_s, which is an ideal text-container (Fig. 1.8).

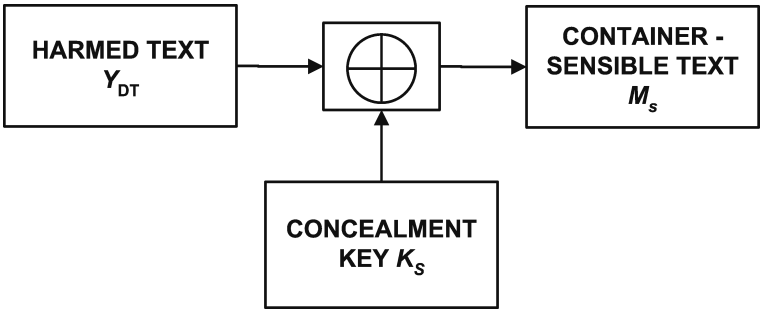

Fig. 1.8: *Forming a conscious text* M_s

As the length of a harmed text can be small, then its additional encryption by an asymmetric system allows to solve problems of a digital signature or to improve security of an encryption system on the whole.

Any byte of information can be recoded into any byte of the ASCII table according to some code key in accordance with the expression $a_i \oplus K_{ij}$, where a_i – a replaceable i-th symbol of the ASCII table, K_{ij} – a code key of substitution of the i-th symbol for the j-th symbol, a_j – j-th symbol of the ASCII table (Appendix D).

A universal method of concealment of any digital information includes the following algorithm:

1. Source information is recorded in form of zeros and ones.
2. Conscious open information with a corrupted meaning of the same size in form of zeros and ones is recorded (a text-container in form of an ideal text container).
3. Code keys are defined byte by byte by congruence addition 2.

Reading of concealed information takes place in the reverse order:

1. Received open conscious information with a corrupted meaning is written.
2. Code keys are recorded.
3. Source information is defined by congruence addition 2.

Peculiarity of this universal method is that another conscious text of an ideal text-container serves as a container, that doesn't allow a cryptanalyst suspect a substitution.

Example 1.3 *Let a container for the message should be organized:*

"Встречайте 27-го на второй платформе в 7 вечера ".

We form an ideal text-container in form of a conscious phrase:

"Сообщаем адрес Института: Минск, ул. Филимонова,69 ".

We shall show how the algorithm works на слове "Meet".

We define a code key:

The code of the word "Встречайте" (10 symbols) :

11000010(В)	*11110001(с)*	*11110010(т)*	*11110000(р)*	*11100101(е)*
11110111(ч)	*11100000(а)*	*11101000(й)*	*11110010(т)*	*11100101(е)*

10 symbols of a part of the container "Сообщаем а":

11010001(С)	*11101110(о)*	*11101110(о)*	*11100001(б)*	*11111000(щ)*
11100000(а)	*11100101(е)*	*11101100(м)*	*10100000(' '- пробел)*	*11100000(а)*

We form the code key byte by byte:

Message:

11000010(В)	*11110001(с)*	*11110010(т)*	*11110000(р)*	*11100101(е)*
11110111(ч)	*11100000(а)*	*11101000(й)*	*11110010(т)*	*11100101(е)*

Ideal text-container:

11010001(С)	*11101110(о)*	*11101110(о)*	*11100001(б)*	*11111000(щ)*
11100000(а)	*11100101(е)*	*11101100(м)*	*10100000(' ')*	*11100000(а)*

Key:

000100011	*000101111*	*00011111*	*00010001*	*00011101*
(В⊕С)	*(с⊕о)*	*(т⊕о)*	*(р⊕б)*	*(е⊕щ)*
00010111	*00000101*	*00000100*	*01010010*	*00000101*
(ч⊕а)	*(а⊕е)*	*(й⊕м)*	*(т⊕' ')*	*(е⊕а)*

Obtaining of an ideal text-container:

Message "Встречайте":

11000010(В)	*11110001(с)*	*11110010(т)*	*11110000(р)*	*11100101(е)*
11110111(ч)	*11100000(а)*	*11101000(й)*	*11110010(т)*	*11100101(е)*

Key:

000100011	*000101111*	*00011111*	*00010001*	*00011101*
(В⊕С)	*(с⊕о)*	*(т⊕о)*	*(р⊕б)*	*(е⊕щ)*
00010111	*00000101*	*00000100*	*01010010*	*00000101*
(ч⊕а)	*(а⊕е)*	*(й⊕м)*	*(т⊕' ')*	*(е⊕а)*

Ideal text-container: "Сообщаем а":

11010001(С)	*11101110(о)*	*11101110(о)*	*11100001(б)*	*11111000(щ)*
11100000(а)	*11100101(е)*	*11101100(м)*	*10100000(' ')*	*11100000(а)*

Key:

000100011	*000101111*	*00011111*	*00010001*	*00011101*
(В⊕С)	*(с⊕о)*	*(т⊕о)*	*(р⊕б)*	*(е⊕щ)*
00010111	*00000101*	*00000100*	*01010010*	*00000101*
(ч⊕а)	*(а⊕е)*	*(й⊕м)*	*(т⊕' ')*	*(е⊕а)*

Message "Встречайте":

11000010(В)	*11110001(с)*	*11110010(т)*	*11110000(р)*	*11100101(е)*
11110111(ч)	*11100000(а)*	*11101000(й)*	*11110010(т)*	*11100101(е)*

As in the given example we transfer letters of the Russian alphabet into letters of the Russian alphabet and at that the

letter "й" can be replaced by the letter "и", and use the letter "й" as a space, then key optimization will be advantageous in comparison with the transmitted information. In our case we can do with a shorter key:
00000101 (a service byte pointing at 5 younger meaningful digits of the byte which follows it) 10001(ВС) 11111(со) 11100(то) 10001(рб) 11100(ещ) 10111(ча) 00101(ае) 00100(им) 11011(тй) 00101(еа)=
= 0000010110001111111110010001111001011100101001001101100101
=58 bits.

A source key contained 80 bits.

Concealment of a ciphertext in form of a container – a conscious text – is of great interest. In this case the whole ASCII table is involved in computing a key, though translating takes place in one of the communication languages.

Example 1.4 *The ciphertext "Ђ;*8as?{6f€9h‡n43,cqN'is given. We should present it in a form of a conscious English text: "I am going to a club.".*

We act like in the previous example:
*Ciphertext: "Ђ;*8as?{6f€9h‡n43,cqN":*

10000000	*00111011*	*00101010*
00111000	*01100001*	*01110011*
00111111	*01111110*	*00110110*
01100110	*10001000*	*00111001*
01101000	*10000111*	*01101110*
00110100	*00110011*	*00101100*
01100011	*01110001*	*01001111*

Ideal text-container: "I am going to a club.":

01001001	*10100000*	*01100001*
01101101	*10100000*	*01100111*
01101111	*01101001*	*01101110*
01100111	*10100000*	*01110100*
01101111	*10100000*	*01100001*
10100000	*01100011*	*01101100*
01110101	*01100010*	*10110111*

———————————————————————-

Key:

11001001	*10011011*	*01001011*
01010101	*100000001*	*00010100*
01100000	*00010111*	*01011000*
00000001	*00101000*	*01001101*
00000111	*00100111*	*00001111*
10010100	*01010000*	*01000000*
00010100	*00010011*	*11111000*

Obtaining of an ideal text-container:

Ciphertext:	*10000000(Ъ)*	*00111011(;)*	*00101010(*)*
	00111000(8)	*01100001(a)*	*01110011(s)*	
	00111111(?)	*01111110(~)*	*00110110(6)*
	01100110(f)	*10001000(€)*	*00111001(9)*
	01101000(h)	*10000111(‡)*	*01101110(n)*
	00110100(4)	*00110011(3)*	*00101100(,)*
	01100011(c)	*01110001(q)*	*01001111(N)*

Key :	*11001001*	*10011011*	*01001011*
	01010101	*100000001*	*00010100*
	01100000	*00010111*	*01011000*
	00000001	*00101000*	*01001101*
	00000111	*00100111*	*00001111*
	10010100	*01010000*	*01000000*
	00010100	*00010011*	*11111000*

Ideal	*01001001(I)*	*10100000(пр)*	*01100001(a)*
text-	*01101101(m)*	*10100000(пр)*	*01100111(g)*	
container:	*01101111(o)*	*01101001(i)*	*01101110(n)*
	01100111(g)	*10100000(' ')*	*01110100(t)*
	01101111(o)	*10100000(' ')*	*01100001(a)*
	10100000(' ')	*01100011(c)*	*01101100(l)*
	01110101(u)	*01100010(b)*	*10110111(.)*

Steganoanalysis becomes completely helpless if we use a conscious text in form of an ideal container as a container.

Disadvantage of this method is that the length of the keys is equal to the length (though not large) of a concealed text, and uniqueness of session keys. Therefore, if you can secretly transmit keys of the size of a concealed text this

method becomes quasi senseless. But if you have a possibility to transmit an ideal text-container via one open channel, and a key in form of a ciphertext via another open channel, in many applications it can play a significantly positive part. At that one of these transmissions is a conscious unconcealed text which doesn't arouse suspicions. For the purpose of masking the length of a text-container can be larger than the length of a concealed ciphertext. For more masking a ciphertext can be "immersed" into a voluminous text-container by the indicated method.

1.6 Harmed texts and data cores. Unicity distance for harmed ciphertexts

In the result of applying the universal mechanism of harming C_m to a text we obtain several information channels: a channel of a harmed text and a set of harms' channels. Peculiarity of this method is that it's manageable by the length of a harmed text which depends on the number of steps m of a cyclic transformation and is approximately equals $\frac{L_0}{\eta^m}$, where L_0 – a length of a plaintext, $\eta > 1$ – some constant coefficient.

In a specific application the number of steps of the mechanism C_m can be fixed, and then the length of a harmed text is defined by the length of the plaintext L_0 only. In other applications the length of a harmed text can be fixed, and then a manageable parameter is a number of steps m.

Definition 1.10 *We shall call a harmed text* Y_{DT},

obtained by using the universal mechanism of harming C_m round by round as a data-pump of a text.

This definition is justified as restoring a plaintext *always* begins with a harmed text Y_{DT} – the core and its length $L(Y_{\mathrm{DT}}) \approx \dfrac{L_0}{\eta^m}$. For the given plaintext of the length L_0 the length of a data-pump is variable and depends on the parameter m, which, it its turn, is determined by a specific application. If necessary, for bigger values L_0 the value $L(Y_{\mathrm{DT}})$ can be made small enough by managing the number of steps of the mechanism C_m.

It's obvious that concealment of the core, due to its small size, or its deliberate corruption leads to disastrous consequences for restoring source information, as to restore it you need to know all the three attributes: the core, harms and transformation parameters. Therefore, unlike hash-functions the core restores information with the help of harms and transformation parameters with zero probability of collisions.

A concealed core plays a part of an additional long key which can be specified only by search. This additional key depends on a plaintext and transformation parameters (secret keys and the number of rounds of an algorithm). Therefore, cryptographic systems built on the basis of concealed or corrupted cores are more secure than practically secure systems. This assumption is based on a sharp increase of Shannon's unicity distance U_0 (see 1.4), that is a positive moment in general, especially for short texts, and on the following reasons.

Let us have two harmed ciphertexts in the output of a two-channel encryption system: a ciphertext of a harmed text in form of the core Y_{DT} and a ciphertext of harms Y_{D}. Then, let the short ciphertext Y_{DT} be concealed and a cryptanalyst observes the ciphertext Y_{D}. only. Then, according to C.

Shannon, the unicity distance will increase by the value $\frac{H(Y_{\mathrm{DT}})}{B}$ and

$$U_0 = \frac{H(K)}{B} + \frac{H(Y_{\mathrm{DT}})}{B}.$$

If the length of the ciphertext of the harms $L_{\mathrm{D}} > U_0$, then there is only one conscious text which has the ciphertext Y_{D}. But let's remember that the ciphertext Y_{D} was derived from a meaningless text and, therefore, even if there is such a conscious text, it *doesn't correspond to the criterion of key finding* . Shannon's theory of unicity distance is true when a ciphertext is obtained with the help of the cryptographic transformation: conscious text – ciphertext. Therefore, we come to the idea of another definition for unicity distance for harmed texts.

Claim 1.1 *If a ciphertext is harmed, then it can happen that there is not a single conscious text which corresponds to the given harmed text.*

If we find conscious texts which correspond to the given harmed ciphertext, these texts are not true.

It's equivalent to infinite unicity distance, that corresponds to the definition of the ideal system (according to Shannon) if the exhaustive search of a concealed core isn't carried out.

1.7 Nine structural encryption schemes based on harmed texts

Different methods of harming together with methods of further concealment or corruption of data-pumps create different structural encryption schemes. A method of harming

without concealment of ciphertexts brings forth a structure showed in Fig. 1.9. For this method encryption and

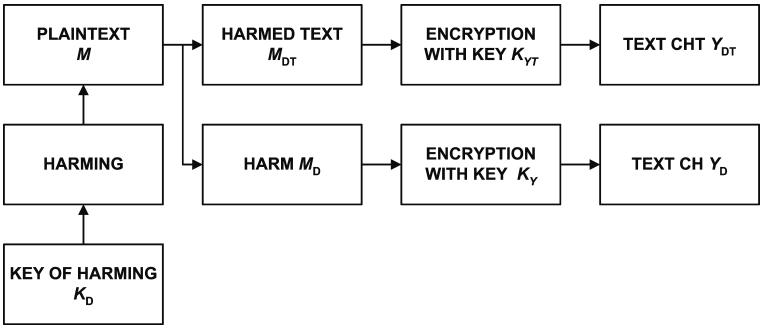

Fig. 1.9: *The first method of harming with observable texts* CHT *and* CH

decryption processes are described by the following equations:

– encryption process:

$$\begin{aligned} M_{\mathrm{DT}} &= D_1(M, K_{\mathrm{D}}); \\ M_{\mathrm{D}} &= D_2(M, K_{\mathrm{D}}); \\ Y_{\mathrm{DT}} &= E_1(M_{\mathrm{DT}}, K_{YT}); \\ Y_D &= E_2(M_{\mathrm{D}}, K_Y); \end{aligned}$$

– decryption process:

$$\begin{aligned} M_D &= E_2^{-1}(Y_{\mathrm{D}}, K_Y); \\ M_{\mathrm{DT}} &= E_1^{-1}(Y_{\mathrm{DT}}, K_{YT}); \\ M &= D_{1\ 2}^{-1}(M_{\mathrm{DT}}, M_D, K_{\mathrm{D}}) \end{aligned} \quad .$$

Informational outputs of the system: ciphertexts Y_{DT} and Y_{D}.

Pay attention that encryption and decryption processes are realizes by different algorithms.

In this structure we evidently have texts and they are not concealed from a cryptanalyst. A key field of such a structure

is evaluated as $K_{\mathrm{D}} \times K_{YT} \times K_Y$, and unicity distance on every from the texts CHT and CH equals infinity due to the statement 1.1. There's a direct evidence of the ideal system in the interpretation of C. Shannon [29], for every ciphertext. At observing the both ciphertexts unicity distance is

$$U_0 = \frac{H(K_{\mathrm{D}}) + H(K_{YT}) + H(K_Y)}{B}.$$

The second structure of this method is based on concealment of the text HCT in form of a conscious text (Fig. 1.10). A cryptanalyst needs to find a ciphertext according to a conscious text! This task seems to be practically impossible to accomplish.

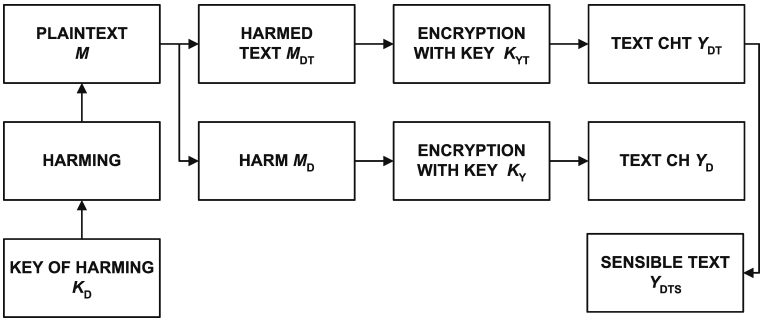

Fig. 1.10: *The first method of harming with concealment of a text* CHT*in a text container*

In this case a cryptanalyst observes the text CH only. The system is ideal according to C. Shannon.

For this structure the following equations are true:

– encryption process:

$$\begin{aligned}
&M_{\mathrm{DT}} = D_1(M, K_{\mathrm{D}});\\
&M_{\mathrm{D}} = D_2(M, K_{\mathrm{D}});\\
&Y_{\mathrm{DT}} = E_1(M_{\mathrm{DT}}, K_{YT});\\
&Y_{\mathrm{D}} = E_2(M_{\mathrm{D}}, K_Y);\\
&Y_{\mathrm{DTS}} = Y_{\mathrm{DT}} \oplus K_S;
\end{aligned}$$

– decryption process:

$$Y_{\mathrm{DT}} = Y_{DTS} \oplus K_S;$$
$$M_{\mathrm{D}} = E_2^{-1}(Y_{\mathrm{D}}, K_Y);$$
$$M_{\mathrm{DT}} = E_1^{-1}(Y_{\mathrm{DT}}, K_{YT});$$
$$M = D_{1\ 2}^{-1}(M_{\mathrm{DT}}, M_{\mathrm{D}}, K_{\mathrm{D}}).$$

Informational outputs of the system: ciphertexts Y_{DTS} and Y_{D}.

The third structure of this method (Fig. 1.11) is based on irretrievable corruption of a harmed text CHT, which is attained by applying a perfect secrecy encryption system due to its small length. It leads to impossibility to find out a ciphertext of a harmed text. The system is ideal according to C. Shannon.

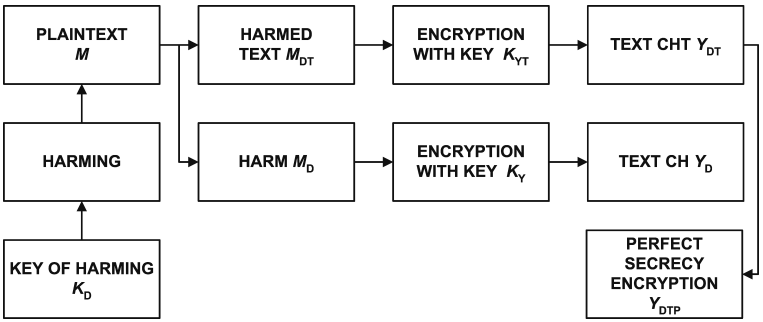

Fig. 1.11: *The first method of harming with perfect secrecy encryption of the text CHT*

For this structure the following equations are true:
– encryption process:

$$M_{\mathrm{DT}} = D_1(M, K_{\mathrm{D}});$$
$$M_{\mathrm{D}} = D_2(M, K_{\mathrm{D}});$$
$$Y_{\mathrm{DT}} = E_1(M_{\mathrm{DT}}, K_{YT});$$
$$Y_{\mathrm{D}} = E_2(M_{\mathrm{D}}, K_Y);$$
$$Y_{\mathrm{DTP}} = Y_{\mathrm{DT}} \oplus K_P;$$

– decryption process:

$$Y_{\mathrm{DT}} = Y_{\mathrm{DTP}} \oplus K_p;$$
$$M_{\mathrm{DT}} = E_1^{-1}(Y_{\mathrm{DT}}, K_{YT});$$
$$M_{\mathrm{D}} = E_2^{-1}(Y_{\mathrm{D}}, K_Y);$$
$$M = D_{1\,2}^{-1}(M_{\mathrm{DT}}, M_{\mathrm{D}}, K_{\mathrm{D}}).$$

Informational outputs of the system: ciphertext Y_{DTP} and Y_{D}.

The second method of harming without concealment of ciphertexts creates a structure showed in Fig. 1.12.

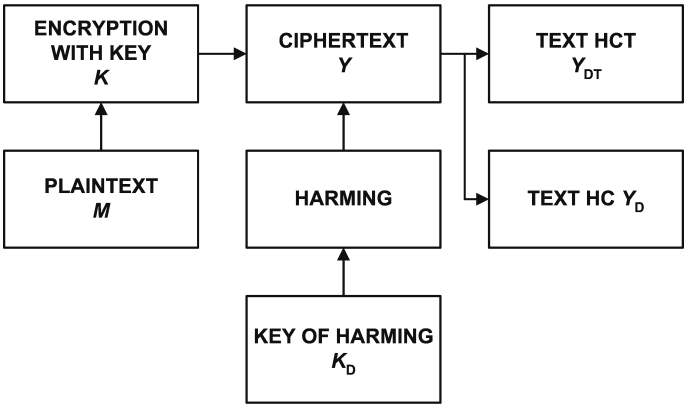

Fig. 1.12: *The second method of harming a ciphertext with observable texts* HCT *and* HC

For this structure encryption and decryption processes are described by the following equations:

– encryption process:

$$Y = E(M, K);$$
$$Y_{\mathrm{DT}} = D_1(Y, K_{\mathrm{D}});$$
$$Y_{\mathrm{D}} = D_2(Y, K_{\mathrm{D}});$$

– decryption process:

$$Y = D_{1\,2}^{-1}(Y_{\mathrm{DT}}, Y_{\mathrm{D}}, K_{\mathrm{D}});$$
$$M = E^{-1}(Y, K).$$

Informational outputs of the system: ciphertexts Y_{DT} and Y_D.

Here a cryptanalyst can observe two harmed texts HCT and HC, each of them having no conscious input text. At that unicity distance is defined as

$$U_0 = \frac{H(K) + H(K_{\mathrm{D}})}{B}.$$

This scheme is equivalent to the one showed in Fig. 1.11, possibly with a smaller key field.

Like in the above case we can obtain two more variants of the method implementation due to concealment or theoretically secure encryption of a harmed ciphertext (Fig. 1.13 and 1.14).

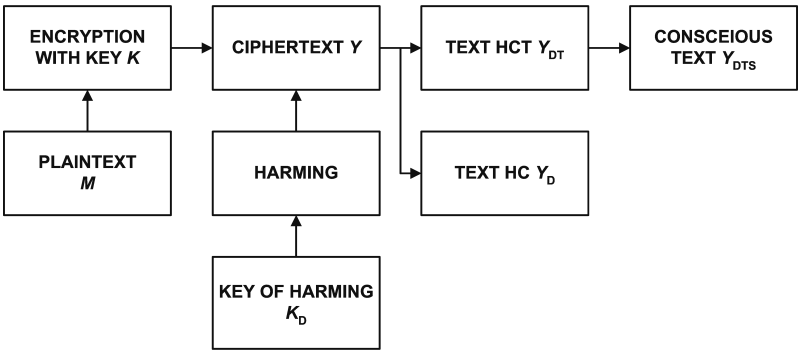

Fig. 1.13: *The second method of harming with concealment of a text* HCT *in a container of a conscious text*

For the structure showed in Fig. 1.13, the following is true:
– encryption equation:

$$\begin{aligned} Y &= E(M, K); \\ Y_{\mathrm{DT}} &= D_1(Y, K_{\mathrm{D}}); \\ Y_{\mathrm{D}} &= D_2(Y, K_{\mathrm{D}}); \\ Y_{\mathrm{DTS}} &= Y_{\mathrm{DT}} \oplus K_S; \end{aligned}$$

– decryption equation:

$$Y_{\mathrm{DT}} = Y_{\mathrm{DTS}} \oplus K_S;$$
$$Y = D_{1\ 2}^{-1}(Y_{\mathrm{DT}}, Y_{\mathrm{D}}, K_{\mathrm{D}});$$
$$M = E^{-1}(Y, K).$$

Informational outputs of the system: ciphertexts Y_{DTS} and Y_{D}.

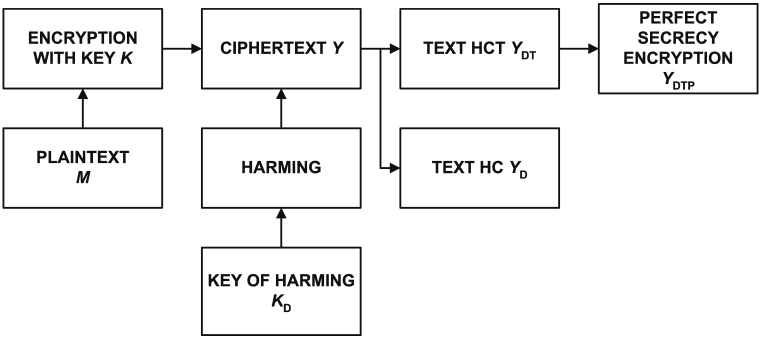

Fig. 1.14: *The second method of harming with perfect secrecy encryption of a text* HCT

For the structure showed in Fig. 1.14, the following is true:
– encryption equation:

$$Y = E(M, K);$$
$$Y_{\mathrm{DT}} = D_1(Y, K_{\mathrm{D}});$$
$$Y_{\mathrm{D}} = D_2(Y, K_{\mathrm{D}});$$
$$Y_{\mathrm{DTP}} = Y_{\mathrm{DT}} \oplus K_P;$$

– decryption equation:

$$Y_{\mathrm{DT}} = Y_{\mathrm{DTP}} \oplus K_P;$$
$$Y = D_{1\ 2}^{-1}(Y_{\mathrm{DT}}, Y_{\mathrm{D}}, K_{\mathrm{D}});$$
$$M = E^{-1}(Y, K).$$

Informational outputs of the system: ciphertexts Y_{DTP} and Y_{D}.

In spite of apparent identity of the schemes which use perfect secrecy encryption and concealment of a harmed ciphertext in a container of a conscious text, the last schemes additionally bear the duty of non-identifying presence of a ciphertext at all.

The third method is a combination of the first two: here we harm a plaintext and a ciphertext. It generates schemes with large key fields and large unicity distances if necessary. Below there are three variants of schemes of this method (Fig. 1.15 – 1.17).

The structure showed in Fig. 1.15 is described by the following equations:

– encryption process:

$$
\begin{aligned}
&M_{\mathrm{DT}} = D_1(M, K_{D^1});\\
&M_{\mathrm{D}} = D_2(M, K_{\mathrm{D}}^{(1)});\\
&Y_{\mathrm{DT}} = E_1(M_{\mathrm{DT}}, K_{YT});\\
&Y_{\mathrm{D}} = E_2(M_{\mathrm{D}}, K_Y);\\
&Y_{\mathrm{DDT}} = D_3(Y_{\mathrm{D}}, K_{\mathrm{D}}^{(2)});\\
&Y_{\mathrm{DD}} = D_4(Y_{\mathrm{D}}, K_{\mathrm{D}}^{(2)}).
\end{aligned}
$$

– decryption process:

$$
\begin{aligned}
&Y_{\mathrm{D}} = D_{3\ 4}^{-1}(Y_{\mathrm{DD}}, Y_{\mathrm{DDT}}, K_{\mathrm{D}}^{(2)});\\
&M_{\mathrm{D}} = E_2^{-1}(Y_{\mathrm{D}}, K_Y);\\
&M_{\mathrm{DT}} = E_1^{-1}(Y_{\mathrm{DT}}, K_{YT});\\
&M = D_{1\ 2}^{-1}(M_{\mathrm{D}}, M_{\mathrm{DT}}, K_{\mathrm{D}}^{(1)}).
\end{aligned}
$$

It's interesting to note that the system structure showed in Fig. 1.15, gives three information channels Y_{DT}, Y_{DDT} and Y_{DD}, in two of them there are harmed texts of a small length. It gives additional degrees of freedom at building other systems and technologies using cryptographic tools.

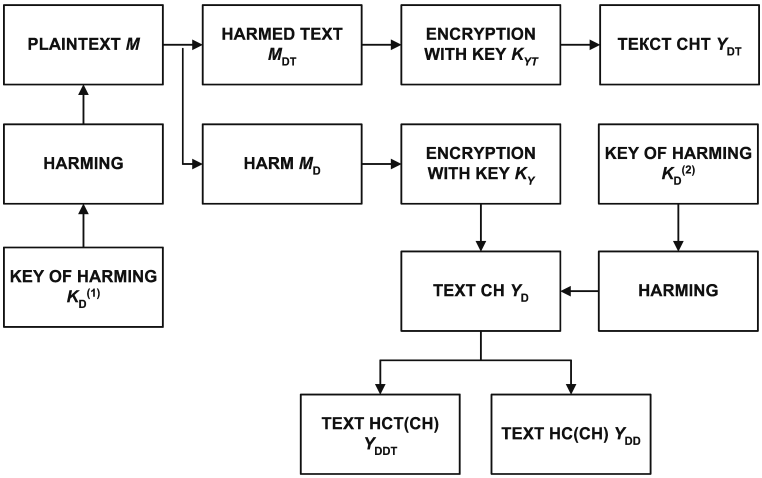

Fig. 1.15: *Implementation of the combined method of harming with observable output texts* CHT

The structure of Fig. 1.16 is described by the equations:
– encryption process:

$$
\begin{aligned}
&M_{\mathrm{DT}} = D_1(M, K_{\mathrm{D}}^{(1)});\\
&M_{\mathrm{D}} = D_2(M, K_{\mathrm{D}}^{(1)});\\
&Y_{\mathrm{DT}} = E_1(M_{\mathrm{DT}}, K_{YT});\\
&Y_{\mathrm{D}} = E_2(M_{\mathrm{D}}, K_Y);\\
&Y_{\mathrm{DDT}} = D_3(Y_{\mathrm{D}}, K_{\mathrm{D}}^{(2)});\\
&Y_{\mathrm{DD}} = D_4(Y_{\mathrm{D}}, K_{\mathrm{D}}^{(2)});\\
&Y_{\mathrm{DDS}} = Y_{\mathrm{DD}} \oplus K_S;
\end{aligned}
$$

– decryption process:

$$
\begin{aligned}
&Y_{\mathrm{DD}} = Y_{\mathrm{DDS}} \oplus K_S;\\
&Y_{\mathrm{D}} = D_{3\ 4}^{-1}(Y_{\mathrm{DD}}, Y_{\mathrm{DDT}}, K_{\mathrm{D}}^{(2)});\\
&M_{\mathrm{DT}} = E_1^{-1}(Y_{\mathrm{DT}}, K_{YT});\\
&M_{\mathrm{D}} = E_2^{-1}(Y_{\mathrm{D}}, K_Y);\\
&M = D_{1\ 2}^{-1}(M_{\mathrm{D}}, M_{\mathrm{DT}}, K_{\mathrm{D}}^{(1)}).
\end{aligned}
$$

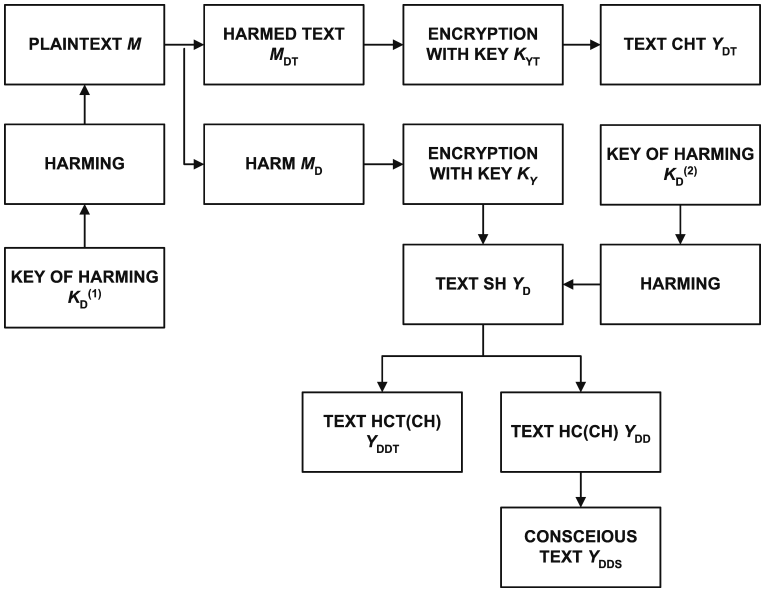

Fig. 1.16: *Implementation of the combined method of harming with concealment of a text* HC(CH) *in a container of a conscious text*

Informational outputs of the system: ciphertexts Y_{DT}, Y_{DDT} and Y_{DDS}.

The structure of Fig. 1.17 is described by the equations:

– encryption process:

$$
\begin{aligned}
&M_{\mathrm{DT}} = D_1(M, K_{\mathrm{D}}^{(1)});\\
&M_{\mathrm{D}} = D_2(M, K_{\mathrm{D}}^{(1)});\\
&Y_{\mathrm{DT}} = E_1(M_{\mathrm{DT}}, K_{YT});\\
&Y_{\mathrm{D}} = E_2(M_{\mathrm{D}}, K_Y);\\
&Y_{\mathrm{DDT}} = D_3(Y_{\mathrm{D}}, K_{\mathrm{D}}^{(2)});\\
&Y_{\mathrm{DD}} = D_4(Y_{\mathrm{D}}, K_{\mathrm{D}}^{(2)});\\
&Y_{DDP} = Y_{DD} \oplus K_P;
\end{aligned}
$$

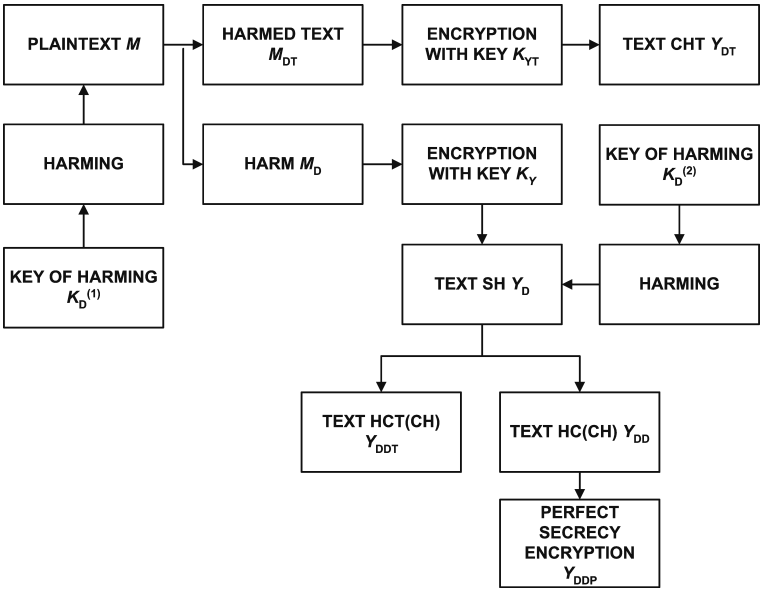

Fig. 1.17: *Implementation of the combined method of harming with perfect secrecy encryption of a text* HC(CH)

– decryption process:

$$
\begin{aligned}
&Y_{\mathrm{DD}} = Y_{\mathrm{DDP}} \oplus K_P;\\
&Y_{\mathrm{D}} = D_{3\ 4}^{-1}(Y_{\mathrm{DD}}, Y_{\mathrm{DDT}}, K_{\mathrm{D}}^{(2)});\\
&M_{\mathrm{DT}} = E_1^{-1}(Y_{\mathrm{DT}}, K_{YT});\\
&M_{\mathrm{D}} = E_2^{-1}(Y_{\mathrm{D}}, K_Y);\\
&M_{\mathrm{DT}} = E_1^{-1}(Y_{\mathrm{DT}}, K_{YT});\\
&M = D_{1\ 2}^{-1}(M_{\mathrm{D}}, M_{\mathrm{DT}}, K_{\mathrm{D}}^{(1)}).
\end{aligned}
$$

Informational outputs of the system: ciphertexts Y_{DT}, Y_{DDT} and Y_{DDP}.

1.8 Summary

Destruction of a meaning of a plaintext or ciphetext in an alphabet of a ciphertext leads to additional key possibilities of an encryption process and increase of Shannon's unicity distance. Actually, this approach itself brings us to the idea of several ciphertexts, each of them having no sense in an alphabet of a ciphertext, and, consequently, doesn't have a conscious plaintext which generates it. It's necessary to mention duality of the analysis of harmed ciphertexts. If a cryptanalyst uses observable harms only, without hypothesis of concealed ciphertexts, he deals with an ideal encryption system. If he builds an analysis based on models of a concealed harmed ciphertext, he deals with an encryption system with a very large unicity distance which is determined by ambiguity of a concealed harmed text. Necessity to have all ciphertexts during encryption sets a very difficult problem of interception when using different channels of information transmission and cryptanalysis in the class of harmed texts.

This new cipher protection breaks Shannon's status of cryptanalysis: "... an adversary has special equipment necessary to intercept and record transmitted signals ". Variety of methods of harming and concealment of harmed texts allows synthesizing new models of cryptographic systems for different purposes with various characteristics and properties.

Chapter 2
Multi-channel cryptography

2.1 Two-channel symmetric encryption algorithm MV2 (first familiarity)

This algorithm is realized according to the second method: harming a ciphertext with observable texts HCT and HC.

A ciphertext is harmed with the help of a universal algorithm of nonuniform substitution with further permutation. This process is cyclic and is repeated with intermediate ciphertexts till the set sizes of a harmed text (data-pump C) are obtained. All intermediate harms called flags F, are united into one information channel where permutation takes place [20].

Thus, we have a classical one round substitution permutation scheme where nonuniform random substitutions with a number of digits smaller than a byte of a source or intermediate harmed text appear as non-linear S-blocks. Actually, this algorithm assures splitting of input information

into two channels: a harmed text's channel (a data-pump) and a harm's channel (auxiliary flags to restore the plaintext). The scheme of this algorithm's implementation is shown in Fig. 2.1. Whitening of a plaintext by a stream cipher breaks statistical dependence that allows obtaining harmed texts at a smaller number of rounds.

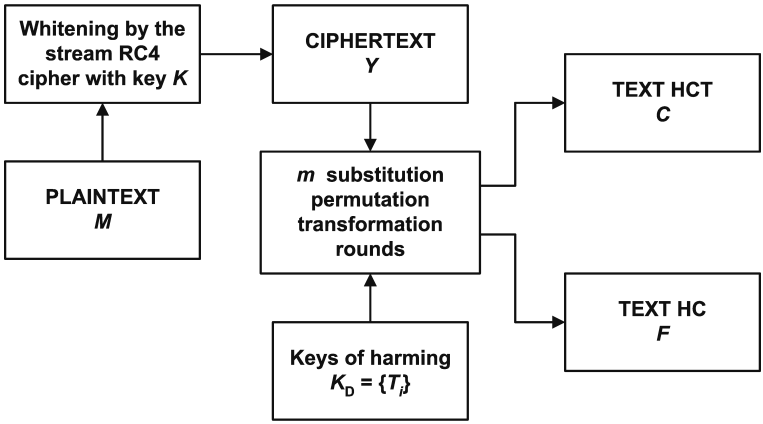

Fig. 2.1: *Scheme of the MV2 algorithm's implementation*

The MV2 algorithm can be used as a basic structure to harm with further encryption for any system of multi-channel cryptography.

In chapter 3 strict mathematical estimations and results of the researches of the algorithm are presented.

2.2 Multi-channel cryptographic transformations

In 1.6 nine cryptographic structures reflecting the methods of obtaining harmed texts and data-pump concealment were

shown. It's practical to present these structures in correlation with the universal mechanism of harming.

In its dynamics the universal method of harming has information in form of intermediate harmed ciphertexts and harm's ciphertexts. For the process of source information restoration intermediate harmed texts are redundant information data, and therefore can be omitted. In all the following should be kept after m steps of the algorithm C_m for the process of source information restoration: the m-th harmed text (data-pump) and m harm's ciphertexts, i.e. altogether we have $m + 1$ information channels. It lets us manipulate degrees of freedom of separate or uniting use of information during engineering. Thus, for instance for $m = 2$ we have one harmed ciphertext HCT and two harm's ciphertexts HC_1 and HC_2, that corresponds to $2 + \binom{3}{2} = 5$ possible combinations of channel information unification:

- one combined channel of the text HCT and the texts HC_1, HC_2;
- a channel of the text HCT and a united channel of the texts HC_1 and HC_2 (two-channel system);
- a united channel of the text HCT with the text HC_1 and a separate channel for the text HC_2 (two-channel system);
- a united channel of the text HCT with the text HC_2 and a separate channel for the text HC_1 (two-channel system);
- three separate channels: the text HCT, the text HC_1 and the text HC_2 (three-channel system).

Use of various combinations is authorized by requirements of practical applications as we deal with texts of various length and can manage the length of a harmed ciphertext by changing the value m, and, consequently, partially change the length of the last harms. One can manage security of a cryptographic system in a whole by combining different information streams with further use of steganography.

Irrespective of the method with the help of which the text HCT and the texts HC_i were obtained we further have a task of manipulating them.

In Fig. 2.2 you can see a general scheme of information manipulating and distributing to information channels after harming a corresponding encryption.

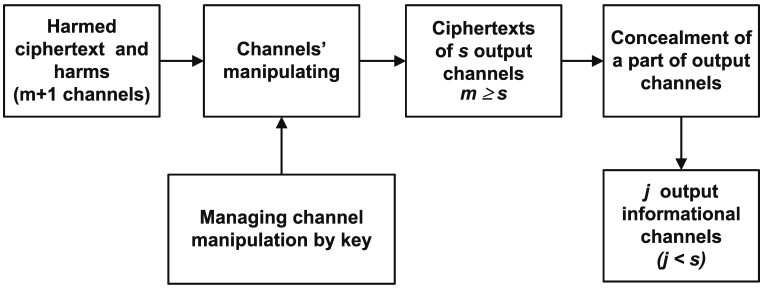

Fig. 2.2: *General scheme of information manipulating and distributing to information channels*

Managing channel manipulation allows uniting information of different harms and a harmed ciphertext into one information channel by a key, that increases general system's key field. As an example manipulation variants for $m = 2$ were given above. A part of informational channels (of a short harmed ciphertext-core HCT and proceeding short texts HC) can be steganographically concealed or uncoverably corrupted by encrypting with the help of a perfect secrecy system. Thus, at using multi-channel encryption a cryptanalyst will

always have a corrupted and non-informative ciphertext which doesn't correspond to the process of an open plaintext encryption.

As a working advice for high security systems we can suggest several practical measures: you can steganographically conceal a harmed text in the perfect container of a sensible text or encrypt it by a perfect secrecy system.

In both cases the true ciphertext of a harmed ciphertext can be considered unavailable for a cryptanalyst.

If it's necessary to have two managing parameters in the application a ciphertext of the last short text HC can be additionally concealed. You can continue doing like that till you get a certain size of the text HCT.

But what can you do if it's necessary to have a bigger number of channels than that obtained with the help of this method? In this case you can build a secondary system of harming a text HC_1. This will double the number of hidden channels and so on.

Let's consider a structure of the cryptographic system MV3 for arbitrary number of rounds m. as an example. The structure of the cryptographic system MV3 can be obtained by various modifications of the MV system.

2.2.1 Variant of little harm

The size of a little harm is close to the size of the text HCT (data-pump). Output information channels are:

✓A harmed ciphertext (data-pump). Concealment is possible;

✓The last text (HC_m). Concealment is possible;

✓The rest of texts HC_i.

This variant is the easiest and fastest one.

2.2.2 Variant of repeated use of the MV2 algorithm to the text HC_1

This variant is a little bit slower, but cryptographically more secure.

Output information channels:

- the text HCT_1 (data-pump) after the first use of the MV2 algorithm. Concealment is possible;
- the text HCT_2 (data-pump) after the second use of the MV2 algorithm. Concealment is possible;
- the rest of texts HC_i.

2.2.3 Variant of repeated use of the MV2 algorithm to the united text $\{\mathrm{HC}_i\}$

This variant is two times slower than the first one.

Output information channels:

- the text HCT_1 (data-pump) after the first use of the MV2 algorithm. Concealment is possible;
- the text HCT_2 (data-pump) after the second use of the MV2 algorithm. Concealment is possible;
- the rest of all texts HC_i after using the second MV2 algorithm.

There are a lot of other possible system modifications implementing the MV3 algorithm. Each of them can be dictated by a certain application.

Let's evaluate speed capabilities of MV3.

In the variant of a little harm $V_{MV3} \approx V_{MV2}$, where V_{MV3} and V_{MV2} are speeds of corresponding algorithms' performance.

In the variant of repeated use of the MV2 algorithm to the first harm it's possible to parallel the process operations performance. As the length of the first harm is considerably smaller than that of the plaintext (see the formula (3.56) in 3.2.4), then at algorithm paralleling the time of its executing is $T_{MV3} \approx T_{MV2}$.

In the variant of a repeated application of the MV2 algorithm to the total harm's ciphertext, the time and speed of its performance are defined by the ratios

$$T_{MV3} \approx 2T_{MV2} \quad \text{and} \quad V_{MV3} \approx \frac{1}{2} \cdot V_{MV2}.$$

2.3 Real symmetric-asymmetric system: MV2 and asymmetric system with an open key

Cryptographic systems with an open key are practically useless for encrypting large amounts of information because of computationally difficult procedures of exponentiation and multiplication. That is why these cryptographic systems are used for encryption during exchange of small session key information, or during creation of an electronic signature. In combination with symmetric systems these systems allow guaranteeing integrity and authenticity during encrypted message exchange. Such systems are called mixes or hybrids. At the same time a joint synthesis of symmetric and asymmetric systems with an open key as systems using one-way functions with a secret considerably increases security

of cryptosystems [54]. Up to now such systems haven't been created. Here we shall formulate requirements for such systems:

- an encryption round must occur simultaneously with the help of secret keys of a symmetric system and open keys of an asymmetric system:

$$Y = E_1(M, K, k_{pb}); \tag{2.1}$$

- a decryption round must occur with the help of secret keys of a symmetric system and those ones of an asymmetric system:

$$M = E_2(Y, K, k_{pr}). \tag{2.2}$$

To compare there's a hybrid scheme below (Fig. 2.3) [41], which uses a symmetric encryption algorithm with a secret key and an asymmetric algorithm with an open key to transmit encrypted messages with an electronic signature.

In the patent [70] there's a suggested structure of the real symmetric-asymmetric system with an open key which satisfies the requirements (2.1) and (2.2). Encryption mode of an electronic signature is shown in Fig. 2.4

For this system capacity of a key set in the encryption mode is equal to $\#\{K \times k_{pr2}\}$ in comparison with the capacity of the set $\#\{K\}$ of the system shown in Fig. 2.3, and the capacity of a key set in the mode of an electronic signature $\#\{K \times k_{pr1}\}$ in comparison with $\#\{k_{pr1}\}$ of the system shown in Fig. 2.3.

Let's evaluate speed characteristics of the system presented in Fig. 2.4, in encryption mode. The total time of processing an input text of the length L_0 at the encryption speed V_{MV2} and encryption speed of a system with an open key V_{pb} equals:

$$T_{\Sigma} = \frac{L_0}{V_{MV2}} + \frac{L_C}{V_{pb}},$$

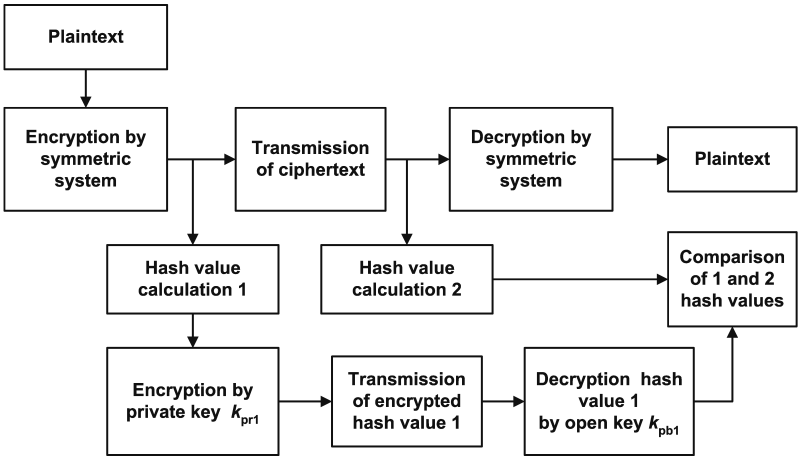

Fig. 2.3: *Hybrid scheme of encryption system and digital signature*

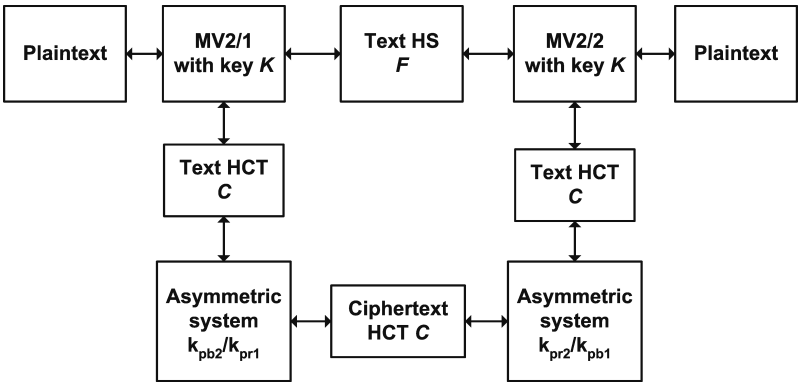

Fig. 2.4: *Real symmetric-asymmetric system with an open key in encryption mode*

where L_C is a length of a data-pump.

The total speed of processing an input text

$$V_{\sum} = \frac{L_0}{T_{\sum}} = \frac{V_{MV2}}{1 + \frac{L_C}{L_0} \cdot \frac{V_{MV2}}{V_{pb}}}. \tag{2.3}$$

At $\frac{V_{MV2}}{V_{pb}} \approx 10^3$ and $\frac{L_C}{L_0} \ll 1$ the speed $V_{\sum}$ is high enough in comparison with the encryption speed of an encryption system with the open key V_{pb}, at that the length of the data-pump can be considered constant, then the bigger the length of an input text, the closer the speed to that one of the symmetric MV2 system.

2.4 Three-channel symmetric-asymmetric MV3 system and a system with an open key

There are no analogs of such a system. It allows encrypting information by two keys in a single round: by a key of a symmetric system and that one of an asymmetric system of the user; it also allows signing messages by a secret key of a sender's asymmetric system. The structure of this system for the first variant of building MV3 (see 2.2) using the RSA system as an example is shown in Fig. 2.5.

The following transformations characterize performance of this system:

$$C = E_1(M, K);$$
$$F_m = E_2(M, K);$$
$$\{F_i\} = E_3(M, K);$$

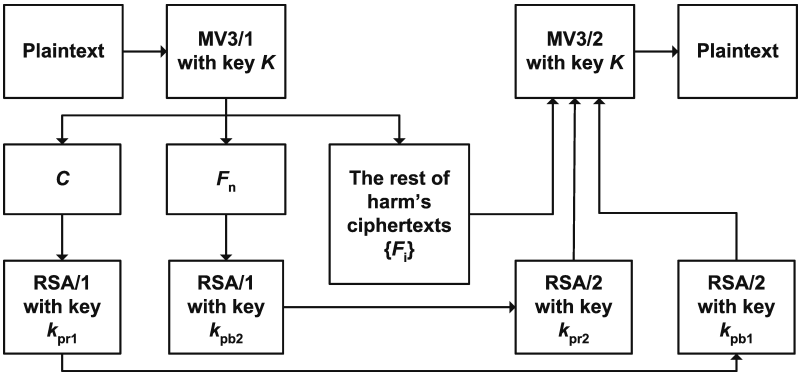

Fig. 2.5: *Three-channel symmetric-asymmetric system MV3-RSA*

$$C^{(1)} = E_4(C, k_{pr1});$$
$$F_m^{(1)} = E_5(F_m, k_{pb2});$$

$$C = E_6(C^{(1)}, k_{pb1});$$
$$F_m = E_7(F_m^{(1)}, k_{pr1});$$

$$M = E_{123}^{-1}(C, F_m, \{F_i\}, K).$$

Let's evaluate speed characteristics of such a system.

In compliance with (2.3) at paralleling the process of a digital signature and message encryption

$$V_\Sigma = \frac{V_{MV2}}{1 + \dfrac{L_C}{L_0} \cdot \dfrac{V_{MV2}}{V_{pb}}}.$$

If only one RSA system is used sequentially the speed will decrease by two times

$$V_\Sigma = \frac{V_{MV2}}{2(1 + \dfrac{L_C}{L_0} \cdot \dfrac{V_{MV2}}{V_{pb}})}.$$

The two-channel MV2-RSA system can solve the same problems if you use the same RSA system for core signing and encryption two times.

2.5 Combined cipher: MV2 and a stream cipher

A classical stream encryption system is presented in Fig. 2.6 where K is a generation key of the gamma G which correlates with the plaintext M in the transformation E:

$$Y = E(M, G);$$
$$M = E^{-1}(Y, G).$$

The gamma G is a pseudorandom sequence of the length equal to the length of a text being encrypted which is generated with the help of the short key K. Bitwise addition by 2 gamma bit with plaintext bits is usually used as the transformation E. A transmitted ciphertext Y contains all the information necessary for cryptanalysis.

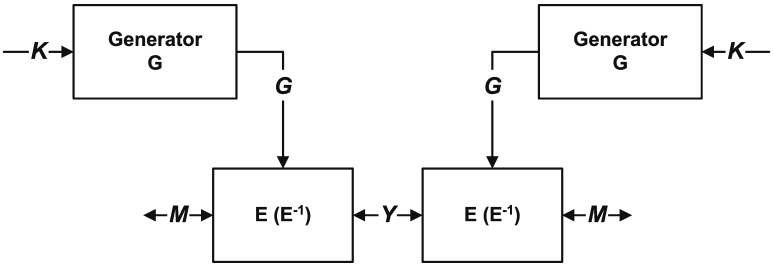

Fig. 2.6: *Classical stream encryption scheme*

We shall consider this scheme for harmed texts with application of the MV2 algorithm MV2 (Fig. 2.7).

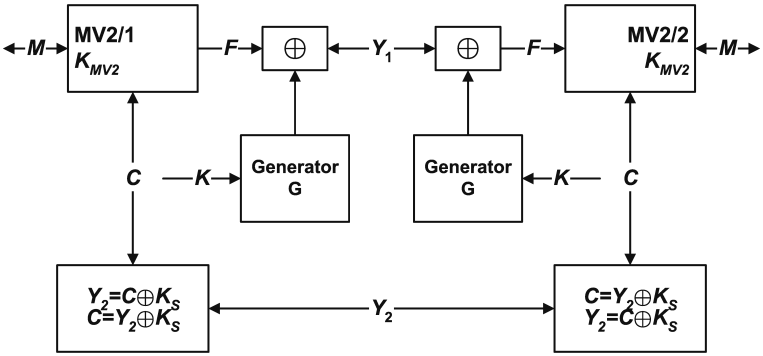

Fig. 2.7: *A two-channel scheme of a stream cipher with the application of the MV2 algorithm*

The following equations are true for the scheme in Fig. 2.7 :

– for encryption mode:

$$
\begin{aligned}
&F = E_1(M, K_{MV2});\\
&C = E_2(M, K_{MV2});\\
&G = G(K);\\
&Y_1 = F \oplus G;\\
&Y_2 = C \oplus K_S;
\end{aligned}
$$

– for decryption mode:

$$
\begin{aligned}
&C = Y_2 \oplus K_S;\\
&G = G(K);\\
&F = Y_1 \oplus G;\\
&M = E_{12}^{-1}(C, F, K_{MV2});
\end{aligned}
$$

The peculiarity of this system is that a system's ciphertext is unavailable for a cryptanalyst, if Y_2 is a conscious text, that can be easily reached with the help of the key K_S.

2.6 Concealed channel of information transmission. Information transmission with the help of MV2 algorithm keys.

Here we shall not consider well-known methods of data concealment [94, 95, 96] in the electronic signature as presence of a signature under harmless open messages arose suspicious thoughts in cryptanalysts' heads. We shall consider this problem from the position of capacities of multi-channel cryptography.

Assume information M_s should be transmitted via a secret channel. As we already mentioned before in 1.4, the information M_s must be presented in the form of a sensible text in an input language, not in the form of a ciphertext. To solve this problem one can use multi-channel cryptographic transformations. We shall consider this problem for a two-channel MV2 transformation based variant.

Let's make a system of equations:

$$\begin{cases} F = E_1(M, K_{MV2}); \\ C = E_2(M, K_{MV2}); \\ M = E_{12}^{-1}(F, C, K_{MV2}). \end{cases}$$

M_s, M are known. It's necessary to find such M_0, K_{MV2}, that M_s would have place under the condition that M_0 is an ideal text-container. As the keys K_{MV2} are secret components, they can be used for concealed transmission of conscious information: $K_{MV2} = M_s \oplus M_0$, from where knowing the open conscious text M_0, we can obtain $M_s = M_0 \oplus K_{MV2}$.

It's obvious that it's necessary to perform the inequality

$$L(K_{MV2}) \geq L(M_0) \geq L(M_s),$$

where $L(K_{MV2})$, $L(M_0)$ and $L(M_s)$ are correspondingly the lengths of the MV2 key, of the ideal text-container and of the concealed message.

We have this possibility as the maximal length of the keys K_{MV2} is about 50 000 bits. The text M is a camouflage for the transmission of the secret K_{MV2}.

Here we have a direct evidence of a crypto-steganographic system with very large key field which doesn't have transmitted ciphertexts.

The considered system allows playing a game under the conditional name of "Espionage passion". In contrast to the game "Prison correspondence under warder's control" described in [94], where a ciphertext of an electronic signature was placed under harmless messages (this should have put the warder Walter in his guard), there's nothing to pick on here.

Here are the game's rules. A resident of an espionage network is in legal public and secret commercial correspondence which doesn't contain any scandalous information. Moreover, due to his "occupation" he should correspond with a large enough number of business clients. If it's necessary he can always open this correspondence and show all public texts to corresponding bodies. There's nothing blameworthy in messages being transmitted. Nevertheless he can transmit concealed information using the following algorithm:

1. For usual business clients he is in public and secret correspondence. For secret correspondence he creates a public message M and encrypts it by the MV2 algorithm with the individual keys K_{MV2}. But for his secret consumer he chooses the key $K^s_{MV2} = M_s \oplus$

M_0, where M_s is an off-stage open text, and M_0 is a camouflage open text. The things to be transmitted are the ciphertext C, the harm F and a camouflage open text. It's a legal correspondence.

2. A secret consumer easily finds the off-stage message $M_s = K^s_{MV2} \oplus M_0$ as he is a legal member of the correspondence. The large legal message M is a camouflage which distracts attention from little semantic open containers.

2.7 Misleading with the help of the MV2 algorithm

The MV2 algorithm allows forming a ciphertext of a dummy pseudosecret message; knowing this message one can read a real secret message or directly transmit a conscious message in the form of a key.

Let M be a plaintext, and M' be a "pseudosecret" text being a camouflage for the plaintext M. At that, both texts have the same length. We create a message $K = M \oplus M'$. Then let the user Alice encrypt the text M' by the MV2 algorithm and find $C = E_1(M', K_{MV2})$, $F = E_2(M', K_{MV2})$ and $C' = C \oplus K'$. Alice sends Bob the short keys K_{MV2} and K' via a secret channel, and the message K which is a ciphertext of the message M, and ciphertexts C' and F - via an open channel.

Bob finds $C = C' \oplus K'$ and $M' = E^{-1}_{12}(C, F, K_{MV2})$, and then $M = M' \oplus K$. Actually it is the message K that contains information about the text M, but this information is protected by the MV2 algorithm with the keys K_{MV2}. Pay attention that the ciphertexts C' and F don't contain any information about the text M. If transmission of the

message K and ciphertexts C' and F is carried out via different channels, the task of breaking such a system is rather problematic without interception over the both channels.

This encryption system can be interpreted in a bit different way. A conscious message M' is transmitted instead of a meaningless message K, and the text K is encrypted by the MV2 algorithm:

Alice: transmits the short keys K_{MV2} and K' via a secret channel, and the conscious text M' and the ciphertexts C' and F - via an open channel.

Bob finds $C = C' \oplus K'$ and $M' = E_{12}^{-1}(C, F, K_{MV2})$, and then restores the plaintext $M = M' \oplus K$.

2.8 Multi-channel quantum cryptography

Here we shall consider possibilities of applying multi-channel cryptographic systems in quantum cryptography built on fiber-optic communication lines (FOCL). In combination with the principles of quantum mechanics and ideas of multi-channel cryptography we can synthesize *absolutely secure cryptographic systems*, protection of which is unavailable both for the mathematical theory of building cryptographic systems and for technical methods of FOCL protection.

It's appropriate to state here the main principles of quantum cryptography which is mainly used to transmit key information.

In 1984 Ch. Bennet (IBM, USA) and G. Brassard (The Montreal University, Canada) suggested using a photon stream for a fundamentally protected information channel in cryptography of relatively small flow volumes. It's obvious

that first of all it referred to transmission of small size key information. They suggested to use their idea to build an absolutely secure channel of distributing cryptographic keys and built such a system named BB84. This system allows transmitting protected information by using polarized photons as a carrier.

A photon is an elementary neutral particle having no mass or electric charge. A photon is an electromagnetic field quantum appearing during interaction of charged particles. Usual light is a photon flux having a chaotic vector location of electric $\vec{E}$ and magnetic $\vec{H}$ fields (at that the orientation plane of intensity vectors of electric and magnetic fields is perpendicular to the direction of the light flux). As the vectors $\vec{E}$ and $\vec{H}$ of an electromagnetic wave are mutually perpendicular, then one of them is chosen, usually the vector $\vec{E}$, to describe behavior of these light components. A physical characteristic describing behavior of the vector $\vec{E}$ in the plane perpendicular to pass of light is called *polarization*. Light is called *polarized* if the vector oscillates with a constant in time phase difference. At that, in the space perpendicular to pass of light its end can circumscribe an ellipse, or a circle of left- or right-side rotation, or it can pulse along the straight line when the ellipse is generated into a straight line segment (Fig. 2.8).

The main idea of using quantum radiation in cryptography consists in applying Gainsburg principle of uncertainty to elementary particles; according to this principle any outside attempt of taking a measurement of polarization, for instance, leads to change in this parameter, and consequently, can be detected, if initial state of this parameter is known. The impulse of horizontally polarized photons goes through horizontal polarized filter. If you rotate the filter, the flux of cutoff photons will decrease till not a single photon from the

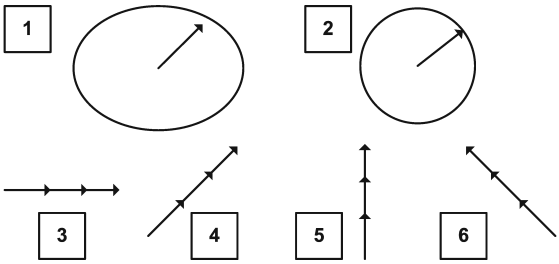

Fig. 2.8: *Types of the end vector $\vec{E}$ hodograph in the plane perpendicular to the direction of polarized light spreading. 1-elliptical polarization; 2 – round polarization; 3 – linear 0°; 4 – liner 45°; 5 – linear 90°; 6 – linear 135°*

horizontally polarized impulse will be able to run through the filter at the 90-degree turn. At the 45-degree turn it will let a horizontally polarized photon through with a probability of 50 %. Thus, it's possible to measure light polarization only if its' known in advance in what system it had been polarized. If you know that the light is polarized either vertically or horizontally, then, passing it through the horizontal filter we shall know by the result whether it was a 0- or 90-degree polarization. If it was a diagonal polarization, but we put a horizontal filter, it's impossible to define by results whether it was a 45- or 135-degree light polarization. Therefore it's impossible to eavesdrop on the channel formed by a stream of photon pulses, because a missed filter breaks the channel.

We shall explain it with an example. Let Alice send Bob a photon flux having a random, but fixed linear polarization (0° type or 90° type) (type of polarization 1) or a diagonal polarization (45° type or135° type) (type of polarization 2) (Table. 2.1).

The first line of the Table 2.1 interprets types and kinds of photon flux polarization (linear polarization 90° or diagonal polarization 135° is interpreted as 1, linear polarization

Tabl. 2.1:

1	1	0	1	0	0	0	1
2	90°	0°	135°	0°	45°	0°	90°
3	90°	90°	45°	135°	135°	90°	0°
4	+	+	-	-	-	+	+
5	да	да	да	нет	да	да	да
6	1	0	1	?	0	0	1

Continuation

1	1	1	1	1	0	1	0
2	90°	135°	90°	90°	45°	135°	0°
3	0°	45°	135°	135°	0°	45°	90°
4	+	-	-	-	+	-	+
5	yes	yes	no	no	no	yes	yes
6	1	1	?	?	?	1	0

1 – information being transmitted by Alice;

2 – polarization of photons being sent by Alice;

3 – a polarization measurement type which Bob chooses;

4 – a measurement type: (+)) – rectilinear: 0°, 90° or (-) – diagonal: 45°,135°);

5 – Bob informs Alice about the kind (but not about the type!), and Alice either confirms or doesn't confirm correctness of Bob's choice;

6 – Bob's interpretation of obtained measurements.

0° or diagonal polarization 45° – as 0). Bob knows the same interpretation. Bob's logical interpretation concerning the obtained measurements consists in the following. If he correctly guesses polarization kind and type, the photon flux will be intensive (a measurement filter is chosen correctly). If he guesses the kind, but not the type of polarization the photon flux will be very weak or will be absent at all due to 90° divergence between flux polarization and a filter. This gives him the opportunity to change his assumption for the correct

one. If he didn't guess the kind of polarization (a photon flux will be of average intensity due to 45 °divergence between flux polarization and a filter) he can't define polarization type either as the measurement result with the probability of 0,5 can belong to the both polarization types.

Bob interprets his measurement results in the following way:

1 – he guessed a polarization type(90°), at that the photon flux was intensive. It means he also guessed the type of polarization 90°. It corresponds to 1;

0 – he guessed a polarization kind (90°), but the photon flux was very weak. It means the polarization was linear, but a different one – 0°. It corresponds to 0;

1 – he guessed a polarization kind (45°), but the photon flux was very weak. It means the polarization was diagonal, but a different one – 135°. It corresponds to 1;

0 – he didn't guess a kind of polarization (135°), as the photon flux was half less intensive than during a univocal guessing. It means the flux had the polarization 0° or 90°. A polarization type is not defined (?);

and so on.

Thus, we obtained a reliable common secret in the form of 1010011110, which can be used as key information.

Light-emitting diodes, lasers or microlasers are used as sources of polarized photons emission; fiber-optic communication lines are used as physical transmission channels.

Along with its main advantage (a high protection degree of a transmission channel) the considered method also has evident disadvantages: little traffic capacity in comparison with electronic channels, and noise. The cost of such systems

is also high enough ($50 000 – 100 000). Today these disadvantages are being successfully overcome, and such systems are being implemented in the modern market mainly for government establishments.

In the framework of this book we shall be interested not only in possibility of quantum-optic communication channels to transmit key information of small volumes, but also in transmission of data-pumps of small capacities and building hybrid systems of multi-channel cryptography.

The peculiarity of generation and transmission of key information consists in its randomness, and, therefore, a common secret can differ from the first-generated one in coordination. Data-pumps are deterministic in a specific transmission and can't be changed. Moreover, changing a core can lead to complete loss of information, therefore, secrecy and reliability of the core transmission channel must be of the highest level.

A structure of a classical cryptographic system using quantum-optic communication channel for transmitting key information is shown in Fig. 2.9. In this system keys are transmitted via a quantum-optic channel in the beginning of a session, and then information encrypted by these keys is transmitted via a traditional open communication channel. It can be intercepted by an adversary in the ordinary way and then subdued to a cryptanalysis. There is only one fact that may calm us down: session keys transmitted via a quantum-optic communication channel are unavailable for it.

A two-channel cryptographic structure using a quantum-optic communication channel for not only transmitting key information, but also ciphertexts of a harmed text (data-pump) and/or a ciphertext of a short harm is shown in Fig. 2.10. This structure works in the following way. In the beginning, as in the case presented in Fig. 2.9 a quantum-

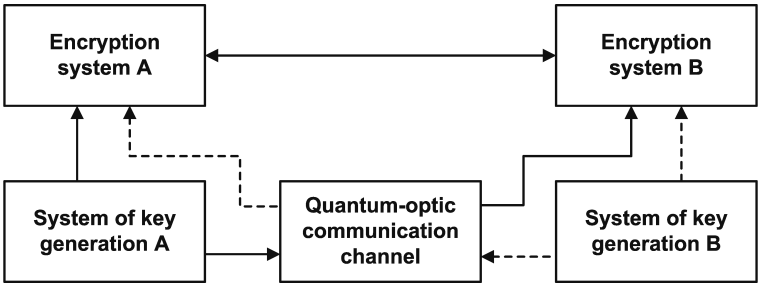

Fig. 2.9: *Structure of a cryptographic system using quantum-optic communication channel for transmitting key information*

optic communication channel is used to transmit session keys. Encrypted harms' data is transmitted via a traditional speed open communication channel, and a short data-pump – via a quantum-optic communication channel. Here principally unexpected troubles are waiting for us. Assume a core is the information of the Table 2.1:

$$10100011111010.$$

Alice transmitted exactly this information, and Bob measured it

$$101?00111???10.$$

It's evident that at least a repeated transmission is needed. Alice must repeat it till Bob interprets absolutely correctly measurement results without communicating with Alice.

There's an interesting variant of a logical conclusion for the structure in Fig. 2.10.

Assume Alice and Bob came to the conclusion that a quantum core transmission channel had not been browsed by Mallory (a curious unpleasant subject). Then they shouldn't worry irrespective of Mallory's attempts on a harm transmission channel: Mallory knows the ciphertexts of the harms, but he does not know the ciphertext of the harmed text. The system has a huge additional key the length of which equals the length

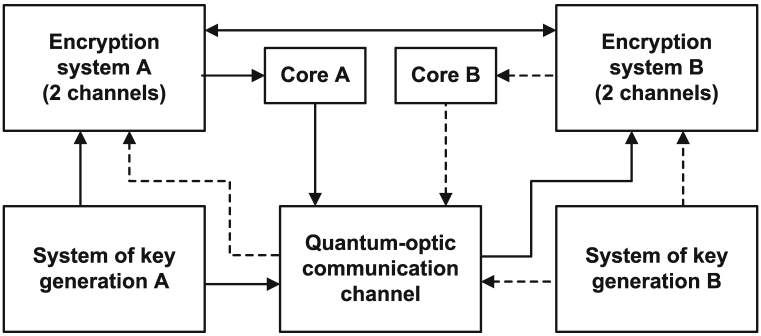

Fig. 2.10: *A two-channel cryptographic structure using a quantum-optic communication channel*

of a harmed text, and a very large unicity distance. But if Alice and Bob came to the conclusion that Mallory browses the core channel, they can take measures to eliminate the leak via this channel. They cannot permit browsing this channel, because the core data will be corrupted! It's important to note that in the both cases Alice and Bob can conclude for sure whether this core channel is being browsed or not.

2.9 Summary

Multichanneling gives a new alternative to cryptography that is a higher security and natural possibility of a reunion with steganography. Actually both cryptography and steganography are called in to provide keeping some secret and to protect communication parties from both an external adversary and dishonest actions of a partner. This bridge of reunion rests upon two piers: steganographic concealment of data of small sizes, and a cryptographic method of obtaining harmed texts of small sizes. Combination of these possibilities

allows synthesizing cryptosystems with principally new features: visible and invisible harmed ciphertexts. In some cases such a possibility will place a practically insuperable problem in front of the cryptanalysis. It will happen because subtle deep connections between an open text and a ciphertext will be broken by a visible corrupted (harmed) ciphertext which has a meaning different from the meaning of an open plaintext or doesn't have it at all. Even if an observer can see all harmed texts he has a key field of harming arisen in front of him which sharply increases the unicity distance.

Multi-channel cryptography gives the possibility of interconnecting cryptographic systems by naturally uniting various cryptosystems under one concept: symmetric and asymmetric which supplement each other and enrich the total system with their features, and, in the first place, with security, and new availabilities in many practical applications.

Chapter 3
Universal mechanism of harming

3.1 Substitution transformation for obtaining harmed texts

3.1.1 Mappings with variable length of an image

In majority of modern ciphers cryptographic transformations that transform a block of a plaintext M of a length n bits into a block of a ciphertext of a length m bits:

$$f : \{0,1\}^n \to \{0,1\}^m.$$

We shall call such transformations as mappings with a fixed length of an image.

We shall further use the term *inputs* for values from a mapping definitional domain and the term *outputs* for mapping values.

To construct harming transformations, we shall consider mappings of the following kind:

$$f : \{0,1\}^n \to \bigcup_{i=r}^{m} \{0,1\}^i.$$

We shall call such mappings as *mappings with variable length of an image*, or mappings *with a variable length output*. To make it simple, we shall introduce symbols:

$$\mathcal{U}_{rm} = \bigcup_{i=r}^{m} \{0,1\}^i. \tag{3.1}$$

Thus, the set $\{0,1\}^i$ contains 2^i elements, then the set $\mathcal{U}_{rm}$ contains

$$\#\mathcal{U}_{rm} = \sum_{i=r}^{m} 2^i = 2^{m+1} - 2^r$$

various binary strings. Note that if $m < n$, then mappings with arbitrary length can not be injective, because a set of inputs contains 2^n different elements, and a set of outputs contains $2^{m+1} - 2^r < 2^n$.

Mappings with variable length images are used by some data compressing techniques, for instance. A Huffman code can be an example of such a mapping. [53].

We shall indicate a number of ranks in the element $x \in \mathcal{U}_{rm}$ as $|x|$. We shall also call the value $|x|$ as the length of the element $x \in \mathcal{U}_{rm}$.

We shall call two binary strings from the set $\mathcal{U}_{rm}$ equal if their length and corresponding ranks are equal.

Metric at the set of binary string of various length

As we know, a metric at the set $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, possessing the following features:

1. for any $x, y \in \mathcal{X}$ $d(x,x) = 0$ if and only if $x = y$;
2. for any $x, y \in \mathcal{X}$ the following is performed: $d(x,y) = d(y,x)$ (symmetry);

3. for any $x, y, z \in \mathfrak{X}$ the triangle inequality is performed:

$$d(x, y) \leq d(x, z) + d(z, y).$$

Usually a metric known as Hemming distance is used at research of mappings with a fixed image length at a set $\{0,1\}^n$:

Let x and y – n be binary strings, where $\mathrm{x}_i, \mathrm{y}_i \in \{0,1\}$ define the corresponding ith binary bit.

Definition 3.1 *Hemming distance between the binary strings x and $y \in \{0,1\}^n$ is a number:*

$$w(x, y) = \sum_{i=1}^{n} \mathrm{x}_i \oplus \mathrm{y}_i. \tag{3.2}$$

At a set of binary strings of variable length one can introduce a metric similar to the Hemming distance.

Definition 3.2 *The distance between the binary strings x and $y \in \mathfrak{U}_{rm}$ is a number*

$$h(x, y) = w(\overline{x}^k, \overline{y}^k) + \big||x| - |y|\big|, \tag{3.3}$$

where $k = \min\{|x|, |y|\}$, and $\overline{x}^k$ and $\overline{y}^k$ denote k-bit binary strings, the corresponding bits of which coincide with the corresponding bits of the binary strings x and y.

Proof of correctness of introducing a metric. We shall prove that the function h introduced in the definition 3.2 is a metric at the set $\mathfrak{U}_{rm}$.

It's obvious, that the first and the second points of the metric definition are performed for the function h, therefore to prove h is a metric at a set of binary strings of various length,

it's necessary to prove performing of triangle inequality.

We shall take arbitrary binary strings x, y, z. Let $k = \min\{|x|, |y|, |z|\}$.

Denote through $\overline{x}^k$, $\overline{y}^k$ and $\overline{z}^k$ – k-bit binary strings, the corresponding bits of which coincide with the corresponding bits of the binary strings x, and y and z.

The Hemming distance w is a metric, therefore, the following is performed:

$$w(\overline{x}^k, \overline{y}^k) \leq w(\overline{x}^k, \overline{z}^k) + w(\overline{z}^k, \overline{y}^k).$$

Lengths of the binary strings – $|x|$, $|y|$ and $|z|$ are integer nonnegative numbers, therefore, the triangle inequality is true for them:

$$\big||x| - |y|\big| \leq \big||x| - |z|\big| + \big||z| - |y|\big|.$$

As

$$h(x, z) = w(\overline{x}^k, \overline{z}^k) + \big||x| - |z|\big|,$$

$$h(y, z) = w(\overline{z}^k, \overline{y}^k) + \big||z| - |y|\big|.$$

then,

$$h(x, y) \leq h(x, z) + h(z, y).\square.$$

We shall need the metric introduced in the definition 3.2 during examination of algorithms of harming.

3.1.2 Definition of the MV2-transformation

To construct harming transformations we shall consider substitution transformations that replace binary strings of the length n bits by strings of variable length that are *smaller*, than n. As the number of elements in the definitional domain of such transformation is bigger than that in the range of

value, then such transformations are not injective mappings. But, to restore a plain text from a transformed one the transformation of a plaintext needs to be an invertible mapping.

We shall set an integral positive value r and n, such that $0 < r < n$. Consider the mappings

$$c : \{0,1\}^n \to \mathcal{U}_{r\ n-1},$$

having the following properties:

1) For any element $y \in \mathcal{U}_{r\ n-1}$ there's at least one element $x \in \{0,1\}^n$, that is an original at the mapping c i.e. $c(x) = y$;

2) for any element $y \in \mathcal{U}_{r+1\ n-1}$ the mapping c has the only original, and various elements have various originals;

3) for any element $y \in \{0,1\}^r$ the mapping c has only two originals.

For every mapping c we shall define an integer function

$$f : \{0,1\}^n \to \{1, \ldots, n-r+1\}$$

connected with it in the following way:

1) $f(x) = n - |c(x)|$, if $|c(x)| > r$;

2) for any $x_1 \neq x_2 \in \{0,1\}^n$, having identical images $c(x_1) = c(x_2) \in \{0,1\}^r$, the function value f is either $n-r$ or $n-r+1$.

It's evident, that if we fix the mapping c, then one can construct 2^{2^r} various functions f. At that, every particular pair (c, f) is an injective mapping of the kind:

$$\begin{aligned} T : \{0,1\}^n \ \to \bigcup_{i=1}^{n-r-1} \Big\{\{0,1\}^{n-i} \times \{i\}\Big\} \bigcup \\ \bigcup \Big\{\{0,1\}^r \times \{n-r, n-r+1\}\Big\}, \end{aligned} \tag{3.4}$$

that transforms a binary string of a fixed length into a pair: (a variable length binary string, a number). We shall denote a set of such mappings as $\mathcal{F}_n^r$. We shall further call these mappings as **MV2- mappings** (transformations).

We shall call the image $c(x) \in \mathcal{U}_{r\ n-1}$ as **a remainder**, and $f(x) \in \{1, \ldots, n-r+1\}$ as **a flag**.

Output values $f(x)$ of the transformation $T = (c, f)$ can be encoded by a binary code (BC) as shown in the Table 3.1. In this table 0^i indicates a bit string of i zeros.

Tabl. 3.1: *Value encoding f*

$f(x)$	1	2	3	...	$n-r$	$n-r+1$
ДК	1	01	$0^2 1$	...	$0^{n-r-1}1$	0^{n-r}

At uniform input distribution such a code coincides with the Huffman code [53], which, as you know, is an optimal one. Further, if it's not specially specified, under the input of the function f we shall understand its representation with the help of a binary code.

It's obvious that any such mapping from $\mathcal{F}_n^r$ can be set with the help of the table, in the left part of which there's a permutation $(s_1, \ldots, s_{2^n})$ of values from 1 to 2^n, that are presented in form of n-byte binary strings, and in the right part there are images consisting of a "remainder" and "flag" parts, as it is shown in the table 3.2. In this table 0^i and 1^i indicate binary strings of i zeros and ones correspondingly.

If in the Table 3.2 we fix the right columns, then some permutation will correspond to every MV2-transformation, and some MV2-transformation will correspond to every permutation. Thus, there's a bijection between sets of permutation $\{(s_1, \ldots, s_{2^n})\}$ and the set $\#\mathcal{F}_n^r$. Therefore:

$$\#\mathcal{F}_n^r = \#\Big\{ (c, f) \Big\} = 2^n!.$$

Tabl. 3.2: *Task of a substitution transformation*

a symbol	a remainder length	a remainder	a flag
s_1	r	0^r	$0^{n-r-1}1$
s_2	r	$0^{r-1}1$	$0^{n-r-1}1$
...	...	...	...
s_{2^r+1}	$r+1$	0^{r+1}	$0^{n-r-2}1$
...	...	...	...
$s_{2^{n-1}-2^r}$	$n-2$	1^{n-2}	01
$s_{2^{n-1}-2^r+1}$	$n-1$	0^{n-1}	1
...	...	...	...
$s_{2^n-2^r}$	$n-1$	1^{n-1}	1
$s_{2^n-2^r+1}$	r	0^r	0^{n-r}
...	...	...	...
s_{2^n}	r	1^r	0^{n-r}

In the table 3.3 there's an example of an MV2-transformation with the parameters $n = 4$ and $r = 2$.

Note, that any flag mapping f splits the set $\{0,1\}^n$ into $n-r+1$ noncrossing subsets $\mathfrak{X}_i$ such that $\forall x \in \mathfrak{X}_i : \ f(x) = i$ and in every set $\mathfrak{X}_i$, for $n-r \le i \le n-1$ exactly 2^{n-i} elements are comtained and the set $\mathfrak{X}_{n-r+1}$ contains 2^r elements.

Thus, there are as many various flag mappings as methods of a set splitting $\{0,1\}^n$ into $n-r+1$ noncrossing subsets $\mathfrak{X}_i$.

Members of the set $\mathfrak{X}_1$ can be chosen with the help of $\binom{2^n}{2^{n-1}}$ methods, then, members of the set $\mathfrak{X}_2$ out of the remained 2^{n-1} elements – with the help of $\binom{2^{n-1}}{2^{n-2}}$ methods and so on ... members of the set $\mathfrak{X}_{n-r}$ – with the help of $\binom{2^{r+1}}{2^r}$ methods and members of the set $\mathfrak{X}_{n-r+1}$ – with the help of one method only.

We shall indicate $\mathcal{F}$ as the set of all such mappings f.

Tabl. 3.3: *Example of an MV2-transformation for $n = 4$ and $r = 2$*

$\mathbf{x}$	0000	0001	0010	0011
$\mathbf{c}(\mathbf{x})$	111	110	10	01
$\mathbf{f}(\mathbf{x})$ (ДК)	1(1)	1(1)	2(01)	3(00)
$\mathbf{x}$	0100	0101	0110	0111
$\mathbf{c}(\mathbf{x})$	10	11	010	11
$\mathbf{f}(\mathbf{x})$ (ДК)	3(00)	2(01)	1(1)	3(00)
$\mathbf{x}$	1000	1001	1010	1011
$\mathbf{c}(\mathbf{x})$	00	001	100	00
$\mathbf{f}(\mathbf{x})$ (ДК)	2(01)	1(1)	1(1)	3(00)
$\mathbf{x}$	1100	1101	1110	1111
$\mathbf{c}(\mathbf{x})$	101	000	011	01
$\mathbf{f}(\mathbf{x})$ (ДК)	1(1)	1(1)	1(1)	2(01)

Then the set $\mathcal{F}$ contains

$$\#\mathcal{F} = \prod_{i=r}^{n-1} \binom{2^{i+1}}{2^i} = \frac{2^n!}{\prod\limits_{i=r}^{n-1} 2^i!} \tag{3.5}$$

different mappings f.

Similarly, any mapping of the remainder can be chosen by $\frac{2^n!}{2^{n-1}!} \cdot \frac{2^{n-1}!}{2^{n-2}!} \cdot \ldots \cdot \frac{2^{r+1}!}{2^r!} \cdot 2^r!$ methods. Thus, the number of different remainder mappings coincides with the number of various MV2-type mappings and is equal to $2^n!$.

Let $y \in \mathcal{U}_{r\ n-1}$ be a binary string of the string set of variable length. Let's define as $y^{(i)}$ a binary string obtained from y by the inversion of an i-th bit, $1 \le i \le |y|$. For example, $y = 01001010$, then $y^{(3)} = 01101010$.

The following lemma occurs.

Lemma 3.1 *Let $T = (c, f) \in \mathcal{F}_n^r$ be an arbitrary fixed MV2 transformation. Then for any $y \in \mathcal{U}_{r\ n-1}$ and for any $1 \le i \le |y|$ the following is carried out*

$$\#\Big\{x \in \{0,1\}^n : c(x) = y\Big\} = \#\Big\{x \in \{0,1\}^n : c(x) = y^{(i)}\Big\}$$

As in every set $\{0,1\}^k$ the number of one-bit and zero bit is the same, conclusion of the lemma evidently follows from the definition of a MV2- transformation.

The statement of the lemma 3.1 expresses a very important property of the MV2- transformation, that is the number of one- and zero-bit in a remainder set coincides.

3.1.3 Information and statistical estimations for an MV2-transformation

In 1.3 we defined a universal mechanism of harming. In this mechanism during harming, codes of text symbols are replaced by binary strings of various length. In the previous paragraph we built the transformation wich can be used to harm arbitrary texts.

The domain of an MV2-transformation is a set $\{0,1\}^n$, it can be considered as an alphabet of a family of plaintexts. Let a probability distribution for the letters in the alphabet is given. Then the input of the MV2-transformation is a random vector and the outputs are random strings. Besides, as outputs of the MV2-transformation are binary strings of variable length, the output lengths can also be considered as random variables.

Let X be a discrete random variable with possible values $x_i \in \{0,1\}^n$ which have probabilities p_i, $i = 1, 2, \dots 2^n$, and let $T = (c, f)$ be an MV2-transformation, where $c : \{0,1\}^n \to$

$\mathfrak{U}_{r\ n-1}$ is a remainder mapping with images of variable length, and $f : \{0,1\}^n \to \{1,2,\ldots,n-r+1\}$ is a flag mapping. The random pair $Y_{\mathrm{DT}}, Y_{\mathrm{D}}$, where $Y_{\mathrm{DT}} = c(X)$ and $Y_{\mathrm{D}} = f(X)$ is an image of the random element X. The random variables Y_{DT} and Y_{D} are random strings, and their lengths $|Y_{\mathrm{DT}}|$ and $|Y_{\mathrm{D}}|$ are random values.

For the joint entropy of random elements X, Y_{DT} and Y_{D} the following identities are satisfied [11]:

$$H(XY_{\mathrm{DT}}Y_{\mathrm{D}}) = H(X) + H(Y_{\mathrm{DT}}Y_{\mathrm{D}}|X),$$
$$H(XY_{\mathrm{DT}}Y_{\mathrm{D}}) = H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) + H(X|Y_{\mathrm{DT}}Y_{\mathrm{D}}).$$

The pair $(Y_{\mathrm{DT}}, Y_{\mathrm{D}}) = T(X)$, when a transformation $T \in \mathfrak{F}_n^r$ is fixed, therefore

$$H(Y_{\mathrm{DT}}Y_{\mathrm{D}}|X) = H(X|Y_{\mathrm{DT}}Y_{\mathrm{D}}) = 0.$$

Consequently, for the joint entropy Y_{DT} and Y_{D} the following is satisfied:

$$H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) = H(X). \tag{3.6}$$

On the other hand

$$H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) = H(Y_{\mathrm{DT}}) + H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) = H(Y_{\mathrm{D}}) + H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}).$$

From that we have the following, using (3.6):

$$H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}) = H(X) - H(Y_{\mathrm{D}}), \tag{3.7}$$

$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) = H(X) - H(Y_{\mathrm{DT}}). \tag{3.8}$$

Due to definition and (3.7) the mutual information between Y_{DT} and Y_{D} is

$$I(Y_{\mathrm{DT}}, Y_{\mathrm{D}}) = H(Y_{\mathrm{DT}}) + H(Y_{\mathrm{D}}) - H(X). \tag{3.9}$$

We shall consider informational dependencies between the input and outputs.

If T is fixed, the output Y_{DT} is completely determined by the input X, therefore $H(Y_{\mathrm{DT}}|X) = 0$.

Then the mutual information between X and Y_{DT} equals

$$I(X, Y_{\mathrm{DT}}) = H(Y_{\mathrm{DT}}). \tag{3.10}$$

For the joint entropy of X and Y_{DT} the following is carried out:

$$H(XY_{\mathrm{DT}}) = H(X) + H(Y_{\mathrm{DT}}|X) = H(Y_{\mathrm{DT}}) + H(X|Y_{\mathrm{DT}}).$$

From which we have

$$H(X) = H(Y_{\mathrm{DT}}) + H(X|Y_{\mathrm{DT}}). \tag{3.11}$$

Similarly, as Y_{D} is the part of the image X and the transformation T is fixed, then $H(Y_{\mathrm{D}}|X) = 0$, therefore, for the joint entropy and the mutual information between the random variables X and Y_{D} the following is carried out:

$$H(X) = H(Y_{\mathrm{D}}) + H(X|Y_{\mathrm{D}}), \tag{3.12}$$

$$I(X, Y_{\mathrm{D}}) = H(Y_{\mathrm{D}}). \tag{3.13}$$

If $T = (c, f)$ is the fixed MV2-transformation with the parameters r and n, then the flag mapping f split the domain into noncrossing subsets $\mathfrak{X}_1, \ldots \mathfrak{X}_{n-r+1} \subset \{0,1\}^n$, such, that for all $x \in \mathfrak{X}_i$ the following is carried out: $f(x) = i$.

Let the inputs x_i is numbered such that for $i = 1, 2, \ldots, 2^{n-1}$ the images $c(x_i) \in \{0,1\}^{n-1}$ and $f(x_i) = 1$; for $i = 2^{n-1} + 1, \ldots, 2^{n-1} + 2^{n-2}$ the images $c(x_i) \in \{0,1\}^{n-2}$, and $f(x_i) = 2$; $\ldots$, for $i = 2^n - 2^{r+2} + 1, \ldots, 2^n - 2^{r+1}$ images $c(x_i) \in \{0,1\}^{r+1}$ and $f(x_i) = n - r - 1$; for $i = 2^n - 2^{r+1} + 1, \ldots, 2^n - 2^r$

images $c(x_i) \in \{0,1\}^r$ and $f(x_i) = n - r$; and, finally, for $i = 2^n - 2^r + 1, \ldots, 2^n$ images $c(x_i) \in \{0,1\}^r$ and $f(x_i) = n - r + 1$.

Then for the remainder entropy $H(Y_{\mathrm{DT}})$ we have

$$H(Y_{\mathrm{DT}}) = -\sum_{i=1}^{2^n - 2^{r+1}} p_i \log p_i - \sum_{i=2^n-2^{r+1}+1}^{2^n-2^r} (p_i + p'_i) \log(p_i + p'_i),$$

where $p'_i = p_{i+2^r}$ for $i = 2^n - 2^{r+1} + 1, \ldots, 2^n - 2^r$.

We can see that the random elements X and Y_{DT} have different distributions in the general case.

Difference of input and output entropies is

$$H(X) - H(Y_{\mathrm{DT}}) = \sum_{i=2^n-2^{r+1}+1}^{2^n-2^r} \Big(p_i \log(1 + \frac{p'_i}{p_i}) + p'_i \log(1 + \frac{p_i}{p'_i}) \Big),$$

and can be estimated as

$$0 \le H(X) - H(Y_{\mathrm{DT}}) \le \sum_{i=2^n-2^{r+1}+1}^{2^n} p_i \log(1 + \frac{1}{p_i}).$$

Denote by P_r the probability event $f(X) = r$, then

$$P_r = P(c(X) \in \{0,1\}^r) = \sum_{i=2^n-2^{r+1}+1}^{2^n} p_i$$

and

$$\sum_{i=2^n-2^{r+1}+1}^{2^n} p_i \log(1 + \frac{1}{p_i}) = P_r \sum_{i=2^n-2^{r+1}+1}^{2^n} \frac{p_i}{P_r} \log(1 + \frac{1}{p_i}).$$

As the function $x \log(1 + \frac{1}{x})$ is convex, then, due to the Jensen inequality we have

$$\sum_{i=2^n-2^{r+1}+1}^{2^n} p_i \log(1 + \frac{1}{p_i}) \le \Big(\sum_{i=2^n-2^{r+1}+1}^{2^n} p_i \Big) \log(1 + \frac{P_r}{\sum p_i}).$$

Thus, the following estimation is true:

$$0 \le H(X) - H(Y_{\mathrm{DT}}) \le P_r. \tag{3.14}$$

At that, the equality can be reached only if all inputs x_i with r-bit images have the same probability $p = p_i$.

For equiprobable inputs all probabilities $p_i = 1/2^n$, $i = 1, \ldots, 2^n$, therefore, we have the following from (3.14):

$$H(X) - H(Y_{\mathrm{DT}}) = n - H(Y_{\mathrm{DT}}) = 2^{r+1-n}. \tag{3.15}$$

Similarly, for flag entropy $H(Y_{\mathrm{D}})$ we have:

$$\begin{aligned} H(Y_{\mathrm{D}}) = {} & \Big(\sum_{i=1}^{2^{n-1}} p_i\Big) \log \frac{1}{\sum\limits_{i=1}^{2^{n-1}} p_i} + \ldots + \\ & + \Big(\sum_{i=2^n-2^{r+1}+1}^{2^n-2^r} p_i\Big) \log \frac{1}{\sum\limits_{i=2^n-2^{r+1}+1}^{2^n-2^r} p_i} + \\ & + \Big(\sum_{i=2^n-2^r+1}^{2^n} p_i\Big) \log \frac{1}{\sum\limits_{i=2^n-2^r+1}^{2^n} p_i}. \end{aligned}$$

Let

$$P_k = \sum_{\{x: f(x)=k\}} P(X = x) \tag{3.16}$$

be the probability that a flag image will possess the value $k = 1, \ldots, n - r + 1$. Then

$$\begin{aligned} H(X) - H(Y_{\mathrm{D}}) = {} & \sum_{i=1}^{2^{n-1}} p_i \log \tfrac{P_1}{p_i} + \ldots + \\ & + \sum_{i=2^n-2^{r+1}+1}^{2^n-2^r} p_i \log \tfrac{P_{n-r}}{p_i} + \\ & + \sum_{i=2^n-2^r+1}^{2^n} p_i \log \tfrac{P_{n-r+1}}{p_i}. \end{aligned}$$

From here, using again Jensen inequality we get:

$$H(X) - H(Y_{\mathrm{D}}) \leq \sum_{k=1}^{n-r}(n-k)\cdot P_k + r\cdot P_{n-r+1}.$$

taking into account that $\sum\limits_{k=1}^{n-r+1} P_k = 1$, we have:

$$H(X) - H(Y_{\mathrm{D}}) \leq n + P_{n-r+k} - \sum_{k=1}^{n-r+1} k\cdot P_k, \qquad (3.17)$$

In the inequality (3.17) equality is reached only if for every $k = 1,\ldots,n-r+1$ probabilities of all the preimages $x \in \mathcal{X}_k$ are equal to $P(X = x : f(x) = k) = \frac{P_k}{2^{|f(x)|}}$ for all $x \in \mathcal{X}_k$, where $|f(x)|$ is a number of bits in the representation of the value $f(x)$ in form of a bit string (see table 3.1).

Whatever probability distribution of the random element X, is, for the entropy of the random elements Y_{DT} and Y_{D} the following inequalities are true:

$$H(Y_{\mathrm{DT}}) \leq \log(2^n - 2^r) \qquad (3.18)$$

$$H(Y_{\mathrm{D}}) \leq \log(n - r + 1) \qquad (3.19)$$

Information estimations for outputs at uniform input distribution

We shall consider an important special case when the inputs $T = (c, f)$ – that are MV2- transformations, are uniformly distributed, i.e. probabilities of any symbol $x \in \{0,1\}^n$ coincide and equal $\dfrac{1}{2^n}$.

In this case remainder probability $p_y^{(c)} = P(Y_{\mathrm{DT}} = y)$ and flag probability $p_k^{(f)} = P(Y_{\mathrm{D}} = k)$ will equal:

$$p_y^{(c)} = \begin{cases} 2^{-n}, & \text{if } y \in \bigcup\limits_{i=r+1}^{n-1}\{0,1\}^i \\ 2^{1-n}, & \text{if } y \in \{0,1\}^r \end{cases}; \qquad (3.20)$$

$$p_k^{(f)} = \begin{cases} 2^{-k}, & \text{if } 1 \le k \le n-r \\ 2^{r-n}, & \text{if } k = n-r+1 \end{cases}. \qquad (3.21)$$

If $T = (c, f)$ is an arbitrary fixed MV2- transformation and the random element $X \in \{0,1\}^n$ has a uniform distribution, then the entropy $H(X) = n$. Therefore, the following equality follows form the expressions (3.10), (3.14), (3.20)

$$I(X; Y_{\mathrm{DT}}) = H(Y_{\mathrm{DT}}) = n - 2^{r+1-n}. \qquad (3.22)$$

Similarly, from the expressions (3.13), (3.17), (3.21) and identity

$$\sum_{k=1}^{m} k \cdot 2^k = (m-1)2^{m+1} + 2$$

we have

$$I(X; Y_{\mathrm{D}}) = H(Y_{\mathrm{D}}) = 2 - 2^{r+1-n}. \qquad (3.23)$$

from (3.7), (3.12) and (3.23) we have:

$$H(X|Y_{\mathrm{D}}) = H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}) = n - 2 + 2^{r+1-n}. \qquad (3.24)$$

And from (3.11) and (3.22) we shall get:

$$H(X|Y_{\mathrm{DT}}) = 2^{r+1-n}, \qquad (3.25)$$

Accordingly, it follows from (3.9), (3.22) and (3.23), that:

$$I(Y_{\mathrm{DT}}; Y_{\mathrm{D}}) = 2 - 2^{r+2-n}. \qquad (3.26)$$

Estimations of output lengths for an MV2-transformation

Let $T = (c, f)$ be an MV2- transformation. If we consider an input as a random element $X \in \{0,1\}^n$, then the remainder output $Y_{\mathrm{DT}} = c(X)$ is a random element which possesses values from a set of variable length binary strings.

Correspondingly the output length of the remainder $|Y_{\mathrm{DT}}|$ is a numeric random value. As it was mentioned above a flag output can be encoded by a binary code (see table 3.1), in this case the output $Y_{\mathrm{D}} = f(X)$ is a random element possessing values from a set of variable length strings, and the output length of the flags $|Y_{\mathrm{D}}|$ is also a numeric random value.

Using the designation (3.16), in a general case, expectations of output lengths of a remainder and flags can be represented by the expression:

$$\mathbf{E}(|Y_{\mathrm{DT}}|) = r \cdot P_{n-r+1} + \sum_{k=1}^{n-r}(n-k) \cdot P_k; \tag{3.27}$$

$$\mathbf{E}(|Y_{\mathrm{D}}|) = (n-r) \cdot P_{n-r+1} + \sum_{k=1}^{n-r} k \cdot P_k. \tag{3.28}$$

In case of uniform distribution of inputs for expectations and random value dispersions $|Y_{\mathrm{DT}}|$ and $|Y_{\mathrm{D}}|$ the following expression is true:

Claim 3.1 *If* $T = (c, f) \in \mathcal{F}_n^r$, *then at an uniform input distribution for expectations* $\mathbf{E}(|Y_{\mathrm{DT}}|)$, $\mathbf{E}(|Y_{\mathrm{D}}|)$ *and dispersions* $D(|Y_{\mathrm{DT}}|)$, $D(|Y_{\mathrm{D}}|)$ *of output lengths the following equalities are executed:*

$$\mathbf{E}(|Y_{\mathrm{DT}}|) = n - 2 + 2^{r+1-n}, \tag{3.29}$$

$$\mathbf{E}(|Y_{\mathrm{D}}|) = 2 - 2^{r+1-n}, \tag{3.30}$$

$$\mathbf{D}(|Y_{\mathrm{DT}}|) = 2 - \frac{2n-2r-1}{2^{n-r-1}} - \frac{1}{4^{n-r-1}}, \tag{3.31}$$

$$\mathbf{D}(|Y_{\mathrm{D}}|) = 2 + \frac{(n-r)^2 + 2(n-r) - 1}{2^{n-r-1}} - \frac{1}{4^{n-r-1}}. \tag{3.32}$$

Proof. The proof of the statement 3.1 is obtained with the help of a direct computation.

In fact,the probability $P(Y_{\mathrm{DT}} \in \{0,1\}^k)$ is defined from (3.20), and $P(Y_{\mathrm{D}} = k)$ from (3.21). To prove the equality (3.29) and (3.30), it's enough to substitute corresponding probability values.

To prove the equality (3.31) and (3.32), it's enough to substitute corresponding probability values for dispersion definition and use the equality

$$\sum_{k=1}^{m} k^2 \cdot 2^{-k} = 6 - (m^2 + 4m + 6) \cdot 2^{-m}. \quad \square$$

3.1.4 MV2-transformation for obtaining harmed texts

Modern computer systems operate with machine words which are binary strings of a fixed bit that is 8, 16, 32 and so on bits as a rule. The minimal discrete is a bit that is a bit possessing the value 0 or 1 (*yes* or *no*). The minimal addressable discrete is usually a byte that equals eight bit. Assume that a plaintext consists of sequential discrete symbols, each of them is being chosen from an alphabet's finite set. In computer systems any text can be considered as a concatenation of bytes. The MV2-transformations that are defined on a binary string set of a fixed length allow extending definitional domain for a set of texts represented by symbols of the binary alphabet $\{0,1\}^n$.

Let an alphabet of the set of initial text coincide with $\{0,1\}^n$. Then the plaintext M is a concatenation of elements from $x_i \in \{0,1\}^n$:

$$M = x_1 \| x_2 \| \ldots \| x_L,$$

where L denotes the number of symbols in the text.

Definition 3.3 *Let $T = (c, f) \in \mathfrak{F}_n^r$ be an MV2-transformation and $M = x_1\|x_2\|\ldots\|x_L$ be a text representing a concatenation of L the symbols $x_i \in \{0,1\}^n$. We shall call the result of using the transformation T to the text M the pair $\Big(c(M), f(M)\Big)$ of binary strings obtained by the corresponding concatination of symbol images:*

$$\begin{array}{c} c(M) = c(x_1)\|c(x_2)\|\ldots\|c(x_L), \\ f(M) = f(x_1)\|f(x_2)\|\ldots\|f(x_L). \end{array} \tag{3.33}$$

And as before the image $c(M)$, obtained by using $T = (c, f)$ – an MV2- transformation – to the text M, we shall call *a remainder*, and $f(M)$ – *flags*. Thus, a remainder of the texts M is a concatenation if image remainders x_i, and flags of the text M is a concatenation of image flags x_i – see (3.33).

We shall make an example.

Example 3.1 *We shall consider an MV2-transformation that is defined by the Table 3.3.*

We take the plaintext

$$M = 0010\|0100\|0010\|1001\|0110\|0110\|0110\|1000$$

and execute the preset transformation:

M	*0010*	*0100*	*0010*	*1001*	*0110*	*0110*	*0110*	*1000*
$c(M)$	*10*	*10*	*10*	*001*	*010*	*010*	*010*	*00*
$f(M)$	*01*	*00*	*01*	*1*	*1*	*1*	*1*	*01*

We get the remainder
$c(M) = 1010\|1000\|1010\|0100\|1000$
and the flags $f(M) = 0100\|0111\|1101$.

About preimage quantity of a remainder and flags

Let $T = (c, f)$ be the fixed MV2-transformation and T is performed at the set of texts $\{M\}$, consisting of L symbols. The component c is a mapping, the output of which has a smaller length than the length of a plaintext, and several preimages can exist for a particular image.

Denote $Y_{\mathrm{DT}} = c(M)$ – a remainder of the text M, and $Y_{\mathrm{D}} = f(M)$ – flags of the text M. We shall further call the plaintext M as an input of the MV2-transformation, and an obtained remainder and flags of the text M – as outputs.

Let $l = |Y_{\mathrm{DT}}|$, it's evident, that

$$L \cdot r \ \le \ l \ \le \ L \cdot (n-1)$$

and the number of possible preimages of the remainder Y_{DT} depends on its length. We shall express this dependence.

Let $\widetilde{K}_{l\ s}(r, n)$ indicate the number of various integer solutions of equation:

$$z_1 + \ldots + z_s = l, \quad r \le z_i < n, \ i = 1, \ldots, s. \tag{3.34}$$

According to [23, c. 215] a number of solutions of the equation (3.34) will equal:

$$\widetilde{K}_{l\ s}(r, n) = \sum_{k=0}^{m} (-1)^k \binom{s}{k} \binom{l - rs - (n-r)k + s - 1}{s-1},$$

where the limit superior $m = \min\left\{ s, \dfrac{l - sr}{n - r} \right\}$.

Let $N_l^{(c)}$ be the number of various preimages for an image of the length l. Then

$$N_l^{(c)} = 2^L \cdot \delta(l - rL) + \sum_{i=0}^{L-1} 2^i \binom{L}{i} \widetilde{K}_{l-r\cdot i\ L-i}(r+1, n), \tag{3.35}$$

$\delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$ – a saltus function.

Note, that for $N_l^{(c)}$ the following estimation is true:

$$N_l^{(c)} \leq 2^{nL-l}, \quad rL \leq l \leq (n-1)L. \tag{3.36}$$

As the possible number of texts, consisting of the concatenation L n-bits strings, equals 2^{nL}, the following equation is executed:

$$\sum_{l=rL}^{(n-1)L} 2^l \cdot N_l^{(c)} = 2^{nL}. \tag{3.37}$$

In Fig. 3.1 there's a chart of dependence of preimage number on an image length for 7-byte inputs and the MV2-transformation with the parameters $r = 3$ and $n = 8$.

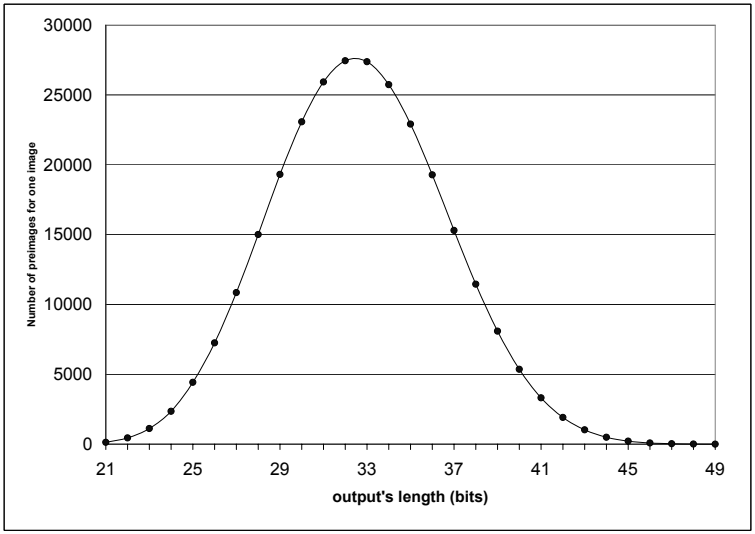

Fig. 3.1: *Distributing preimage number per image depending on its length when* $L = 7,\ n = 8,\ r = 3$

We shall estimate a number of possible preimages for flags output. An output of the flags

$$Y_{\mathrm{D}} = f(M) = f(x_1)\| \dots \|f(x_L)$$

can possess $(n-r+1)^L$ various values. We shall indicate k_j – a multiplicity with wich the value $j = 1, \dots, n-r+1$ enters Y_{D}. Then

$$\sum_{j=1}^{n-r+1} k_j = L, \qquad 0 \le k_j \le L \tag{3.38}$$

and a number of flag preimages $N^{(f)}$ will equal

$$N^{(f)} = 2^{k_{n-r+1}} \cdot \prod_{j=1}^{n-r+1} 2^{(n-j)\cdot k_j}. \tag{3.39}$$

Due to (3.5) there are $\prod_{i=1}^{n-1} 2^i!$ mappings $T = (c, f) \in \mathfrak{F}_n^r$, giving the same flag images in all texts.

Evaluations of output lengths of an MV2-mapping

Output lengths are of great importance for analysis of MV2- transformations. It's impossible to evaluate estimated values of output lengths in a general case. Therefore we won't go beyond a special case when the text M going to the input of the transformation, is randomly and uniformly chosen from the set $\{0,1\}^{nL}$. There's an example of probability distribution of remainder lengths for 7-byte equiprobable inputs in case of an arbitrary MV2-transformation with the parameters $r = 3$ and $n = 8$ in Fig. 3.2.

Let $T = (c, f)$ be an MV2-transformation.

$$M = x_1\| \dots \|x_L$$

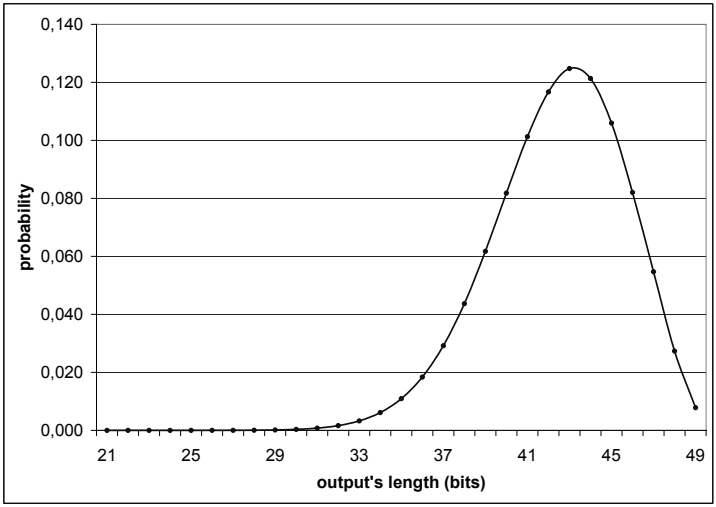

Fig. 3.2: *Chart of probability distribution of remainder output lengths at a uniform distribution of 7-byte inputs for an arbitrary fixed MV2-transformation with the parameters* $n = 8,\ r = 3$.

is a plaintext consisting of L n-bit strings $x_i \in \{0,1\}^n$,

$$Y_{\mathrm{DT}} = c(M),\ Y_{\mathrm{D}} = f(M)$$

are images of the text M at the transformation T.

Then the expectation of the remainder output length will be:

$$\mathbf{E}(|Y_{\mathrm{DT}}|) = 2^{-n\cdot L} \sum_{i=r\cdot L}^{(n-1)\cdot L} i \cdot N_i^{(c)} \cdot 2^i,$$

where $N_i^{(c)}$ is a number of various preimages for the remainder output length i.

On the other hand, as the text M is being randomly and uniformly chosen from the set $\{0,1\}^{nL}$, the remainder output length can be considered as a total of integer independent random values $|c(X_i)|$, $X_i \in \{0,1\}^n$. Due to (3.29) for the

expectation of the remainder length at uniformly distributed inputs we have

$$\mathbf{E}(|Y_{\mathrm{DT}}|) = \left(n - 2 + 2^{r+1-n}\right) \cdot L.$$

Consequently, the following equality is executed:

$$2^{-n \cdot L} \sum_{i=r \cdot L}^{(n-1) \cdot L} i \cdot N_i^{(c)} \cdot 2^i = \left(n - 2 + 2^{r+1-n}\right) \cdot L. \qquad (3.40)$$

Similarly, the output length of the flags $|Y_D|$ can be considered as a total of independent integer random values $|f(X_i)|$, $X_i \in \{0,1\}^n$..

Then, due to (3.30) for the expectation of flags output length we have

$$\mathbf{E}(|Y_D|) = \left(2 - 2^{r+1-n}\right) \cdot L.$$

Thus, the following statement is true

Claim 3.2 *Let a text* $M = x_1 \| \dots \| x_L$, *consist of* L *symbols* x_i, *being independently of one another, randomly and uniformly chosen from* $\{0,1\}^n$ *and some* $T = \left(c, f\right) \in \mathfrak{F}_n^r$ *is set.* $Y_{\mathrm{DT}} = c(M)$ *and* $Y_{\mathrm{D}} = f(M)$ *are outputs of the remainder and flags obtained in the result of using the transformation* T *for the text* M, *and* $|Y_{\mathrm{DT}}|$, $|Y_{\mathrm{D}}|$ *are their lengths. Then, expectations of the remainder output lengths* $\mathbf{E}\Big(|Y_{\mathrm{DT}}|\Big)$ *and those ones of flags* $\mathbf{E}\Big(|Y_D|\Big)$ *are equal to:*

$$\mathbf{E}(\Big(|Y_{\mathrm{DT}}|\Big) = \Big(n - 2 + 2^{r-n+1}\Big) \cdot L; \qquad (3.41)$$

$$\mathbf{E}(\Big(|Y_{\mathrm{D}}|\Big) = \Big(2 - 2^{r-n+1}\Big) \cdot L; \qquad (3.42)$$

The statement 3.2 allows defining coefficients that show estimated decrease of output length of the MV2- transformation relative to the length of a plaintext.

Definition 3.4 *We shall call the number*

$$\mathrm{K_c} = 1 - \frac{2 - 2^{r-n+1}}{n}. \tag{3.43}$$

as a compression ratio of the remainder at the transformation $T = (c, f) \in \mathfrak{F}_n^r$

Definition 3.5 *We shall call the number*

$$\mathrm{K_f} = \frac{2 - 2^{r-n+1}}{n}. \tag{3.44}$$

as a compression ratio of flags at the transformation $T = (c, f) \in \mathfrak{F}_n^r$.

It's evident, that

$$\mathrm{K_c} = 1 - \mathrm{K_f}. \tag{3.45}$$

The following statement is executed for probability of deviation of output lengths from average values

Claim 3.3 *Let the text* $M = x_1 \| \dots \| x_L$, *consist of* L *symbols* x_i, *being independently of one another, randomly and uniformly chosen from* $\{0,1\}^n$ *and some* $T = \left(c, f\right) \in \mathfrak{F}_n^r$. *is set. Then for probabilities of length rejection of the obtained remainder* $|Y_{\mathrm{DT}}| = |c(M)|$ *from* $\mathbf{E}(|Y_{\mathrm{DT}}|)$ *and the one of the obtained flags* $|Y_{\mathrm{D}}| = |f(M)|$ *from* $\mathbf{E}(|Y_{\mathrm{D}}|)$ *the following is executed*

$$P\left(\left||Y_{\mathrm{DT}} - \mathbf{E}(|Y_{\mathrm{DT}}|)\right| < \sigma_c \cdot L\right) > 1 - \frac{1}{L^2}, \tag{3.46}$$

$$P\left(\left||Y_{\mathrm{D}}| - \mathbf{E}(|Y_{\mathrm{D}}|)\right| < \sigma_f \cdot L\right) > 1 - \frac{1}{L^2}, \tag{3.47}$$

where

$$\sigma_c = \sqrt{2 - (2n - 2r - 1) \cdot 2^{r-n+1} - 4^{r-n+1}},$$

$$\sigma_f = \sqrt{2 + ((n-r)^2 + 2(n-r) - 1) \cdot 2^{r-n+1} - 4^{r-n+1}}.$$

The proof of the statement 3.3 follows from the statements 3.1, 3.2 and from the Chebyshev ineqiality [25].

Information and statistical estimations for remainder and flags outputs

In a general case the distribution of probabilities of a remainder and flags outputs essentially differs from the one of probabilities of a plaintext. It's obvious, for instance, that at a uniform distribution of inputs probabilities of outputs are distributed non-uniformly. In Fig. 3.3 there's a chart of distribution of remainder image probability depending on their length for uniformly distributed 7-byte inputs.

As before, we shall assume, that M, Y_{DT} and Y_{D} are random elements with the corresponding probability distributions. For these random elements the equalities (3.6) – (3.13) are executed, if we put $X = M$.

As the number of elements in the remainder range is $2^{(n-1)L+1} - 2^{rL}$, then for the entropy of the remainder $H(Y_{\mathrm{DT}})$ the following inequality is executed

$$H(Y_{\mathrm{DT}}) \le \log\left(2^{(n-1)\cdot L+1} - 2^{r\cdot L}\right); \tag{3.48}$$

Similarly, a number of elements in the flags range is $(n-r+1)^L$, therefore for the entropy of the flags $H(Y_D)$ the following inequality is carried out

$$H(Y_D) \le L \cdot \log(n - r + 1); \tag{3.49}$$

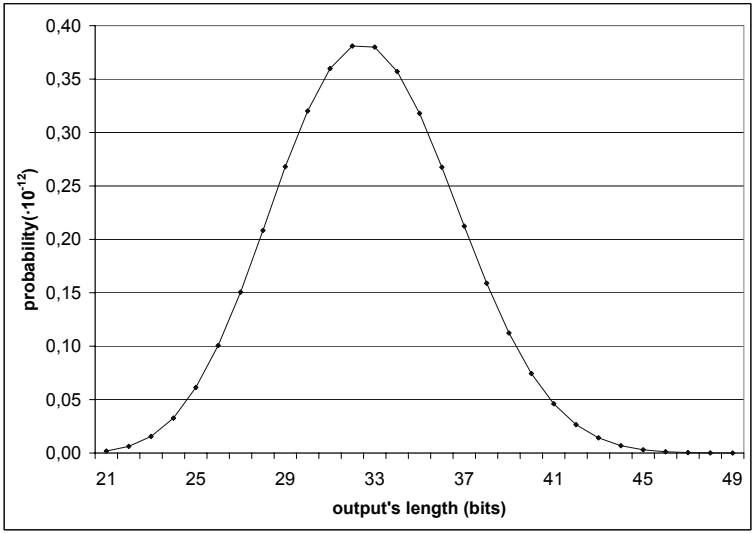

Fig. 3.3: *A chart of distribution of remainder probability depending on its length at uniformly distributed 7-byte inputs for an arbitrary fixed MV2- transformation with the parameters* $n = 8,\ r = 3$

Then we shall consider a case, when the text M is being randomly and uniformly chosen from the set $\{0,1\}^{nL}$.

Let's evaluate the entropy of the remainder output in this case.

From (3.35) we have

$$H(Y_{\mathrm{DT}}) = \sum_{i=rL}^{(n-1)L} 2^{i-nL} N_i^{(c)} \log \frac{2^{nL}}{N_i^{(c)}}.$$

Due to the inequality (3.36) we have

$$H(Y_{\mathrm{DT}}) \geq 2^{-nL} \sum_{i=rL}^{(n-1)L} i \cdot 2^i N_i^{(c)} \geq (n - 2 + 2^{r+1-n}) \cdot L.$$

Taking into account (3.48) we get the valuation:

$$\left(n-2+2^{r-n+1}\right)\cdot L \le H(Y_{\mathrm{DT}}) < (n-1)\cdot L + 1. \quad (3.50)$$

For a conditional entropy $H(M|Y_{\mathrm{DT}})$ the equality (3.11) is true, from which we have: $H(M|Y_{\mathrm{DT}}) = H(M) - H(Y_{\mathrm{DT}})$. Then, from (3.50) executing of the inequality follows

$$L - 1 \le H(M|Y_{\mathrm{DT}}) \le 2 - 2^{r+1-n}L. \quad (3.51)$$

Let's evaluate the entropy of flags outputs in case, when the text M is randomly and uniformly chosen from the set $\{0,1\}^{nL}$.

In this case $Y_{\mathrm{D}} = f(M) = f(x_1)\|\ldots\|f(x_L)$ can possess $(n-r+1)^L$ various values:

$$1 \le f(x_i) \le n-r+1,\ i = 1\ldots L.$$

Through k_j we shall indicate multiplicity with which the value $j \in [1, n-r+1]$ enters Y_{D}. Then $\sum\limits_{j=1}^{n-r+1} k_j = L$ and from (3.39) it follows that probability of any image at a uniform distribution of inputs will be

$$P = 2^{k_{n-r+1}} \cdot \prod_{j=1}^{n-r+1} 2^{-j\cdot k_j},$$

where $\sum\limits_{j=1}^{n-r+1} k_j = L,\ k_j \ge 0.$

Therefor, using [25] for flags entropy from a message we have:

$$H(Y_{\mathrm{D}}) = L! \sum_{\sum\limits_{j=1}^{n-r+1} k_j = L} \left(2^{k_{n-r+1}} \prod_{i=1}^{n-r+1} \frac{2^{-ik_i}}{k_i!}\right)\left(-k_{n-r+1} + \sum_{j=1}^{n-r+1} jk_j\right),$$

where $k_i \geq 0,\ i = 1 \ldots n - r + 1$.

The number of items in the sum $\sum_{k_1+\ldots+k_{n-r+1}=L}$ is equal to the number of various non-negative integer solutions of the equation (3.38) and equals $\binom{n-r+L}{n-r} = \binom{n-r+L}{L}$ (see, for example, [25, chapter II]), and the concave parenthesis equals the output length. Therefore, the entropy of flags output just coincides with the expectation of the output length, and correspondingly from (3.42) it follows, that

$$H(Y_{\mathrm{D}}) = (2 - 2^{r+1-n}) \cdot L. \tag{3.52}$$

Then, owing to (3.7) and (3.12), in case of uniform distribution of inputs M the following is executed

$$H(M|Y_{\mathrm{D}}) = H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}) = (n - 2 + 2^{r+1-n}) \cdot L. \tag{3.53}$$

About distribution of bit remainder

Let c_k be the k-th bit of the remainder obtained during performing an MV2-transformation over a text. Due to the lemma 3.1 the number of preimages corresponding to $\mathrm{c}_k = 0$, and the number of preimages corresponding to $\mathrm{c}_k = 1$ are equal. Therefore, the following theorem is true.

Theorem 3.1 *Let $T = (c, f)$ be an MV2-transformation. $Y_{\mathrm{DT}} \in U_{r\ n-1}$ is an output of the remainder, $|Y_{\mathrm{DT}}|$ is a length of the remainder output and c_k – k-th bit of the remainder output. Then, at a uniform distribution of inputs the probability that the value of the k-th bit is* 0 *provided, that the output length is no less than k, is*

$$P\left(\mathrm{c}_k = 0 \;\middle|\; 0 \leq k \leq |Y_{\mathrm{DT}}|\right) = \frac{1}{2}. \tag{3.54}$$

According to the theorem 3.1 at uniform input distribution the reminder looks like a uniform text.

3.1.5 Composition of two MV2-transformations

Let $T = (c, f) \in \mathfrak{F}_n^r$ be an MV2- transformation, and $\mathcal{M} = \bigcup\limits_{i=1}^{L} \{0,1\}^{n \cdot i}$ be a set of texts containing from 1 to L symbols, and $M \in \mathcal{M}$ be a text. In this case texts from the set $\mathcal{M}$ are binary strings with a length divisible by n. It's evident, that $c(M)$ is a remainder output at the MV2- transformation under the text M, generally speaking, doesn't belong to the set $\mathcal{M}$, because its length in a general case is not divisible by n. On the other hand, the remainder $c(M)$ can always be augmented from the right by a bit string $\mathbf{b}(c, M)$ so, that the concatenation $c(M) \| \mathbf{b}(c, M)$ would belong to the set of texts $\mathcal{M}$. To be certain we shall assume that $\mathbf{b}$ is either an empty string or a binary string which contains from 1 to $n-1$ zeros. Similarly one can augment a flags output. After $\mathbf{A}_n$ we shall indicate the operation of augmentation from the right by zero bits of an arbitrary binary string till its length is divisible by n.

Let two MV2-transformations:

$$T_1 = (c_1, f_1), T_2 = (c_2, f_2) \in \mathfrak{F}_n^r.$$

be randomly selected. We shall indicate c_i' – a transformation being a composition of the mapping c_i and of the above described augmenter: $c_i'(M) = A_n \circ c_i(M) = A_n(c_i(M))$, $i = 1, 2$. Then, we can define the composition of MV2-transformations in the following way:

$$T_2 \circ T_1(M) = \Big(c_2'\Big(c_1'(M)\Big),\ f_2\Big(c_1'(M)\Big) \| f_1(M) \Big). \qquad (3.55)$$

Example 3.2 *Composition of two MV2-transformations. MV2-transformation s with the parameters $n = 4$ and $r = 2$*

The first transformation

x	*0000*	*0001*	*0010*	*0011*	*0100*	*0101*	*0110*	*0111*
c(x)	*01*	*110*	*10*	*111*	*01*	*11*	*010*	*11*
f(x)	*00*	*1*	*01*	*1*	*01*	*00*	*1*	*01*
x	*1000*	*1001*	*1010*	*1011*	*1100*	*1101*	*1110*	*1111*
c(x)	*00*	*001*	*100*	*10*	*101*	*000*	*011*	*00*
f(x)	*01*	*1*	*1*	*00*	*1*	*1*	*1*	*00*

The second transformation

x	*0000*	*0001*	*0010*	*0011*	*0100*	*0101*	*0110*	*0111*
c(x)	*100*	*001*	*101*	*00*	*011*	*10*	*110*	*010*
f(x)	*1*	*1*	*1*	*00*	*1*	*01*	*1*	*1*
x	*1000*	*1001*	*1010*	*1011*	*1100*	*1101*	*1110*	*1111*
c(x)	*11*	*01*	*00*	*10*	*01*	*11*	*111*	*000*
f(x)	*00*	*01*	*01*	*00*	*00*	*01*	*1*	*1*

A plaintext $M =$
0010‖0100‖1010‖1001‖0110‖1110‖0110‖1000‖1010‖1000‖1100‖1011‖0100‖1110‖1100‖1001

After the first transformation we have:
the remainder $C1 = 10‖01‖100‖001‖010‖011‖010‖00‖100‖00‖101‖10‖01‖011‖101‖001$
and the flags $F1 = 01‖01‖1‖1‖1‖1‖1‖01‖1‖01‖1‖00‖01‖1‖1‖1.$

We shall augment the remainder $C1$ *by zeros from the right and get the text* $(Text1)$*, that goes to the input of the second transformation.*
$Text1 = 1001‖1000‖0101‖0011‖0100‖0100‖0010‖1100‖1011‖1010‖01\mathbf{00}.$

We get a new remainder and flags:
$C2 =$ 01 11 10 00 011 011 101 01 10 00 011,
$F2 =$ 01 00 01 00 1 1 1 00 00 01 1.

Finally we have:
$C =$ 0111 1000 0110 1110 1011 0000 11**00**.
$F = F2‖F1 =$ 0100 0100 1110 0000 11**00** 0101 1111 1011 0110 0011 11**00.**

The following lemma takes place.

L e m m a 3.2 *For any text* $C \in \bigcup_{i=r}^{\infty} \{0,1\}^{i \cdot n}$ *and for any transformation* $T = (c, f) \in \mathfrak{F}_n^r$ *there's such a text* $M \in \bigcup_{i=r}^{\infty} \{0,1\}^{i \cdot n}$*, that* $c(M) = C$.

Solution of the lemma follows directly from the possibility of presenting $i \cdot n$ in form of a sum of integer numbers $i \cdot n = \sum k_i$, each of them $r \le k_i \le n - 1$.

According to the lemma 3.2 any text can be a remainder output at performing an MV2-transformation from a text.

Transformations $T \in \mathcal{F}_n^r$ don't form a group, as the following is true

Claim 3.4 *For any three mutually different transformation*

$$T_1 = (c_1, f_1), T_2 = (c_2, f_2), T_3 = (c_3, f_3) \in \mathcal{F}_n^r$$

there's such a $M \in \bigcup_{i=r}^{\infty} \{0,1\}^{i \cdot n}$, *that*

$$c_1\Big(\mathbf{A}_n(c_2(M))\Big) \neq c_3(M).$$

The proof of the statement 3.4 follows from the lemma 3.2, as one can choose such a text M, consisting of n^2 symbols, that the length $|c_3(M)| = (n-1)n$. In this case the length

$$\Big|c_1\Big(c_2(M)\Big)\Big| < (n-1)n.$$

It follows from the statement 3.4, that the composition of MV2-transformations is not an MV2-transformation, therefore, there won't be such an MV2- transformation that would let go back to a plaintext at one round from any core obtained in the result of several transformation rounds at one round.

3.2 A general scheme of harming

3.2.1 The device for harming

A system of harming can be considered as a ciphersystem, a ciphertext of which consists of two or more interrelated parts.

In [71] and [20] an encryption method with splitting into two output channels and a device of its implementation were suggested. The scheme of this device is showed in Fig. 3.4.

The device suggested in the application includes:

1) an input unit;
2) a database (actually a key);
3) a decision making unit;
4) a transformation unit;
5) a random number generator;
6) an accumulator of a transformed text;
7) an accumulator of auxiliary information;
8) a switch;
9) an output unit.

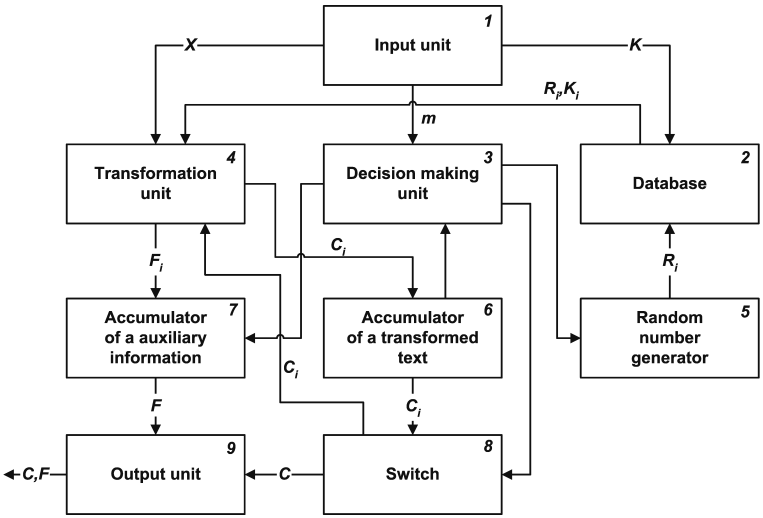

Fig. 3.4: *Method of encryption and transmission of information and device for its implementation.*

This device works in the following way:

1. A plaintext X goes to the transformation unit.

2. A round key K_i is generated from the key K and random data R_i obtained from the random number generator.

3. The transformation unit performs a round transformation the result of which is *a remainder* C_i which is placed to the accumulator of a transformed text and *flags* F_i which are already in the accumulator of auxiliary information.

4. When all the data from the accumulator of a transformed text are taken (a current round is over) the decision making unit is switched on which analyses whether the condition of finishing an encryption process was complied or not (i.e. either the required number of rounds m performed or the set remainder length is reached).

5. If the condition of finishing the encryption process is complied, the results (C, F) are sent to the output unit.

6. If the condition of finishing the encryption process is not complied, then the remainder C_i from the accumulator of a transformed text goes to the transformation unit and a new round is performed (the counter i increases by 1) (passing to the point 2).

The remainder C obtained after performing the last round forms a ciphertext together with all flags F.

3.2.2 A method of harming based on MV2-transformation

If we use an MV2-mapping as a transformer in this device we'll get a device for harming.

Each MV2-transformation $T = (c, f)$ maps an n-bit binary string x into a pair $(c(x), f(x))$, consisting of two variable length strings. It can be set by the table, in the left part of which there's a permutation of values from 1 to 2^n (see 3.1.2), and in the right part there are images consisting of "remainder" and "flag" parts.

Let us fix parameters r and n and chose an ordered set $T_1, T_2, \ldots, T_k \in \mathfrak{F}_n^r$ of random MV2-transformations. Further this set will be considered as a key.

We shall divide the whole process into rounds. At each round a permutation transformation and an MV2-transformation taken from the key will be performed over input data of the round. The output of an MV2-transformation is the remainder and the flags. We shall send an obtained remainder to the input of the following round, and accumulate the flags. Note, that a binary string of arbitrary length goes to the round input, and on the output we have two strings.

The number of rounds can be set directly or indirectly. As it is seen from the properties of MV2-transformations, the remainder has a length smaller than that of an input string. Therefore the threshold length for the length of the last remainder can be used as indicator stoping the transformation process. In this case rounds will be repeated till the remainder length smaller than the threshold appears.

We shall call the remainder of the last performed round as a *core*.

Such a scheme reminds a substitution permutation network

(SPN) (see [48]). The architecture of SPN is a fundamental architecture of block ciphers. It is based on principles of "confusion" and "diffusion", suggested by C. Shannon [92]. These principles are implemented with the help of substitution and permutation transformations. Permutation considerably complicates interrelations between statistical and analytical characteristics of open and encrypted texts. Dispersion spreads influence of particular bits of an open text on as much as possible number of a ciphertext bits. It also masks statistical interrelations and complicates cryptanalysis. One of the main methods is to interleave periodically diffusion (with considerably smaller tables) and permutation in the same cipher in various combinations. Cryptographic functions are implemented by means of combinations of substitution and permutation transformation. Permutation transformations are linear, and substitution ones are the main source of non-linearity in the cipher. A lot of works (see [48, 98, 32, 33, 76] and others) are dedicated to the criteria of choice of substitution transformations.

In the architecture of the offered scheme there's a significant difference from the architecture of SPN of block ciphers. At performing each round the whole text, rather than one block, is processed.

A suggested method of harming can be represented in form of a global structure, represented in Fig. 3.5.

The encryption process, performed according to this method, is divided into rounds. Linear and non-linear transformation alternate. Each round consists of a *linear layer* and a *none-linear layer*.

MV2-mappings are used to implement non-linear transformations. These mappings are set with the help of the secret tables which is key information.

At encryption a round procedure $Round(M)$ is used which

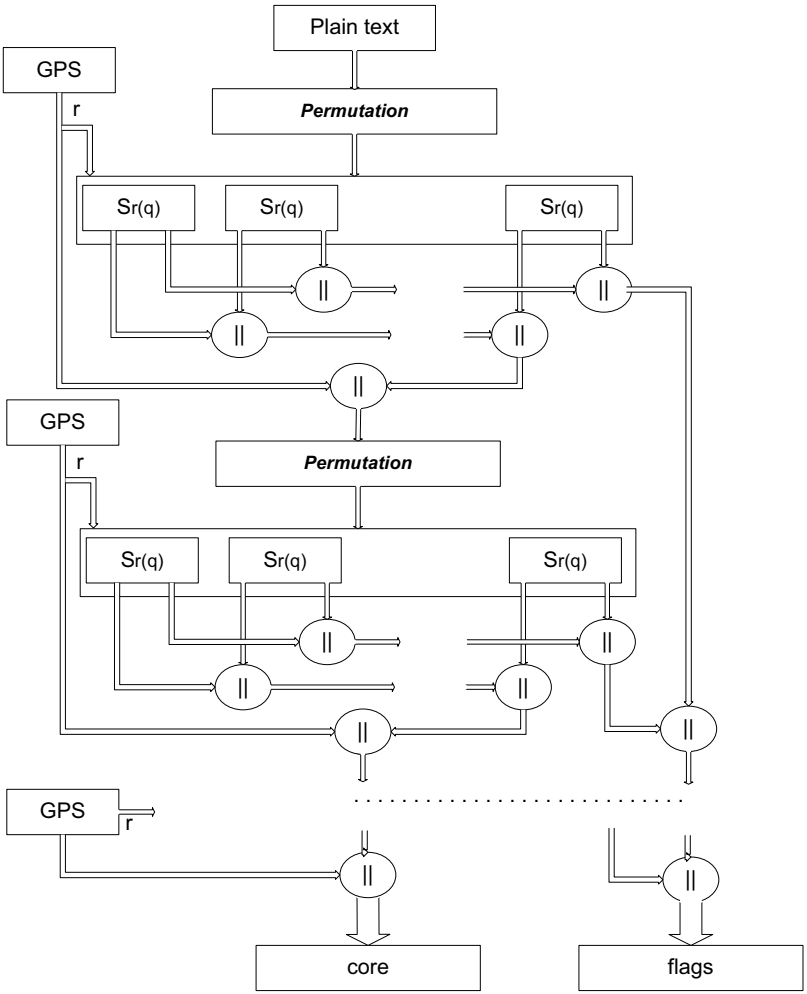

Fig. 3.5: *Presentation of the global structure as SPN. Here GPS – Random number generator, Permutation – a linear transformation, Sr(q) – a substitution transformation.*

is probabilistic and is performed according to the following scheme:

$$Raund(M) = \Big(R||C(K,R,M)\,,\ F(K,R,M)\Big),$$

where R is a randomly generated bit block, $C(K,R,M)$ and $F(K,R,M)$ are the first and the second output components of the substitution transformation, M is a round input and K is a key.

The remainder after one round is $R||C(K,R,M)$.

During decryption a deterministic procedure $Raund^{-1}$ works according to the following recurrent scheme:

$$(R_i\|C_i, F_i) = Raund^{-1}\Big(R_{i+1}||C_{i+1}\,,\ F_{i+1}\Big) \text{ for } 0 < i < P$$

$$\text{and } (M,\Lambda) = Raund^{-1}\Big(R_1||C_1\,,\ F_1\Big),$$

where Λ – is the empty string.

As we have already said the number of rounds can be set directly or indirectly by indicating the maximal core length. Besides, if the maximal core length is set, the number of rounds is determined automatically as soon as the remainder obtained at some round has a length less than the set one.

3.2.3 Preliminary analysis of the general scheme which uses MV2-transformations

It is accepted that evaluation of cipher security is executed by building attacks or by indirect features. At that in traditional ciphers at least a ciphertext is considered to be known, and the task of a cryptanalyst is to recover the key or the plaintext.

For schemes of harming, similar to those considered above, a ciphertext consists of two parts – a core and flags. Therefore,

to use such schemes it's necessary to consider additional variants when only a core or a core and keys are known, or only flags or flags and keys are known.

First of all it's important to point out that in the given scheme a substitution transformation is defined at a set of 2^n elements, and a number of different symbols from an alphabet used in an input text can be considerably less than the overall number of symbols in the alphabet. It means that at initial rounds of harming the key won't be used completely, and, consequently, is too large.

Secondly, in the suggested scheme accumulation of all flag outputs obtained at each round takes place. It's evident, that the total size of the flags of all the rounds can be close to the size of a plaintext at performing large enough number of rounds. Though the statistics of flags in the general case considerably differs from the statistics of a plaintext, nevertheless a question arises: isn't it possible to restore information according to the known flags only, for instance, on the basis of frequency analysis of separate symbols, especially at known keys. For some families of texts this question can be answered positively, that gives concern for cryptographic security of the suggested scheme.

In ciphersystems randomization of an open text is used to resist attacks based on frequency analysis. Randomization is an old technique and can be performed in different ways. One of the ways is "whitening"from a random (pseudorandom) number generator (GPS).

Using a stream cipher for whitening, firstly, leads to substitution of a message alphabet for an alphabet of this cipher, and, secondly, counteracts attacks based on frequency analysis. In this case we deal with the second method of harming (see 1.7).

A round of the general scheme of harming

As we have already mentioned, the whole process of harming in the suggested scheme is divided into rounds. Each round consists of permutation and substitution transformations. A permutation transformation provides bit dispersion of a plaintext which goes to the input of a round. An ideal variant is a bit permutation of the entire plaintext. But the authors don't know the algorithms that would implement such a permutation transformation. Therefore, we use a local permutation. One of the variants is the following one: an input string is divided into l-bit blocks, and a transformation is carried out under each of them.

After the permutation transformation one of MV2-transformations is carried out, which is chosen from the key according to a random value R, that is obtained from a GPS. The outputs of this transformation are the remainder and the flags. As the remainder goes to the input of the next round, its length should be divisible by n bit and a binary string should be added to it in the general case.

Randomization in the general scheme of harming

Determinate cryptosystems have leak of information; for example it's easy for an adversary to determine a situation when the same message is sent repeatedly. Another disadvantage of these systems becomes apparent when little message space is used. A brute force attack is possible in this case. If a plaintext has a lot of ciphertexts it leads to additional indeterminacy at a cryptanalysis.

A general scheme of harming allows performing randomization of a plaintext, for example, with the help of whitening. The general scheme also allows performing randomization of a cipher (see [21, 22]). For example, a substitution transformation can be randomly chosen from the key at every round.

For this purpose a random (pseudorandom) number generator (GPS) can be used. The nature of a generator doesn't matter for a cipher, therefore, any GPS, including a physical one, can be used.

When using a good GPS for the same plaintext k^m various output texts are possible if m encryption rounds are performed.

3.2.4 Examination of the general scheme

Evaluation of output lengths

Let a plaintext M, consisting of L n-bit binary strings (codes of symbols of a plaintext) go to the input of the device. We denote expectations of the number of symbols (n-bit binary strings) by $\mathbf{E}(L_m^{(c)})$ and $\mathbf{E}(L_m^{(f)})$, accordingly, the cores and the total flags after executing m rounds of the transformation.

As whitening of a plaintext M is executed before performing the main algorithm rounds, then, an input text of the first round can be considered as even.

From the statement 3.3, using the coefficients (3.43) and (3.44) introduced in the definitions 3.4 and 3.5 we have:

$$L_1^{(c)} \approx \mathrm{K_c} \cdot L$$
$$L_1^{(f)} \approx \mathrm{K_f} \cdot L.$$

Let's assume that s n-bit binary strings are added to every remainder, as texts obtained after an MV2 transformation should be augmented by bits till their length is divisible by n, and as in the general scheme it can be required to transmit the number of the transformation chosen from the key. Therefore, if C_i is a remainder after the i-th round and $L_i^{(c)}$ – the number of n-digit binary strings from which it consists, then,

$$L_i^{(c)} \approx \mathrm{K_c} \cdot L_{i-1}^{(c)} + s.$$

Consequently for the flags of the i-th round we have

$$L_i^{(f)} \approx \mathrm{K_f} \cdot L_{i-1}^{(c)} + 1.$$

Then at the m-th round (for great enough m) an estimated number of symbols in the remainder is:

$$L_m^{(c)} = {\mathrm{K_c}}^m \cdot \left(L + s \cdot \sum_{i=1}^{m} {\mathrm{K_c}}^{-i} \right) = {\mathrm{K_c}}^m \cdot L + s \cdot \frac{1 - {\mathrm{K_c}}^m}{1 - \mathrm{K_c}}.$$

From which, using (3.45), we have

$$L_m^{(c)} \approx {\mathrm{K_c}}^m \cdot L + \frac{1}{\mathrm{K_f}} \cdot s. \tag{3.56}$$

Then, an estimated number of symbols in the flag output will be:

$$L_m^{(f)} = \mathrm{K_f} \cdot L + \mathrm{K_f} \cdot \sum_{i=1}^{m-1} \left(L \cdot {\mathrm{K_c}}^i + \frac{1 - {\mathrm{K_c}}^i}{\mathrm{K_f}} \cdot s \right) =$$

$$= (1 - {\mathrm{K_c}}^m) L + \left(m - 1 - \frac{1 - {\mathrm{K_c}}^m}{\mathrm{K_f}} \right) \cdot s.$$

From where

$$L_m^{(f)} \approx L + m \cdot s. \tag{3.57}$$

If at encryption there is a requirement, that a number of n-bit symbols in the core output shouldn't exceed a threshold L_c, then, due to (3.56), the estimated number of rounds m_L is roughly:

$$m_L \approx 1 + \frac{\log L_c - \log L}{\log \mathrm{K_c}}. \tag{3.58}$$

From the statement 3.3 it follows, that at executing a round of a transformation under a text the length of which is L bytes with the probability no less than $1 - L^{-2}$ the number

of n-bit symbols in the obtained remainder will be within the limits of :

$$(\mathrm{K_c} - \sigma_c/n) \cdot L \leq |C|/n \leq (\mathrm{K_c} + \sigma_c/n) \cdot L,$$

and the number of n-bit symbols in the obtained flags:

$$(\mathrm{K_f} - \sigma_f/n) \cdot L \leq |F|/n \leq (\mathrm{K_f} + \sigma_f/n) \cdot L,$$

where σ_c and σ_f are root-mean-square deviations of lengths of output texts from evaluated average values.

Evaluation of a number of texts having the known remainder and unknown flags

If a round permutation is fixed, then a set of texts giving the same remainder is only determined by the substitution transformation $T \in \mathcal{F}_n^r$.

If only the core is known, and there's no limitations for the number of rounds, then, even if the keys are known there's an infinite set of texts giving such a core. At a limited number of rounds a set of plaintexts that corresponds to the given core is finite.

If the number of rounds m is known, then, from the expression (3.58) we can get evaluation for the number of symbols in a text which went to the input of the round.

In the considered scheme an MV2-transformation is performed at each round. As it is shown in 3.1.4 for the known core C of the length $|C| = l$ bits the number of its preimages for one round is determined by the expression (3.35).

As whitening is used in the input of the scheme, then, according to the theorem 3.1 the output of the remainder can be considered as an even sequence. Therefore, to evaluate the number of preimages N_m of the core C after performing m

rounds of the scheme one can use the inequality (3.51). Due to this inequality we have:

$$\log N_m > \frac{1}{\mathrm{K_c}}\frac{|C|}{n} + \ldots + \frac{1}{{\mathrm{K_c}}^m}\frac{|C|}{n},$$

From where we get:

$$N_m \geq 2^{\frac{1-{\mathrm{K_c}}^m}{\mathrm{K_f}}\cdot\frac{|C|}{n}}. \tag{3.59}$$

Accordingly, if L is a number of symbols in a plaintext received at the system's input is known, then for the evaluation of N_L from the inequality (3.59) we have

$$N_L \geq 2^{\frac{1-{\mathrm{K_c}}^m}{\mathrm{K_f}}\cdot L}. \tag{3.60}$$

Evaluations of the number of texts having known flags and an unknown remainder

If only flags are known, but the length of a plaintext and a number of rounds are unknown, then, probability of guessing a plaintext decreases sharply.

If a number of encryption rounds m, is known, then, the length of the core $|C|$ can be evaluated in average as

$$\mathbf{E}(|C|) \approx \frac{{\mathrm{K_c}}^m}{1-{\mathrm{K_c}}^{m+1}}\cdot|F|, \quad \mathrm{K_c} = \frac{97}{128},$$

where $|F|$ is a total length of output flags (bits).

Consequently, at known outputs of the flags F and a known key the number of plaintexts N_F, corresponding to the output of the flags F will be on average equal to

$$N_F \approx k\cdot 2^{(n-2+2^{r+1-n})|F|/n}, \tag{3.61}$$

Note that the expression (3.61) is true for large enough m and long enough flags F.

Evaluations of the number of texts at an unknown key

Let's remind that a key K is an ordered set from k transformations:

$$T_i = (c_i, f_i) \in \mathfrak{F}_n^r,\ i = 1, \dots k.$$

At each j-th round, where $j = 1, \dots, m$, a permutation transformation and some randomly chosen i_j-th substitution transformation T_{i_j} from the set K are performed. When a plaintext M goes to the input of the algorithm, the output of the cryptographic algorithm MV2 after m rounds is the core $C = c_{i_m}(C_{m-1})$ and the concatenation:

$$F = f_{i_m}(C_{m-1}) \| f_{i_{m-1}}(C_{m-2}) \| \dots \| f_1(C_0),$$

where C_0 coincides with the plaintext M, and $C_j = c_{i_j}(C_{j-1})$, $j = 1, \dots m$.

As it was mentioned in 3.1.2, the power of the set of substitution transformations

$$\#\mathfrak{F}_n^r = 2^n!.$$

If (C, F) is an image of a text M at performing an unknown transformation $T \in \mathfrak{F}_n^r$, then, for long enough texts there are minimum

$$\prod_{i=r}^{n-1} (2^i)! \tag{3.62}$$

different transformations giving the same outputs though for other texts, i.e.:

$$\#\{T_i \in \mathfrak{F}_n^r \,|\, \exists M_i \,:\, T_i(M_i) = (C, F)\} \geq \prod_{i=r}^{n-1} 2^i!.$$

If the plaintext M contains all the values from $\{0,1\}^n$, then the exact equality is performed:

$$\#\{T_i \in \mathfrak{F}_n^r \,|\, \exists M_i \,:\, T_i(M_i) = (C,F)\} = \prod_{i=r}^{n-1} 2^i!.$$

At known outputs (C,F) the number of performed rounds m can be determined from the expressions (3.58), (3.56), (3.57). Thus for a long enough core C at an unknown key the number of possible plaintexts N_K can be calculated by the formula:

$$N_K = \left(\prod_{i=r}^{n-1} 2^i!\right)^m. \tag{3.63}$$

3.3 Two channel encryption algorithm MV2

We have already mentioned that popular modern computer systems work with data the minimal addressable unit of which is a byte that equals eight bit. Consequently, it's reasonable to choose a parameter n, divisible by 8: 8, 16, 32, ... and so on at building a device of harming designed to work in such systems.

On the basis of the scheme of harming suggested in 3.2.2 we created a symmetric probabilistic cipher which implements the universal mechanism of harming – the MV2 algorithm.

To perform a permutation transformation we use two procedures: **Mix** – at encryption, and the reverse one – **ReMix** – at decryption, and Λ defines a blank string.

In this cipher encryption is executed according to the following algorithm:

Encryption

Input: a plaintext $M\,(8 \times L(bits))$
a private key $K = T_1, \ldots, T_k,\ T_i \in \mathfrak{F}_8^3$
a number of rounds m

BEGIN

$C_0 = M;\ F = ();$ (*() – empty string *)

choose random $j_0 \in \{1, \ldots, k\}$ (* select the number of substitution transformation; *)

whitening of the plaintext: $C_0 = j_0 \| (C_0 \oplus GPS(T_{j_0}))$;

For $i = 1;$ **To** $i = m$ **Do** – m rounds are performed;

1. $C_{i-1} = \mathbf{Mix}(C_{i-1})$ – a remainder of a previous round is permuted block by block.
2. choose random $j_i \in \{1, \ldots, k\}$ (* select the number of substitution transformation; *)
3. $T_{j_i}(C_{i-1}) = \Big(C_i, F_i\Big)$ – a substitution transformation T_j is carried out. In the result we obtain a new remainder C_i and flags F_i.
4. $C_i = j_i \| C_i;\ F = F_i \| F.$

   End(Do)

End(For).

$C = C_m$;

END.

Output: Ciphertext $\Big(C\,,\ F\Big)$.

Decryption is executed according to the following algorithm:

Decryption

Output: Ciphertext $\Big(C\,,\,F\Big)$

A private key $K = T_1, \ldots, T_k,\ T_i \in \mathcal{F}_8^3$

BEGIN

$C_m = C$;

While $F \neq ()$

1. Parse $C = j \| C_m$;
2. $C := ()$;
3. **While** $(C_m \neq ())$ **do**

   $f := ()$; extract the first code of the flag f from F and delete it from F;

   Mark a prefix substring c from C_m, the length of a string c is determined according to the flag f;

   Delete c from C_m;

   $x := T_j^{-1}(c, f)$;

   $C := C \| x$;

   If $f = ()$ AND $C_m \neq ()$ **then** $C :=$ 'error message'; **break**.

   End(While)

4. **If** (not 'error message') **then**
   $C =$ **ReMix**(C_m) // shuffle the result.
   end(If)

End(While).

If ($\neq$ 'error message') **then**

Extract j from $C = j \| C_m$;

$C = C_m \oplus GPS(T_j)$ (* remove whitening of the plaintext *)

end (If)

$M = C$;

END;

Output: Error message
or a plaintext M $(8 \times L)$ bits.

This algorithm describes a general method of harming for byte-oriented texts. The exact implementation of the MV2 algorithm and results of testing are described in B.

3.4 Shannon's security model for two-channel ciphers

3.4.1 Shannon's security model

C. Shannon [92] introduced a model of security known as *perfect security* or *unconditional security*. It is assumed that an adversary possesses infinite computational powers.

Let M be a plaintext, K will be a key and $Y = E(M, K)$ will be a cryptogram. C. Shannon formulated a property of a perfect security as

$$H(M) = H(X|Y)$$

and showed that the following inequality should be performed for a perfect cipher:

$$H(K) \geq H(M).$$

In this model it is supposed that a cryptanalyst has a possibility to observe a ciphertext Y.

For practical cipher security it's required that the inequalities

$$H(K|Y) \geq \alpha, \qquad H(M|Y) \geq \alpha$$

are performed for values of the conditional entropies $H(K|Y)$ and $H(M|Y)$.

Nowadays for practical cipher resistance it's considered that the value

$$\alpha > 80.$$

We shall consider a cipher from the item 1.3, described by the system. Such systems can be used in different modes. Let's enumerate (formally) possible usage modes of such systems and requirements for their security.

1. A key K is transmitted via a secure channel, and the outputs $Y_{\mathrm{DT}}, Y_{\mathrm{D}}$ – via an open channel. As a cryptanalyst has a possibility to observe both the output Y_{DT} and Y_{D}, this case is similar to that one of a classical symmetric system considered by Shannon, and for perfect security in this case it's obvious to perform the equation:

   $$H(M) = H(M|Y_{\mathrm{DT}}Y_{\mathrm{D}}).$$

   And for practical security the necessary condition is:

   $$H(M|Y_{\mathrm{DT}}Y_{\mathrm{D}}) \geq \alpha.$$

2. The key K and the output Y_{DT} are transmitted via a secure channel, and the output Y_{D} – via an open channel. In this case an adversary cryptanalyst observes only a

part of a cryptotext, and it's obvious that in this case for perfect security it's necessary to perform the equality

$$H(M) = H(M|Y_{\mathrm{D}}).$$

And for practical security the necessary condition is

$$H(M|Y_{\mathrm{D}} \geq \alpha.)$$

3. The output Y_{DT} is transmitted via a secure channel, and the key K and the output Y_{D} – are via an open channel. In this case an adversary cryptanalyst observes only a part of a cryptotext and the key K, therefore, it's evident, that in this case for perfect security it's necessary to perform the equality

$$H(M) = H(M|Y_{\mathrm{D}}K).$$

And for practical security the necessary condition is

$$H(M|Y_{\mathrm{D}}K \geq \alpha.)$$

4. The key$K$ and the output Y_{D} are transmitted via a secure channel, and the output Y_{DT} – via an open channel. In this case an adversary cryptanalyst observes only a part of a ctyptotext, and in this case for perfect security it's necessary to perform the equality

$$H(M) = H(M|Y_{\mathrm{DT}}).$$

And for practical security the necessary condition is

$$H(M|Y_{\mathrm{DT}} \geq \alpha.)$$

5. The output Y_{D} is transmitted via a secure channel, and the key K and the output Y_{DT} – via an open channel. In this case an adversary cryptanalyst observes only a part of a ctyptotext and the key K, therefore, in this case for perfect security it's necessary to perform the equality

$$H(M) = H(M|Y_{\mathrm{DT}}K).$$

And for practical security the necessary condition is

$$H(M|Y_{\mathrm{DT}}K \geq \alpha.)$$

6. The key K and a plaintext M are transmitted via a secure channel, and the outputs $Y_{\mathrm{DT}}, Y_{\mathrm{D}}$ – via an open channel. This case is similar to the case of a classical symmetric system considered by Shannon.

7. The key K and the output Y_{DT} are transmitted via a secure channel, and the output Y_{D} and a plaintext M – via an open channel. In this case condition of perfect security can be formulated in form of

$$H(K) = H(K|Y_{\mathrm{D}}M),$$

and condition of practical security

$$H(K) = H(K|Y_{\mathrm{D}}M) \geq \alpha.$$

If a cryptanalyst solves the problem of defining the output Y_{DT} only, then, for perfect security it's necessary to perform the equality

$$H(Y_{\mathrm{DT}}) = H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}M),$$

and for the practical one –

$$H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}M) \geq \alpha.$$

8. The output Y_{DT} and a plaintext M are transmitted via a secure channel, and the key K and the output Y_{D} – via an open channel. In this case condition of perfect security can be formulated in form of

$$H(M) = H(M|Y_{\mathrm{D}}K),$$

and condition of practical security

$$H(M) = H(M|Y_{\mathrm{D}}K) \geq \alpha.$$

If a cryptanalyst solves the problem of defining the output Y_{DT} only, then for perfect security it's necessary to perform the equality

$$H(Y_{\mathrm{DT}}) = H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}K),$$

and for the practical one –

$$H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}K) \geq \alpha.$$

9. The key K and the output Y_{D} are transmitted via a secure channel, and the output Y_{DT} and a plaintext M – via an open channel. In this case condition of perfect security can be formulated in form

$$H(K) = H(K|Y_{\mathrm{DT}}M),$$

and condition of practical security

$$H(K) = H(K|Y_{\mathrm{DT}}M) \geq \alpha.$$

If a cryptanalyst solves the problem of defining the output Y_{D} only, then, for perfect security it's necessary to perform the equality

$$H(Y_{\mathrm{D}}) = H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}M),$$

and for the practical one –

$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}M) \geq \alpha.$$

10. The output Y_{D} and a plaintext M are transmitted via a secure channel, and the key K and the output Y_{DT} – via an open channel. In this case condition of perfect security can be formulated in form

$$H(M) = H(M|Y_{\mathrm{DT}}K),$$

and condition of practical security

$$H(M) = H(M|Y_{\mathrm{DT}}K) \geq \alpha.$$

If a cryptanalyst solves the problem of defining the output Y_{D} only, then, for perfect security it's necessary to perform the equality

$$H(Y_{\mathrm{D}}) = H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}K),$$

and for the practical one –

$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}K) \geq \alpha.$$

3.4.2 General information ratios for two-channel systems

Let M be a random message from a set of messages $\mathcal{M}$, K is a key which is randomly and uniformly chosen from the set $\mathcal{K}$, and Y_{DT} andY_{D} are outputs in accordance with the system (1.2).

In the general case irrespective of a certain kind of the used transformations entropies of inputs, outputs and keys of two-channel systems formed in accordance with the system

(1.2) are interconnected by the following information ratios:

$$\begin{aligned} H(M, Y_{\mathrm{DT}}, Y_{\mathrm{D}}, K) = \\ = H(MK) + H(Y_{\mathrm{DT}}Y_{\mathrm{D}}|MK) = \\ = H(M) + H(K) = \\ = H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) + H(KM|Y_{\mathrm{DT}}Y_{\mathrm{D}}) = \\ = H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) + H(K|Y_{\mathrm{DT}}Y_{\mathrm{D}}). \end{aligned} \tag{3.64}$$

It follows from these ratios:

$$H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) + H(K|Y_{\mathrm{DT}}Y_{\mathrm{D}}) = H(M) + H(K). \tag{3.65}$$

For the joint entropy Y_{DT}, Y_{D} and K the following chain of equations is performed:

$$\begin{aligned} H(Y_{\mathrm{DT}}Y_{\mathrm{D}}K) = H(Y_{\mathrm{DT}}Y_{\mathrm{D}}) + H(K|Y_{\mathrm{DT}}Y_{\mathrm{D}}) = \\ = H(Y_{\mathrm{DT}}) + H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) + H(K|Y_{\mathrm{DT}}Y_{\mathrm{D}}) = \\ = H(Y_{\mathrm{D}}) + H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}) + H(K|Y_{\mathrm{DT}}Y_{\mathrm{D}}) = \\ = H(K) + H(Y_{\mathrm{DT}}Y_{\mathrm{D}}|K) = \\ = [\text{ в силу (3.65)}] = H(M) + H(K). \end{aligned} \tag{3.66}$$

From where, in particular, performance of the equation follows:

$$H(Y_{\mathrm{DT}}Y_{\mathrm{D}}|K) = H(M). \tag{3.67}$$

Similarly, for a joint entropy of the input M, output Y_{DT} and key K Y_{D} the following chain of equations is performed:

$$\begin{aligned} H(MY_{\mathrm{DT}}K) = H(Y_{\mathrm{DT}}) + H(MK|Y_{\mathrm{DT}}) = \\ = H(Y_{\mathrm{DT}}) + H(M|Y_{\mathrm{DT}}) + H(K|Y_{\mathrm{DT}}M) = \\ = H(Y_{\mathrm{DT}}) + H(K|Y_{\mathrm{DT}}) + H(M|Y_{\mathrm{DT}}K) = \\ = H(M) + H(Y_{\mathrm{DT}}K|M) = \\ = H(M) + H(Y_{\mathrm{DT}}|M) + H(K|Y_{\mathrm{DT}}M) = \\ = H(M) + H(K); \end{aligned} \tag{3.68}$$

Similarly for a joint entropy of the input M, output Y_{D} and key K the following equations are performed:

$$\begin{aligned} H(MY_{\mathrm{D}}K) &= H(Y_{\mathrm{D}}) + H(MK|Y_{\mathrm{D}}) = \\ &= H(Y_{\mathrm{D}}) + H(M|Y_{\mathrm{D}}) + H(K|Y_{\mathrm{D}}M) = \\ &= H(Y_{\mathrm{D}}) + H(K|Y_{\mathrm{D}}) + H(M|Y_{\mathrm{D}}K) = \\ &= H(M) + H(Y_{\mathrm{D}}K|M) = \\ &= H(M) + H(Y_{\mathrm{D}}|M) + H(K|Y_{\mathrm{D}}M) = \\ &= H(M) + H(K). \end{aligned} \tag{3.69}$$

The ratios (3.64) – (3.69) can be applied for evaluating security of using a universal algorithm of harming described in 3.2.2 and 3.3.

3.5 Analysis of security of using a universal algorithm of harming

3.5.1 Analysis of the general scheme security at unknown flag output

If the flag output is unknown, then depending on a usage mode an attacker can have different information.

1. An attacker knows the output Y_{DT}. In this case the attacker breaking a two-channel system can have the following tasks:

   1.a) Find the plaintext M;

   1.b) Find the key K;

   1.c) Find the output Y_{D};

   1.d) Find the plaintext M and the key K;

1.e) Find the plaintext M and the output Y_{D};

1.f) Find the key K and the output Y_{D};

1.g) Find the plaintext M, the key K and the output Y_{D}.

2. An attacker knows the output Y_{DT} and the key K.In this case the attacker breaking a two-channel system can have the following tasks:

2.a) Find the plaintext M;

2.b) Find the output Y_{D};

2.c) Find the plaintext M and the output Y_{D};

3. An attacker knows the output Y_{DT} and the corresponding plaintext M. In this case the attacker breaking a two-channel system can have the following tasks:

3.a) Find the key K;

3.b) Find the output Y_{D};

3.c) Find the key K and the output Y_{D};

4. An attacker knows the output Y_{DT}, the key K and the corresponding plaintext M. In this case the attacker breaking a two-channel system can have the following tasks

4.a) He needs to determine the output Y_{D}.

Note, that there's no need to consider all problems, because if an attacker can't solve a problem where one of the components needs to be determined, he can't solve a corresponding problem where several components need to be determined.

Claim 3.5 *The complexity of solving the problems 1.a – 1.g, 2.a – 2.c, 3.a – 3.c – correlate in the following way.*

The problem 1.a, is not more difficult than the problems 1.d, 1.e, 1.g.

The problem 1.b, is not more difficult than the problems 1.d, 1.f, 1.g.

The problem 1.c, is not more difficult than the problems 1.e, 1.f, 1.g.

The problem 2.a, is not more difficult than the problem 2.c.

The problem 2.b, is not more difficult than the problem 2.c.

The problem 3.a, is not more difficult than the problem 3.c.

The problem 3.b, is not more difficult than the problem 3.c.

Proof As $H(M|Y_{\mathrm{DT}}) \geq H(M|Y_{\mathrm{DT}}K)$, then, the problem 1.a is not easier than the problem 2.a.

Due to $H(K|Y_{\mathrm{DT}}) \geq H(K|Y_{\mathrm{DT}}M)$, the problem 1.b is not easier than the problem 3.a.

Similarly, due to $H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) \leq H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}K)$ the problem 1.c is not easier than the problem 2.b, and due to $H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) \leq H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}M)$ the problem 1.c is not easier than the problem 3.b.

As the following ratios are performed

$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}K) \geq H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}KM),$$
$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}M) \geq H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}KM),$$

then the problems 2.b and 3.b are not easier than the problem 4.a $\square$.

Let's evaluate complexity of solving the problems put in front of an attacker.

To do that we shall assume that a round permutation is fixed. In this case a set of texts giving the same remainder at every transformation round is determined by the used substitution transformation $T \in \mathfrak{F}_n^r$. only.

Problem 4.a.

Let us know the plaintext M, the key K and the remainder output Y_{DT}. If at each round a substitution transformation is chosen according to the deterministic rule, then an attacker will definitely determine the flags, i.e. $H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}KM) = 0$.

We shall consider a case when a round substitution transformation is randomly chosen from a known key. In this case an attacker can evaluate the number of performed rounds m by the formula (3.58). For this case from the ratio (3.66) we have $H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) = m \cdot \log(k) - H(Y_{\mathrm{DT}})$. If the length of the output Y_{DT} is large enough, then in the general case an attacker has k^m variants of pairs $(Y_{\mathrm{DT}}, Y_{\mathrm{D}})$ and with probability close to 1, there's the only pair with the set core among the variants. Note that with increase of m, entropy $H(Y_{\mathrm{DT}})$ decreases. Therefore, required security of the cipher can be obtained thanks to increasing the number of performed rounds.

Problem 2.b.

Let the key K and the remainder output Y_{DT} be known. If at each round a substitution transformation is chosen by the deterministic rule, then, from the ratio (3.66) we have

$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) = H(M) - H(Y_{\mathrm{DT}}).$$

If a round substitution transformation is randomly chosen

from a known key, then the conditional entropy

$$H(Y_{\mathrm{D}}|Y_{\mathrm{DT}}) = m \cdot \log(k) + H(M) - H(Y_{\mathrm{DT}}).$$

Consequently, the required security of the cipher can be obtained thanks to increasing the number of performed rounds.

Problem 2.a.

Let the key K and the remainder output Y_{DT} be known. If the plaintext M is unknown and the number of performed rounds m is unknown, then for the known core Y_{DT} even at known keys K there's an infinite set of texts at which you can get such a core. At the limited number of rounds a set of texts corresponding to the given core is finite. Further we shall assume that an attacker knows the number of performed rounds m.

Let's evaluate the number of texts having the same remainder.

In the considered scheme an MV2-transformation is performed at each round. As it is shown in 3.1.4, for the known core C of the length $|C| = l$ bits, the number of its preimages for one round is determined by the expression (3.35).

As whitening is used at the scheme input, then, according to the theorem 3.1, the remainder output can be considered as a uniform sequence. Therefore, for evaluating the number of preimages N_m of the core C after performing m rounds of the scheme one can use the inequality (3.51). Due to this inequality we have:

$$\log N_m > \frac{1}{\mathrm{K_c}}\frac{|C|}{n} + \ldots + \frac{1}{\mathrm{K_c}^m}\frac{|C|}{n},$$

From where we have:

$$N_m \geq 2^{\frac{1-\mathrm{K_c}^m}{\mathrm{K_f}} \cdot \frac{|C|}{n}}. \qquad (3.70)$$

Accordingly, if L is the number of symbols in the plaintext which went to the scheme input is known, then, to evaluate N_L – the number of texts corresponding to the known core from the inequality (3.70) we have

$$N_L \geq 2^{\frac{1-\mathrm{K_c}^m}{\mathrm{K_f}} \cdot L}. \tag{3.71}$$

Thus, complexity of recovering the plaintext at known keys can be evaluated as

$$H(M|Y_{\mathrm{DT}}K) = O(\frac{1-\mathrm{K_c}^m}{\mathrm{K_f}} \cdot L).$$

Problem 3.a.

Let the plaintext M and the remainder output Y_{DT} are known. In this case from the ratios (3.68) we have:

$$H(K|Y_{\mathrm{DT}}M) = H(K) - H(Y_{\mathrm{DT}}).$$

If each transformation T_i from the key K is at random and uniformly chosen from a set of the transformations $\mathfrak{F}_n^r$, then entropy $H(K) = k \cdot \log(2^n!)$, using the Stirling formula we have

$$H(K) \approx k \cdot ((n - 1.443) \cdot 2^n + \frac{n+1}{2} + 0.826) \geq k \cdot (n-2) \cdot 2^n.$$

In this case, for secure use at $n \geq 8$ it's enough to require performing such the number of rounds that for the length of the core output the following inequality would perform

$$|Y_{\mathrm{DT}}| \leq (k-1)(n-3) \cdot 2^n \quad \text{bits}.$$

If a round substitution transformation is chosen at random from the key K, then for each m-round encryption of a message M entropy of the real key K^* will equal to

$$H(K^*) = H(K) + m \cdot \log k.$$

3.5.2 Analysis of the general scheme security at unknown core output

If the core output is unknown, then, depending on a usage mode an attacker can have different information.

5. An attacker knows the output Y_{D}. In this case the attacker breaking a two-channel system can have one of the following tasks:

   5.a) Find the plaintext M;

   5.b) Find the key K;

   5.c) Find the output Y_{DT};

   5.d) Find the plaintext M and the key K;

   5.e) Find the plaintext M and the output Y_{DT};

   5.f) Find the key K and the output Y_{DT};

   5.g) Find the plaintext M, the key K and the output Y_{DT}.

6. An attacker knows the output Y_{D} and the key K. In this case the attacker breaking a two-channel system can have one of the following tasks:

   6.a) Find the plaintext M;

   6.b) Find the output Y_{DT};

   6.c) Find the plaintext M and the output Y_{D};

7. An attacker knows the output Y_{D} and the corresponding plaintext M.In this case the attacker breaking a two-channel system can have one of the following tasks:

   7.a) Find the key K;

7.b) Find the output Y_{DT};

7.c) Find the key K and the output Y_{DT};

8. An attacker knows the output Y_{D} the key K and the plaintext M.In this case the attacker breaking a two-channel system can have one of the following tasks:

8.a) He needs to determine the output Y_{DT}.

As in the previous item there's no need to consider all problems, because if an attacker can't solve a problem where one of the components needs to be determined, he can't solve a corresponding problem where several components need to be determined.

Claim 3.6 *Complexity of solving the problems 5.a – 5.g, 6.a – 6.a, 7.a – 7.c correlate in the following way.*

The problem 5.a, is not more difficult than the problems 5.d, 5.e, 5.g.

The problem 5.b, is not more difficult than the problems 5.d, 5.f, 5.g.

The problem 5.c, is not more difficult than the problems 5.e, 5.f, 5.g.

The problem 6.a, is not more difficult than the problem 6.c.

The problem 6.b, is not more difficult than the problem 6.c.

The problem 7.a, is not more difficult than the problem 7.c.

The problem 7.b, is not more difficult than the problem 7.c.

As whitening of the plaintext by a stream cipher takes place before performing main transformation rounds, one can assume that the plaintext consists of symbols that are chosen equiprobably from $\{0,1\}^n$. In this case at analysis of the flags output $H(M) = n \cdot L$. Then, after performing the first round in the output we have the flag F_1 and the remainder C_1, for which $H(F_1) = \mathrm{K_f} \cdot n \cdot L$, and $H(C_1) \geq \mathrm{K_c} \cdot n \cdot L$. Due to the theorem about distributing bit remainder, an obtained remainder looks like a uniform text. And due to the statement about the expectation of the remainder length $|C_1| = \mathrm{K_c} \cdot n \cdot L$. From where

$$H(Y_\mathrm{D}) \geq n \cdot L \cdot (1 - {\mathrm{K_c}}^m). \tag{3.72}$$

Note that if in the general scheme of the cipher a substitution transformation is chosen at random from the fixed key K, then, there are k^m variants of the real key at performing m rounds of the algorithm for the set key K. Consequently, at equiprobable choice of variants for entropy of the real key used for the encryption K_{in} the following is performed

$$H(K_{in}) = H(K) + m \cdot \log k. \tag{3.73}$$

To determine complexity of these problems we shall use general information dependencies (3.65) – (3.67), (3.69).

From (3.66) we have:

$$H(Y_\mathrm{D}) + H(Y_\mathrm{DT}|Y_\mathrm{D}) + H(K|Y_\mathrm{DT}Y_\mathrm{D}) = H(M) + H(K). \tag{3.74}$$

From (3.69) we have:

$$H(Y_\mathrm{D}) + H(M|Y_\mathrm{D}) + H(K|Y_\mathrm{D}M) = H(M) + H(K). \tag{3.75}$$

Problem 8.a.

Let the plaintext M, the key K and the flag output Y_{D} be known. If at each round a substitution transformation is chosen by the deterministic rule, then an attacker will exactly determine the core.

Let a round substitution transformation be chosen from a known key. If a cryptanalyst knows the flags output Y_{D}, the key K and the plaintext M, then he can exactly determine the remainder output. Indeed, the key K consists of k MV2-transformations $T_i = (c_i, f_i)$, at that $k \ll \#\mathcal{F}$, therefore, the probability that among flags transformations there will be at least 2 identical of them is

$$1 - \frac{\#\mathcal{F} \cdot (\#\mathcal{F} - 1) \cdot \ldots \cdot (\#\mathcal{F} - s + 1)}{(\#\mathcal{F})^s} \approx 0,$$

where $\#\mathcal{F}$ can be found by the formula (3.5).

Then, comparing M and the first flags one can determine the transformation T_{i_1} which was used to obtain them, and consequently, get the remainder of the first round. A remainder of the first round is an input text of the second round, therefore, comparing it with the second flags one can determine the transformation T_{i_2} which was used to obtain them, and so on... till we shall get a remainder of the last round that is the required core.

Thus, $H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}KM) = 0$ and using mode 8 is not secure.

Problem 6.a.

Let the key K and the flag output Y_{D} be known. It's evident, that in this case the ratio $H(M|Y_{\mathrm{D}}K) = H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}K)$ is performed.

If the number of performed rounds m is known, then one may evaluate the input length M and the estimated output length $|Y_{\mathrm{DT}}|$ can be evaluated as $|Y_{\mathrm{DT}}| = \mathrm{K_c}^m \cdot |M|$.

If a round substitution transformation is chosen at random from the key K, then from the formulae (3.73) and (3.75) we have $H(M|Y_{\mathrm{D}}) = H(M) - H(Y_{\mathrm{D}}) + m \cdot \log k - H(K|Y_{\mathrm{D}}M)$.

From where, due to (3.75)

$$H(M|Y_{\mathrm{D}}) \geq {\mathrm{K_c}}^m \cdot n \cdot L + m \cdot \log k.$$

Therefore, for the complexity of the plaintext recovering by known the key and the flags output to satisfiy modern requirements it's necessary to demand:

$${\mathrm{K_c}}^m \geq \frac{\alpha - m \cdot \log k}{|Y_{\mathrm{D}}|}. \tag{3.76}$$

The requirement (3.76) is always performed from a pseudorandom choice of substitution transformations if $m \geq \dfrac{\alpha}{\log k}$ rounds are performed. Especially at $\alpha = 80$ for a basic implementation of the MV2 cipher the requirement is fulfilled after 16 transformation rounds irrespective of the length of an input message.

Note, that if we reject randomization of the MV2 cipher which occurs at each round due to random choice of a substitution transformation, then from (3.76) it follows, that $m \leq \dfrac{\log(n \cdot L) - 7}{\log(1/\mathrm{K_c})}$, i.e. the number of possible transformation rounds depends on the plaintext length. For instance, for the basic implementation of the cipher, in case of rejecting a pseudorandom choice of a substitution transformation at each round, for a 1 MB text it's secure to perform no more than 41 round, and for a 1 kB text – no more than 16 rounds, and for a 16-byte text – no more than one round.

Problem 6.b.

Let the key K and the flag output Y_{D} be known. Due to (3.53) the problems 6.b and 6.a. – are equivalent and should satisfy the same ratios for the required number of rounds.

Problem 7.a.

Let the plaintext M and the flag output Y_{D} be known. From the ratio (3.64) it follows:

$$H(Y_{\mathrm{DT}}Y_{\mathrm{D}}K|M) = [\text{ so } H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}KM) = 0] = \\ = H(Y_{\mathrm{D}}K|M) = H(K).$$

From where

$$H(K|Y_{\mathrm{D}}M) = H(K) - H(Y_{\mathrm{D}}|M).$$

At a fixed choice of a round substitution transformation from the key K entropy $H(K) = k \cdot \log\bigl(2^n!\bigr)$, and conditional entropy at performing m rounds doesn't exceed $H(Y_D|M) \leq m \cdot \log\bigl(\prod_{i=r}^{n-1} 2^i!\bigr)$. Therefore,

$$H(K|Y_{\mathrm{D}}M) \geq \log \frac{\bigl(2^n!\bigr)^k}{\bigl(\prod_{i=r}^{n-1} 2^i!\bigr)^m}.$$

If the key is chosen at random at each round, then

$$H(K|Y_{\mathrm{D}}M) \geq k \cdot \log\bigl(2^n!\bigr) + m \cdot \log k - m \cdot \log\Bigl(\prod_{i=r}^{n-1} 2^i!\Bigr). \quad (3.77)$$

Problem 7.b. Let the plaintext M and the flag output Y_{D} be known.

From the ratio (3.64) it follows:

$$H(Y_{\mathrm{DT}}|Y_{\mathrm{D}}M) = H(K|Y_{\mathrm{D}}M).$$

Therefore, the problems 7.a and 7.b have the same complexity.

3.6 Usage modes of two-channel systems

From the point 3.5.1 and 3.5.2 we can draw the following conclusions

- for security of the general scheme the length of the core output shouldn't be too small $|Y_{\mathrm{DT}}| \geq \alpha$;
- a pseudorandom choice of a substitution transformation from a key increases security of the scheme in case, when one of the outputs is unknown;
- using the general scheme is not secure, if an adversary cryptanalyst knows the flag output, the plaintext and the key.

The universal mechanism of harming allows creating new applications due to manipulating input and output data.

For example, the following modes of data transmission are possible at using the universal mechanism of harming:

2. The key K and the output Y_{DT} are transmitted via a secure channel, and the output Y_{D} – via an open channel.
3. The output Y_{DT} is transmitted via a secure channel, and the key K and the output Y_{D} – via an open channel.
4. The key K and the output Y_{D} are transmitted via a secure channel, and the output Y_{DT} – via an open channel.
5. The output Y_{D} is transmitted via a secure channel, and the key K and the output Y_{DT} – via an open channel.

Moreover, the mechanism of harming can be used for message authentication and in this case the following variants can be considered:

6. The key K and the output Y_{DT} are transmitted via a secure channel, and the output Y_{D} and a plaintext M – via an open channel.

7. The key K and the output Y_{D} are transmitted via a secure channel, and the output Y_{DT} and a plaintext M – via an open channel.

3.7 Statistical testing

3.7.1 Dependence criteria

One can very seldom build an analytical model for modern ciphers which would allow evaluating their security. Usually some criteria are defined and some tests are carried out for conformance evaluation of an algorithm to these criteria. For substitution-permutation networks usually some criteria are used known as dependence criteria. These criteria were formulated by different authors (see, [59, 56, 32, 33]). They are destined for evaluation security of S-Boxes .Here we shall state the definitions from [82], which were used during the testing of the AES finalists [75].

Let x – an n-digit binary string. We shall define though $x^{(i)},\ 1 \le i \le n$, – a binary string obtained from the string x by the inversion of the i-th bit.

The *dependence matrix* of the function $f : (GF(2))^n \to GF(2))^m$ is an $n \times m$ matrix A with elements a_{ij} equal to the number of inputs for which complementing the i-th input bit

results in a change of the j-th output bit, i.e.

$$a_{ij} = \#\{x \in (GF(2))^n \mid (f(x^{(i)}))_j \neq (f(x))_j\} \tag{3.78}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

The *distance matrix* of a function $f : (GF(2))^n \to GF(2))^m$ is $n \times (m+1)$ matrix B with elements b_{ij} equal to the number of inputs, for which complementing the i-th input bit results in a change of j output bits, i.e.

$$b_{ij} = \#\{x \in (GF(2))^n \mid w(f(x^{(i)}), f(x)) = j\} \tag{3.79}$$

for $i = 1, \ldots, n$ and $j = 0, \ldots, m$.

Obviously, at the input size $n > 30$ due to limits of memory resources it's not possible to compute matrices of dependence and distance for all possible inputs. Therefore, one usually considers a "suitable" number of randomly chosen inputs. The dependence and distance matrices are then defined as follows:

$$a_{ij} = \#\{x \in \mathcal{X} \mid (f(x^{(i)}))_j \neq (f(x))_j\} \tag{3.80}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, m$, and

$$b_{ij} = \#\{x \in \mathcal{X} \mid w(f(x^{(i)}), f(x)) = j\}, \tag{3.81}$$

where $\mathcal{X}$ is a "suitable" randomly chosen subset of $(GF(2))^n$.

Assume we've computed the dependence matrix A and the distance matrix B of a function $f : (GF(2))^n \to GF(2))^m$ for a set of inputs $\mathcal{X}$, where $\mathcal{X}$ is either $(GF(2))^n$, or a rather large random subset of $(GF(2))^n$.

The *degree of completeness* of a function f is defined as

$$d_c = 1 - \frac{\#\{(i,j) | a_{ij} = 0\}}{nm}. \tag{3.82}$$

The *degree of avalanche effect* of a function f is

$$d_a = 1 - \frac{1}{nm} \cdot \sum_{i=1}^{n} \left| \frac{2}{\#\mathcal{X}} \sum_{j=1}^{m} j \cdot b_{ij} - m \right|. \tag{3.83}$$

The *degree of strict avalanche criterion* of a function f is defined as

$$d_{sa} = 1 - \frac{1}{nm} \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \left| \frac{2a_{ij}}{\#\mathfrak{X}} - 1 \right|. \tag{3.84}$$

For the function f having good degrees of completeness, avalanche effect, and strict avalanche criterion, the following must be satisfied:

$$d_c = 1,\ d_a \approx 1,\ d_{sa} \approx 1.$$

3.7.2 Dependence criteria for substitution transformations with a variable length output

The expressions (3.82), (3.83) and (3.84) are cited for the mappings $f : \{0,1\}^n \to \{0,1\}^m$. An MV2-transformation is used as a substitution transformation in the general scheme. This transformation is a pair of mappings with images of various length $g : \{0,1\}^n \to \bigcup\limits_i \{0,1\}^i$. In general, output lengths don't coincide for such a transformation, i.e.:

$$g(x_1) = y_1 \in \{0,1\}^i,\ g(x_2) = y_2 \in \{0,1\}^j,\ i \neq j.$$

The Hamming distance is used in dependence criteria to measure the degree of difference of binary strings. Using dependence criteria for research of the general scheme of harming requires reconsidering definitions and formulae for computing distance and dependency matrixes. We defined its analogue for variable length mappings (see the definition 3.2 from i. 3.1.1).

During the test we shall use the following formulae to compute degrees of completeness, of avalanche and of strict avalanche:

$$b_{ij} = |\{x \in \mathfrak{X} \mid h(g(x^{(i)}), g(x)) = j\}|, \tag{3.85}$$

$$\overline{d}_a = 1 - \frac{1}{mn} \cdot \sum_{i=1}^{n} \left| \frac{2}{\#\mathfrak{X}} \sum_{j=1}^{\overline{m}_i} j \cdot b_{ij} - \overline{m}_i \right|, \tag{3.86}$$

$$\overline{d}_{sa} = 1 - \frac{1}{mn} \cdot \sum_{i=1}^{n} \sum_{j=1}^{\overline{m}_i} \left| \frac{2a_{ij}}{\#\mathfrak{X}} - 1 \right|, \tag{3.87}$$

where $\overline{m}_i = \max\{|g(x^{(i)})| \,:\, x \in X\}$ – the maximal output length at changing the i-th bit, and $m = \max\limits_i\{\overline{m}_i\}$.

In the expressions (3.86) and (3.87) instead of the maximal lengths of $\overline{m}_i$ output one can take the average output lengths:

$$\widetilde{m}_i = \frac{1}{\#\mathfrak{X}} \sum_{x \in \mathfrak{X}} |g(x^{(i)})|.$$

In this case the expressions (3.86) and (3.87) will look like:

$$\widetilde{d}_a = 1 - \frac{1}{mn} \cdot \sum_{i=1}^{n} \left| \frac{2}{\#\mathfrak{X}} \sum_{j=1}^{\widetilde{m}_i} j \cdot b_{ij} - \widetilde{m}_i \right|, \tag{3.88}$$

$$\widetilde{d}_{sa} = 1 - \frac{1}{mn} \cdot \sum_{i=1}^{n} \sum_{j=1}^{\widetilde{m}_i} \left| \frac{2a_{ij}}{\#\mathfrak{X}} - 1 \right|, \tag{3.89}$$

where $m = \max\limits_i\{\widetilde{m}_i\}$.

In Fig. 3.6 there are charts of results of testing the basic implementation of the MV2 encryption algorithm (see Application B) for correspondence to a strict avalanche criteria. One can see from the chart that values computed

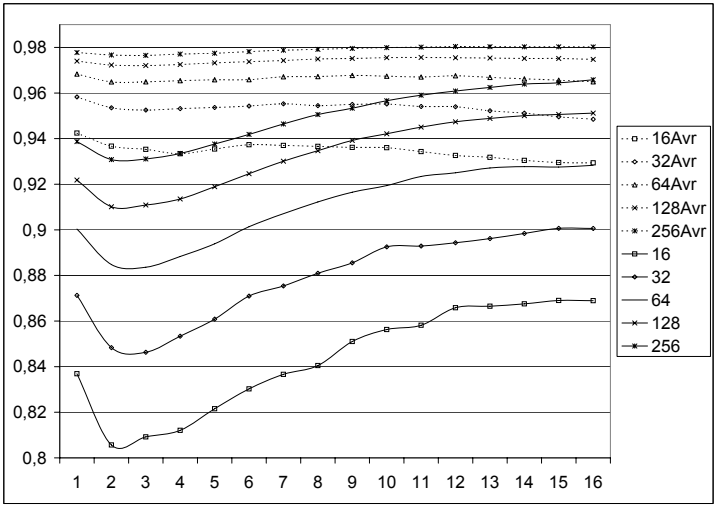

Fig. 3.6: *Dependence of the degree of a strict avalanche criterion* $\overline{d}_{sa}$ *and* $\widetilde{d}_{sa}$ *(Avr) on a number of rounds at different output lengths (16, 32, 64, 128 and 256 bytes)*

at using the average lengths according to the formulae (3.89) are greater than those computed according to the formulae (3.87) at using the maximal output lengths.

The expressions (3.86, 3.87) and (3.88, 3.89) reflect more precisely the specific character of the transformation, than (3.83) and (3.84), but still add some error to "tails" of strings of distance and dependence matrices.

It's connected to the fact, that in the basic expressions (3.83) and (3.84) the normalization coefficient $1/mn$ due to the mapping properties is used correctly. Therefore, the absolute values of the degrees d_a and d_{sa} reflect quality of cryptographic transformations. In case with variable length transformations the normalization coefficients used in formulae (3.86), (3.87), (3.88) and (3.89) are not exactly correct. For different classes of transformations at such normalizations

various hard errors are included, which make the degree value less. In the given case these criteria can serve for comparison of cryptographic transformations of the same class.

In appendix B the results of testing of one of the implementations of the general scheme for compliance with dependence criteria are given. The analysis of the results (see B.3) let choose permutation transformations, evaluate influence of whitening, and the value of a pseudorandom choice of substitution transformations. Besides, testing for compliance with dependence criteria confirmed the supposition about increase of the algorithm efficiency at increasing a plaintext length.

3.8 Summary

Thus, an MV2-transformation allows implementing the universal mechanism of harming. A special case of this scheme – a two-channel algorithm MV2 – possesses a number of distinctive features:

- a ciphertext consists of **two parts** – a core and flags;
- to restore a plaintext it's necessary to have all the three components – the key, core and flags;
- irrespective of the size and content of a plaintext the core can be made small enough (but not smaller than a value dependent on the implementation);
- the MV2 algorithm is pseudorandom; at repeated encryption of the same plaintext different pairs of cores and flags are obtained, that allows not to change keys at their long-term use;

- the MV2 algorithm allows paralleling an encryption process;
- the MV2 algorithm can be easily developed till the variant which provides message authentication.

On the basis of the two-channel MV2 algorithm features we have a possibility to develop fundamentally new multi-channel systems of data protection (see chapter 5 and [69], [70]).

Chapter 4
Protocols and multi-channel cryptography

4.1 Concept of protocol. Language of protocols

Definition 4.1 *A protocol is a distributed algorithm of interaction between two or more parts (people, machines) communicating with each other according to agreed message specifications with necessary synchronization and certain actions during worst-case situations.*

Definition 4.2 *A cryptographic protocol is a protocol which protects its participants from external attacks and dishonest participants of the protocol.*

If cryptographic systems deal with problems of transferring confidential information, then, the problems of information integrity, authentication, digital signature and non-tracking are the tasks of cryptographic primitive protocols. Different

compilations of these protocols give a group of applied protocols which allow solving some practical task in the area of electronic circulation of documents, bank payment, sale, electronic medicine, protection of intellectual property, protection of paper documents from forgery and so on.

As in majority of cases protocols are interactive it's advisable to write their scenario in some language. The phrase structure of this language contains the following operators:

<ID of a protocol party>;
<actions set by the protocol>;
<{ object of transmission}>;
< address of transmitted information>;
<logical conclusions>.

For example, Alice generates a random number r, encrypts it by an algorithm E with a key K and sends this result to Bob:

A: <ID of a protocol party>: A;
<his actions set by the protocol>: $r \in_R Z_p$;
<{ object of transmission }>: $\{Y = E(r, K)\}$;
< address of transmitted information >: $\rightarrow B$.

4.2 Authentication of a communication participant

Authentication is a process of proving identity of a person (computer) which got in contact with another person (computer). Usually it's a protocol between two or more parts. The central moment of any authentication is checking a part for *knowing a certain secret*. Authentication can be personal or mutual. In the first case one of the parts wants to get

something from another part; in the second case parts want to get something from each other or form a mutual secret, for example, encryption keys.

Any authentication protocol must go through checking for *correctness* and *reliability*. Correctness of an authentication protocol means its consistency and executing in a finite number of rounds. Reliability of an authentication protocol means impossibility of the third party to present himself as one of the participants of this protocol.

We shall consider four possible parts of the authentication protocol: Alice and Bob are two direct personages of the authentication protocol, a trusted part Trent, and a curious person Mallory. Here are the roles of these personages: Alice and Bob want to prove each other their belonging to the names, Trent is a trusted part of Alice and Bob, Mallory has a possibility of intercepting all Alice, Bob and Trent's messages and pretending he is one of them.

At such a situation Alice, Bob and Trent should have a secret which Mallory doesn't know. A type of this secret depends on a kind of an applied encryption system: it can be secret keys of a symmetric system, secret keys of the system with an open key, secret passwords. At that a certain technology of applying these secret data must be maintained. This technology would exclude Mallory's attempts to solve his falsification task. It must prevent Mallory from stealing some constants applied at authentication. This technology should use *randomness of authentication parameters generated immediately in the moment of authentication* . Here are logical general steps of this technology:

- Alice tells Trent she would like to get in touch with Bob;
- Trent tells Bob about it, and if Bob agrees, tells them a certain secret;

- having a mutual or personal secrets Alice and Bob generate a random parameter and prove each other that they *know of this secret*;
- if Mallory intercepts the whole secret he will be able to solve the falsification problem;
- if Mallory intercepts a part of the secret, he'll be able to solve his problem at certain conditions;
- if Mallory doesn't intercept the secret he won't be able to solve his problem.

For more reliability a secret can be valid during some time to slack resistance of Mallory who will be limited by this period.

Let's consider an authentication scheme of Schnorr [90].

In a one-channel variant Schnorr's algorithm works according to the following scheme to authenticate Alice: I. Prior operation.

1. Let p and q be prime numbers such that q divides $p-1$. Let then $g \in Z_p$ such that $g^q \equiv 1(\mathrm{mod}\, p)$, $g \neq 1$.
2. Let $k_{pr} = x \in_R \{1 \ldots, q-1\}$ be a secret key. Then $k_{pb} = g^{-x}(\mathrm{mod}\, p)$ is a public key.
3. Alice and Bob have correspondingly k_{Apr}, k_{Apb} and k_{Bpr}, k_{Bpb}. Trent guarantees safety and correctness of public keys.

II. Schnorr's algorithm.

1. A: A;
   $w \in_R \{1, \ldots, q-1\}$;
   $\{r = g^w(\mathrm{mod}\, p)\} \to B$.

2. B: B;
   $\{e \in_R \{0, ...2^t - 1\}\} \rightarrow A$, where t is a system parameter.

3. A: A;
   $s = (w + ek_{Apr}) (\mathrm{mod}\, q)$;
   $\{s\} \rightarrow B$.

4. B:
   $r' = g^s k_{Apb}^e (\mathrm{mod}\, p)$;
   $r' = r$?!
   The prove of Alice authenticity is accepted at an equality, and it is denied at an inequality.

Note that Alice showed she knew her secret parameter k_{Apr} without telling it to Bob. As only Alice knows this parameter it is supposed that it was she who used it[1].

Let's consider a two-channel variant of Schnorr's scheme [90]. A two-channel variant should provide better security of an authentication process than the one-channel one due to the following reasons:

- each party has at least two secrets: a mutual and a personal one;
- a combination possible when Trent doesn't know one of the secrets, e.g. a mutual secret in form of keys of a symmetric encryption system;
- Mallory doesn't have any possibility of knowing even accidental authentication parameters;
- mutual authentication is always provided.

[1]It's not exactly true. We can say with confidence that somebody who new Alice's secret pushed the buttons.

An authentication protocol using a two-channel symmetric-asymmetric encryption system with a public key can be implemented according to the MV2 - RSA scheme (see 2.3). We shall mark the scheme of Fig. 2.4 as shown in Fig. 4.1.

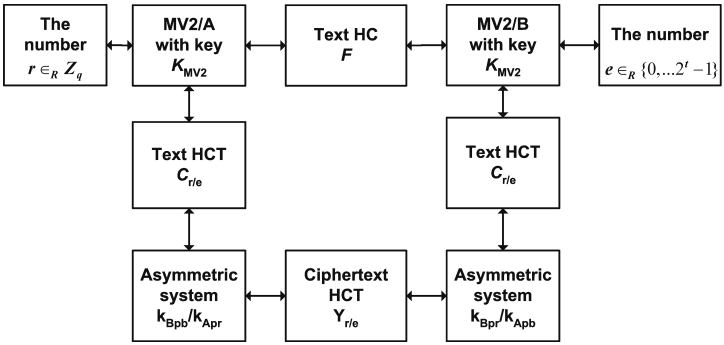

Fig. 4.1: *Using the MV2-RSA system for mutual authentication*

Here is the protocol:

1. A: A;
   $r \in_R \{0, 1, \dots 2^t - 1\}$;
   $C_r = E_1(r, K_{MV2})$;
   $F_r = E_2(r, K_{MV2})$;
   $Y_r = E_3(C_r, k_{Apr})$;
   $\{Y_r, F_r\} \to B$.

2. B: B;
   $C_r = E_4(Y_r, k_{Apb})$;
   $r = E_{12}^{-1}(F_r, C_r, K_{MV2})$;
   $e \in_R \{0, ...2^t - 1\}$;
   $C_e = E_1(e, K_{MV2})$;
   $F_e = E_2(e, K_{MV2})$;
   $Y_e = E_3(C_e, k_{Bpr})$;
   $\{Y_e, F_e\} \to A$.

3. A: A;
   $C_e = E_4(Y_e, k_{Bpb})$;
   $e = E_{12}^{-1}(F_e, C_e, K_{MV2})$;
   $s = (e + r)(\mathrm{mod}\, 2^t)$;
   $C_s = E_1(s, K_{MV2})$;
   $F_s = E_2(s, K_{MV2})$;
   $Y_s = E_3(C_s, k_{Apr})$;
   $\{Y_s, F_s\} \to B$.

4. B:
   $C_s = E_4(Y_s, k_{Apb})$;
   $s' = E_{12}^{-1}(F_s, C_s, K_{MV2})$;
   $e' = (s' - r)(\mathrm{mod}\, 2^t)$;
   $e' = e$?!

In the considered scheme Mallory doesn't have even open information about transmitted random numbers. At that a total key field is $K_{\sum} = K_{MV2} \times k_{pr}$.

4.3 Message authentication (digital signature)

The process of *message authentication* allows obtaining information saying that the given message was sent from an authenticated person and received in a non-corrupted form. *The protocol of digital signature* solves this problem. All protocols of a digital signature are non-interactive, i.e. executed in one step.

We shall take Schnorr's protocol of a digital signature as an example [90]. Here is the protocol:

I. Prior operation.

1. Let p and q be prime numbers such that q divides $p-1$. Let further $g \in Z_p$ be such that $g^q \equiv 1(\mathrm{mod}\, p)$, $g \neq 1$.

2. Let $k_{pr} = x \in_R \{1, \ldots, q-1\}$ be a private key. Then $k_{pb} = g^{-x}(\mathrm{mod}\, p)$ is a public key.

3. Alice and Bob have correspondingly k_{Apr}, k_{Apb} and k_{Bpr}, k_{Bpb}. Trent guarantees safety and correctness of public keys.

4. Alice and Bob have an algorithm of forming a hash-function $h(r, M)$, where M is a message being signed, r is some parameter.

II. Schnorr's signature algorithm.

5 A:A;
$w \in_R \{1, \ldots, q-1\}$;
$r = g^w(\mathrm{mod}\, p)$;
$e = h(r, M)$;
$s = (w + ek_{Apr})(\mathrm{mod}\, q)$;
$\{e, s, M\} \to B$.

6 B:
$r' = g^s k_{Apb}^e(\mathrm{mod}\, p)$;
$e = h(r', M)$?!
Alice's signature is accepted at a hash-function equality and it is denied at an inequality

We implement this protocol using the material listed in 2.3. Here we shall formulate requirements for such protocols:

- an encryption round with a simultaneous digital signature should be carried out according to the following algorithm:

$$Y = E(M, K_{MV2}, k_{rp1}); \tag{4.1}$$

- a decryption round with simultaneous checking of the digital signature should be carried out according to the following algorithm:

$$M = E^{-1}(Y, K_{MV2}, k_{pb1}). \tag{4.2}$$

Below for comparison there's a hybrid scheme of Fiat and Shamir's system (Fig. 4.2) [49], using a symmetric encryption algorithm with a secret key and an asymmetric algorithm with an open key to transmit encrypted messages with an digital signature.

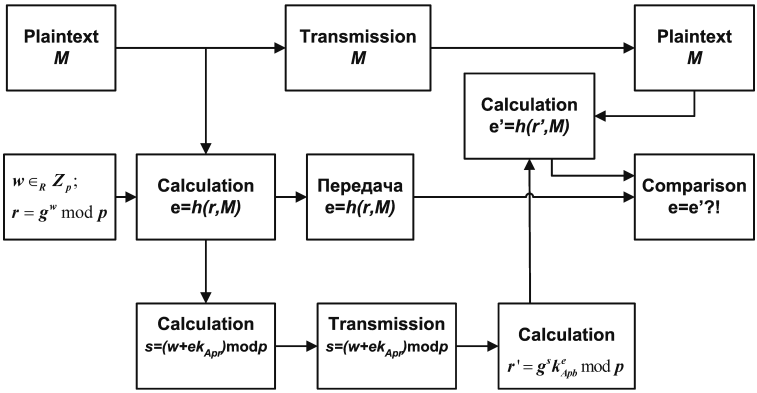

Fig. 4.2: *Schnorr's digital signature scheme*

In the patent [70] there's a structure of the real symmetric-asymmetric system with an open key is suggested which satisfies the requirements (4.1) – (4.2). An encryption mode of a digital signature is shown in Fig. 4.3. A key field in the encryption mode for this system equals $K \times K_{pr2}$ in comparison with the field K of the system presented in Fig. 4.2, and a key field of the mode of the digital signature $K \times K_{pr1}$ in comparison with the field K_{pr1} of the system presented in Fig. 4.2.

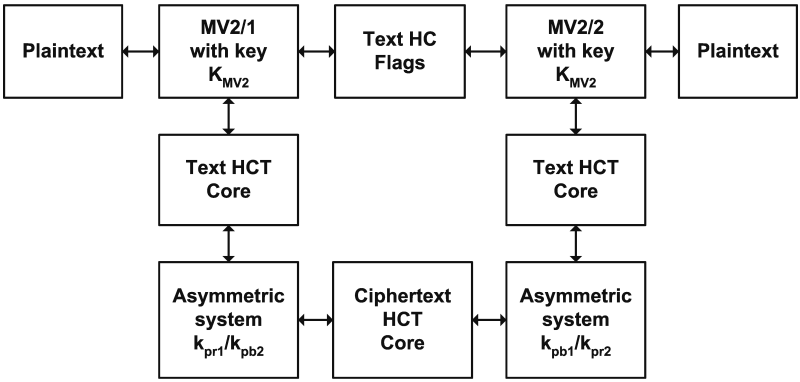

Fig. 4.3: *Real symmetric-asymmetric system with an open key in encryption/ digital signature mode*

The protocol of a digital signature and its checking for the presented scheme in Fig. 4.3 looks in the following way:

A: A; $C = E_1(M, K_{MV2})$;
$F = E_2(M, K_{MV2})$;
$Y_{AC} = E_3(C, k_{Apr})$;
$\{F, Y_{AC}\} \to B$.

B: B; $C = E_4(Y_{AC}, k_{Apb})$;
$M = E_{12}^{-1}(F, C, K_{MV2})$;
[Y_{AC} – Alice's signature].

Protocol of reciprocal signature with checking:

A: A; $C = E_1(M, K_{MV2})$;
$F = E_2(M, K_{MV2})$;
$Y_{AC} = E_3(C, k_{Bpr})$;
$\{F, Y_{AC}\} \to B$
[Y_{AC} – is an Alice's signature.]

B: B; $C = E_4(Y_{AC}, k_{Apb})$;
$M = E_{12}^{-1}(F, C, K_{MV2})$;
$Y_{BC} = E_3(C, k_{Apr})$;
$\{F, Y_{BC}\} \to A$;
[Y_{BC} – is a Bob's signature.].

A: $C' = E_4(Y_{BC}, k_{Bpb})$;
$M' = E_{12}^{-1}(F, C', K_{MV2})$;
$M' = M$?!

If a number of participants is more than two, then one of them, for instance under number 1, takes the distribution upon himself and generates a generic key K_{MV2} :

1: 1; $C = E_1(M, K_{MV2})$;
$F = E_2(M, K_{MV2})$;
$Y_C^{(1)} = E_3(C, k_{Apr})$;
$\{F, Y_C^{(1)}\} \to 2, 3...n$;
.....

i: i; $C = E_4(Y_C^{(1)}, k_{Apb})$;
$M = E_{12}^{-1}(F, C, K_{MV2})$;
$Y_C^{(i)} = E_3(C, k_{pr}^{(i)})$;
$\{Y_C^{(i)}\} \to 1, 2...n$;
[$Y_C^{(i)}$ – is a signature of a i-th participant]

i: $C_i' = E_4(Y_C^{(i)}, k_{ipb})$;
$M_i' = E_{12}^{-1}(F, C_i', K_{MV2})$;
$M_i' = M$?!;
[$Y_C^{(i)}$ – is a signature of a i-th participant]
$i = 1, 2...n$;

MAC – Message Authentication Code depends on the key one-way hash-function which is imbedded in a message. If an obtained hash-function is meant for checking message authentication, the MAC allows only the person who knows the key doing it. In multichannel cryptography the core (a harmed text) plays a part of a MAC. In this case forming

such a function dependent on the key and its checking happen automatically in a message decryption mode.

4.4 Non-accountability. Electronic money

An important moment during paying with banknotes is a property of non-accountability: practically nobody can track that banknotes obtained by a certain person were spent on specific purposes. If it were true it would mean total spying on life activity of every member of society. In case of financial transactions this principle isn't carried off completely: a bank knows client's expenses, and the payment address can be defined if the payment is made by card. But if a client gets cash from the card, anonymity is kept only partially, as the bank knows client's expenses only, but doesn't know the addressee and purpose of spending.

In case *electronic money* in form of some data files is used for payment, the issue of owner's anonymity and his expenses must be kept strictly, otherwise, nobody will use such money. Electronic money must possess the following properties:

- money's owner doesn't have a possibility to make copies of electronic banknotes;
- a bank can't ascertain the owner of the money during payment;
- a bank has a possibility to define electronic banknotes that were already once spent by a client.

It was Chaum who first solved all these problems. [38]. He created a so called *blind signature* [39]. Participants in the

payments system are a bank, a buyer and a seller.

The bank generates two large prime numbers p and q and keeps them secret, but publishes their product $N = pq$. Let the used RSA system have two numbers: $k_{pb} = e$ – an open key and $k_{pr} = d$ – a secret key, at that $ed = 1(\mathrm{mod}\,\phi(N))$. It's possible to sign some electronic document a :

$$s = a^d(\mathrm{mod}\,N).$$

with the help of RSA.

In its turn it's possible to read the signed document:

$$s^e = a^{de}(\mathrm{mod}\,N) = a(\mathrm{mod}\,N).$$

Besides, the bank publishes a one-way function $f : Z_N \to Z_N$. At that, each key pair is used by the bank for electronic banknotes of the same type. For example, during operations with USD the bank should have 7 key pairs for the banknotes of the following tenor \$1, \$2, \$5, \$10, \$20, \$50 and \$100.

Suppose the buyer wants to buy \$100 from the bank. He generates a random number $n \in Z_N$ and computes the value $f(n)$. Further he asks the bank to sign the computed value $f(n)$:

$$s = f^d(n)(\mathrm{mod}\,N).$$

with the key d.

The bank is ready to do it by previously withdrawing \$100 from the buyer's account. But in this case the bank knows that the banknote

$$(n, s) \;=\; (\; n,\; f^d(n)(\mathrm{mod}\,N)\;)$$

was sold to this very buyer, and the principle of non-accountability will be broken. Then the buyer acts according to the Chaum's algorithm: he doesn't send the value $f(n)$, to

the bank, but some another number – $f(n)r^e$, where $r \in Z_N$ is one more random number.

The bank signs the number

$$f(n)r^e : \; s' = f^d(n)r^{ed}(\mathrm{mod}\, N) \; = \; f^d(n)r(\mathrm{mod}\, N)$$

and sends it to the buyer. The buyer gets rid of the random number r known only to him and receives a banknote

$$(n, s) = (n, \frac{s'}{r}) = (n, f^d(n)(\mathrm{mod}\, N)).$$

signed by the bank.

Note that the bank signed the number

$$s' \; = \; f(n)^d r^{ed}(\mathrm{mod}\, N) \; = \; f^d(n)r(\mathrm{mod}\, N)$$

and it doesn't know the number $s = f^d(n)(\mathrm{mod}\, N)$, but its signature is under this number which can't be identified with the buyer.

During the purchase the buyer gives the banknote

$$(n, s) = (n, f^d(n)(\mathrm{mod}\, N)).$$

to the bank.

The seller sends this banknote to the bank to make sure the banknote wasn't spent before as a dishonest buyer can try to make copies of the same banknote. If the bank confirms that this banknote is being spent for the first time, the seller gives the goods and the bank enters \$100 to the seller's account.

Let's consider protective mechanisms of all the participants who uses electronic money.

Protection of the bank is determined by security of the RSA system's electronic signature. In particular, a one-

way function $f(n)$, which doesn't allow synthesizing a third signature if the two others are known [1].

Buyer's anonymity is based on buying from the bank the random number

$$s' = f^d(n)r^{ed}(\mathrm{mod}\, N) = f^d(n)r(\mathrm{mod}\, N),$$

which is not further identified with the owner of the banknote $(n, s) = (n, f^d(n)(\mathrm{mod}\, N))$.

Security of the seller is based on the fact that it's impossible for the buyer to make copies of real banknotes.

Let's consider Chaum's idea with blind signature using the MV2 algorithm. The bank randomly generates an abracadabra A and calls it as \$100. Then it encrypts it by the MV2 algorithm with a key and gets two data files: C_{0A} and F_A. The file C_{0A} is subject to sale, and the file F_A is kept in the bank. The latter also publishes its open key e of an asymmetric encryption system.

The buyer asks the bank to sell him a digital banknote of \$100. The bank sends to the buyer the file C_{0A}. The buyer generates a random number r and sends the file $(C_{0A}r^e)(\mathrm{mod}\, N)$ to the bank. The bank signs this file:

$$(C_{0A}r^e)^d(\mathrm{mod}\, N) = C_{0A}^d r(\mathrm{mod}\, N),$$

where d is a secret key of the bank, and sends it to the buyer. The buyer easily gets the multiplier r known to him and receives the file $C_{0A}^d(\mathrm{mod}\, N)$. signed by the bank.

[1]Indeed, let $s_1 = (n_1, f^d(n_1) \bmod N)$ and $s_2 = (n_2, f^d(n_2)(\mathrm{mod}\, N))$. Then

$$s_1 s_2 = (n_3,\ f^d(n_1)f^d(n_2)(\mathrm{mod}\, N)) = (n_3,\ (f(n_1)f(n_2))^d(\mathrm{mod}\, N)).$$

But this value can't be defined because the function is a one-way function.

The buyer gives the file $C_{0A}^{d}(\mathrm{mod}\, N)$, to the seller who sends it to the bank. The latter checks validity of the file

$$E^{-1}(C_{0A}^{de}(\mathrm{mod}\, N), F_A, K_{MV2}) = E^{-1}(C_{0A}, F_A, K_{MV2}) = A$$

and enters \$100 to the seller's account. At that the kept file F_A is destroyed. It prevents the owner of the banknote from making copies of it.

Note that there's no need to have the one-way function $f(n)$ to counteract the multiplicative property of the RSA system in this scheme, as the buyer cannot make a false banknote for any generated core without knowing the key K_{MV2} and the file F_A. The core C_{0A} plays the role of this function.

4.5 The problem of key deposition and the MV2 algorithm

Cryptography is required to protect law-abiding users from violators, and therefore it must use strong encryption methods. But nobody can guarantee that attackers won't use cryptography in their criminal goals. That is why, on one hand, cryptography should be under a governmental control, and, on the other hand, it should ensure rights of personal immunity and protect a person against shadowing from the side of a government. As for this there are a lot of contradictory opinions. Some specialists consider governmental control over private correspondence inadmissible and suspect the government of perlustration of commercial cryptosystems, other specialists think that a governmental control is a must. In the latter case a juridical mechanism (legal

resolutions, governmental regulations) is required. This juridical mechanism is activated in some specific situations connected primarily with the threat of terrorism, with criminality control and other society and government threats. These contradictory requirements are usually solved with the help of deposition of secret keys of cryptographic systems. Silvio Mikali actively studied these questions for secret key cryptography (American standard EES [73], and also for a public key cryptography [65, 66, 67, 68]. Mikali called all systems using the idea of deposition legal cryptosystems. It's obvious that the rest of the systems are illegal. Here are the main ideas of these methods.

Every user of a legal cryptosystem has his ID number and a secret key. This key is divided into two or more (u) parts, each of them is stored in corresponding mediating organizations. They can, or probably, must not know about existence of each other. Then it works according to the following protocol, where $K^{(SAB)}$ is a secret key of users A and B; $K^{(iAB)}$ is a i-th part of the secret key calculated by the method of secret splitting; $K^{(s)}$ is a session key; O_i is a i-th mediating organization; Gov is a governmental body:

$A : A;$
$K^{(SAB)} = \bigoplus_{i=1}^{u} K^{(iAB)};$
$K^{(s)} \in_R \mathfrak{Z}$, $\mathfrak{Z}$ – a set of numbers;
$K = E_S(K^{(s)}, K^{(SAB)});$
$\{K^{(iAB)}\} \to O_i,\ i = 1, \ldots, u;$
$\{K\} \to Gov;$
$\{K^{(s)}\} \to B;$
$Y = E_M(M, K^{(s)});$
$\{Y\} \to B;$

$B : B;$
$B = E_M^{-1}(Y, K^{(s)});$

Gov;
$\{Request\} \to O_i,\ i = 1, \ldots, u$;

$O_i : O_i$;
$\{K^{(iAB)}\} \to Gov$;

Gov;
$K^{(SAB)} = \bigoplus_{i=1}^{u} K^{(iAB)}$;
$K^{(s)} = E_S^{-1}(K^{(SAB)}, K)$;
$M = E_M^{-1}(Y, K^{(s)})$;

For cryptosystems with an open key Mikali method has a slightly different protocol. Let p and q be prime numbers, such that q divides $p-1$ and $g \in Z_p$ such that $g^q \equiv 1 (\mathrm{mod}\, p)$, $g \neq 1$. Let $k_{pr} = d \in_R \{1 \ldots, q-1\}$ be a secret key A. Then $k_{pb} = g^{-d} (\mathrm{mod}\, p)$ is an open key A. Designate KDC – a center of open key deposition and $d(O_i)$ – an open key of a i-th trusted mediating organization. Mikali protocol for systems with an open key looks like:

$A : A$;
$k_{pr} = \left(\sum_{i=1}^{u} d_i \right) (\mathrm{mod}\, p-1),\ d_i \in Z_p$;
$e_i = g^{d_i} (\mathrm{mod}\, p),\ i = 1, \ldots, u$;
$\{k_{pb}\} \to KDC$;
$\{d_i, e_i\} \to O_i,\ i = 1, \ldots, u$;

O_i;
$e_i = g^{d_i} (\mathrm{mod}\, p)$!?
$e_i' = E(e_i, d(O_i))$;
$\{e_i'\} \to KDC$;

KDC;
$k_{pb} = \left(\prod_{i=1}^{u} e_i \right) (\mathrm{mod}\, p)$!?

In Mikali technology, as we can see from these protocols, the government organization *Gov* restores session keys and reads correspondence of the users A and B.

Multi-channel cryptography can successfully solve this problem by other means. The first method lies in concealing the core for everybody except corresponding controlling units. In this case it plays the role of a very long additional key. The second method is manipulating parts of the flags. Like in Mikali protocols it's necessary to use a juridical mechanism here. We shall consider these ways of problem solving.

First of all we shall consider participants of these protocols: Alice (A) and Bob (B), a government organization *Gov* and a hacker Mallory. A problem definition: on some assumptions *Gov* can read secret correspondence of Alice and Bob, but it should be unreadable for Mallory. We shall proceed from the real circumstances: the government organization *Gov* possesses a bigger or the same computational resource as the hacker Mallory.

The protocol of a concealed core.

$A; A;$
$K_{MV2}^{(s)} \in_R \mathcal{Z}_1;$
$K^{(s)} \in_R \mathcal{Z}_2;$
$K_1 = E_K(K_{MV2}^{(s)});$
$K_2 = E(K^{(s)});$
$\{K_1, K_2\} \to B;$
$C = E_1(M, K_{MV2}^{(s)});$
$F = E_2(M, K_{MV2}^{(s)});$
$C' = C \oplus K(s);$
$K = E_S(K^{(s)});$
$\{K\} \to Gov;$[1]

[1] According to a court decision or a government regulation in a certain period.

$\{C', F\} \to B;$

$B;$
$K_{MV2}^{(s)} = E_K^{-1}(K_1);$
$K^{(s)} = E_K^{-1}(K_2);$
$C = C' \oplus K^{(s)};$
$M = E_{12}^{-1}(C, F, K_{MV2}^{(s)});$

$Gov;$
$K^{(s)} = E_S^{-1}(K);$
$C = C' \oplus K^{(s)};$ [1]
$M!?$

Here $K^{(s)}$ is a session key for core encryption which is chosen from a large set $\mathfrak{Z}_2$ and transmitted to *Gov* via a secure channel. The keys $K_{MV2}^{(s)}$ and $K^{(s)}$ are transmitted to B via a secure channel as well. At this it is supposed that the cryptosystem is built in such a way that the key $K_{MV2}^{(s)}$ is chosen from the bounded set $\mathfrak{Z}_1$, therefore it can be discovered by the powerful computational resources of *Gov*. Then *Gov* can restore a plaintext: $M = E_{12}^{-1}(C, F, K_{MV2}^{(s)})$. Mallory can't restore the plaintext even if he has the same resources for the search of $K_{MV2}^{(s)}$, because he doesn't know the long key $K^{(s)}$, and, consequently, the core C.

One can manipulate the components $F = \{F_i\}$ in the same manner at every step in case he needs to have several go-betweens.

4.6 Summary

A data-pump plays the role of a peculiar hash-function in multichannel cryptography. Therefore, in many protocols where this primitive is required there's no need to specially form it – it's enough to use a short data-pump. In combination

with asymmetric systems with an open key the core is a successful object for encrypting or forming an electronic signature. At that together with the flags and keys K_{MV2} the core in contrast to a hash-function provides zero collision probability (see 1.6). Therefore the core signature at data restoring is an incontestable signature for the set values F and K_{MV2}.

Chapter 5
Mass technologies and multi-channel cryptography

5.1 Concept of mass technologies

There are such market technologies in the world in which a great number of people is involved, sometimes practically the whole country's population. These technologies relate to trade, multimedia services, computer usage, medicine, transport and so on. We shall further call technologies of such a type as mass technologies.

Due to a big number of users mass technologies are provided with a big flow of money; and serious financial investments in form of advertisement, overcoming monopolism of old technologies of big companies, psychological barriers of population are required to promote new technologies to the market. Therefore introduction of such technologies into the modern market of goods and services is in competence of financially powerful companies or government.

Mass technologies have the peculiarity of infringing on

interests of a great number of users and answer urgent questions of the society on the whole or necessary needs of the market. These technologies can be connected with political problems of the society, for example, struggling against political terrorism which keeps in awe the whole population of a country, a computer terrorism which deactivates control systems of important establishments or damages greatly economics on the governmental level. We can say that human community of the end of the 20th – beginning of the 21st century faced the necessity of solving problems of mass technologies on a global scale. First of all it touches upon security problems. Practically the whole population of a country needs them in different spheres of this term: personal security, security of relatives, health, property, intellectual property, business and so on. We shall be interested in the questions connected with the following: security of intellectual property, security of information transmission and storage, some mass technologies used in trade, computer technologies, mass technologies in multimedia and a number of others.

Considerable money flows running between mass technologies make criminal elements violate these technologies, palming off substitutes of services and goods as originals. Today we can speak about a world of forgery existing in such spheres as pharmaceutics, multimedia production, spirits and tobacco goods, in brands of famous companies and so on.

Mass technologies require some specific protective mechanism which would provide, on one hand, security of a technology's owner, and, on the other hand security of this technology's users. Therefore, we can speak about two security parameters: a security parameter of the technology's owner k_0 and a security parameter of this technology's user k_u. The security parameter k_0 with certain probability guarantees the owner of the technology protection from non-authorized

attempts of its using by a third person, and the security parameter k_u with certain probability guarantees the user the quality of this technology's service. For example, a pharmaceutical company producing some medicine has a protective mechanism which guarantees its owner protection from possible forgeries of this medicine. In their turn, users have some mechanism for determining authenticity of the bought medicine. Very often these mechanisms intersect, though they can also have independent components. Attempts to complicate these mechanisms at reproduction of distinctive marks of the real goods and services were not crowned with success.

Nowadays there's no such a universal protective mechanism, though common requirements can be outlined. First of all, protective mechanisms must contain several components: legal, organizational, technical and informational. Up to now a part of these components, in particular legal and organizational, is not realized in many technologies, though it's possible to do that. A technical component doesn't protect safely in majority of cases. Nowadays' poligraphy, special materials, physical devices, computers are not only available for producers, but for criminal elements as well. Very often a forgery can be detected with the help of special devices in special laboratories only. Secondly, an informational component must be principally authentic or forged with a high level of costs. This very new informational component promises the possibility of obtaining such a universal protective mechanism. In some cases traditional cryptographic technologies can be used for this component. But traditional cryptography doesn't have high enough degrees of freedom at projecting protective mechanisms in mass technologies. It can provide only confidentiality of a data stream together with the known authentication and integrity

protocols, a protocol of an electronic signature and a number of other protocols.

At the same time mass technologies have a character distributed in dimension and time and require some controlled parameters for their functioning. Here we shall try to build a model of mass technology.

Mass technology is a process of goods production and sale or rendering of services. An industrial process (technology) can be concealed, while selling is usually a public process. The owner of the technology tries to place identifying marks in form of labels, bar-codes or other marks on his goods with the help of hardware. An attacker, usually using a cheaper technology which doesn't correspond to the required parameters, produces similar goods and places the same identifying marks. In outward appearance is doesn't differ from the real one. This process can formally be written as

$$V = D_0(\overrightarrow{A}, S_0);$$

$$V = D_C(\overrightarrow{B}, S_C),$$

where V is an appearance of the goods, $D_0(\overrightarrow{A}, S_0)$ is a real technology with the vector of goods parameters $\overrightarrow{A}$; $D_C(\overrightarrow{B}, S_C)$ – technology of forgery with the vector of goods parameters $\overrightarrow{B}$; S_0, S_C – accordingly the price of goods production (usually $S_0 > S_C$).

Protecting from forgery, the producer marks appearance of the goods: $V_C = D_0(\overrightarrow{A}, S_0)$. The attacker can do the same: $V_C = D_C(\overrightarrow{B}, S_C)$ – and the problem is again in its initial state, because the attacker easily makes copies of the real goods in mass quantity using his cheap technology. It's evident that the key for solving this problem lies in making the attacker use such a technology that would be expensive at individual

forging of every item and much more expensive at forging big consignments. At that the following equation must be maintained $S_0 < S_C$, then, the process of forging looses sense.

Moreover, the buyer should be directed towards appearance of goods only. He should get an authentication certificate for this item *directly from the producer*, who can *distantly* "recognize" his production. Thus, the scheme of mass technology protection is lead to the following:

1. The producer marks each product (service) with an individual sign (parameter k_0).

2. The sign must be impossible to copy (or possible to copy but with unreasonably high expenses).

3. At selling the producer must be able to "recognize " his sign and inform the buyer about that.

4. As there's a mediator (the seller) between the buyer and the producer, certain measures should be taken to protect the buyer from a dishonest seller (parameter k_u).

5. There's a number of organization measures which allow uncovering dishonest actions of the seller.

The security parameter k_0 protects the producers (the owner of the technology). It means that this parameter is unavailable for the attacker. The security parameter k_u which is unavailable for the seller protects the goods and the buyer.

Implementation of such a protective technology became possible thanks to multichannel cryptography which has necessary degrees of freedom for projecting different practical applications.

The MV2 algorithm allows to get several ciphertexts of various length for a plaintext M depending on the set steps

of the algorithm m:

$$Y_{\mathrm{DT}}^{(i)} = E_1(Y_{\mathrm{DT}}^{(i-1)}, K_i),$$
$$Y_{\mathrm{D}}^{(i)} = E_2(Y_{\mathrm{DT}}^{(i-1)}, K_i),$$

$i = 1, \ldots, m$ and $Y_{\mathrm{DT}}^{(0)} = M$, at that expected lengths of these texts are correspondingly equal (see 3.2.4):

$$|Y_{\mathrm{DT}}^{(i)}| \approx \mathrm{K_c}^{\,i} \cdot |M|,$$
$$|Y_{\mathrm{D}}^{(i)}| \approx \mathrm{K_f} \cdot \mathrm{K_c}^{\,i-1} \cdot |M|,$$

also the normalization takes place

$$|Y_{\mathrm{DT}}^{(i)}| + \sum_{i=1}^{m} |Y_{\mathrm{D}}^{(i)}| \approx |M|.$$

In the Table 5.1 there are relative lengths of these ciphertexts depending on the number of rounds of the algorithm at using the MV2 algorithm with the parameter $n = 8$.

Tabl. 5.1: *Relative lengths of the ciphertexts depending on the executed number of rounds*

i :	1	2	3	4	5
$\frac{\lvert Y_{\mathrm{DT}}^{(i)}\rvert}{\lvert M\rvert}$	0,758	0,574	0,435	0,330	0,250
$\frac{\lvert Y_{\mathrm{D}}^{(i)}\rvert}{\lvert M\rvert}$	0,242	0,184	0,139	0,105	0,080

i :	6	7	8	9	10
$\frac{\lvert Y_{\mathrm{DT}}^{(i)}\rvert}{\lvert M\rvert}$	0,190	0,144	0,109	0,083	0,063
$\frac{\lvert Y_{\mathrm{D}}^{(i)}\rvert}{\lvert M\rvert}$	0,060	0,046	0,035	0,026	0,020

For m steps of the algorithm we have $m + 1$ informational channel: the channel $Y_{\mathrm{DT}}^{(m)}$ and m channels $Y_{\mathrm{D}}^{(i)}$. Different

combinations of this information by channels allow to obtain several ciphertexts of various length:

- one-channel system: combining of $Y_{\mathrm{DT}}^{(m)}$ and all $Y_{\mathrm{D}}^{(i)}$ into one informational channel;
- two-channel system: the channel $Y_{\mathrm{DT}}^{(m)}$ and the common channel for all $Y_{\mathrm{D}}^{(i)}$ or combining $Y_{\mathrm{DT}}^{(m)}$ and a part of $Y_{\mathrm{D}}^{(i)}$ into one channel and the remained part of $Y_{\mathrm{D}}^{(i)}$ in another channel;
- three-channel system: two channels of the previous case and the third channel as a part of $Y_{\mathrm{D}}^{(i)}$;
- and so on.

Thus, multi-channel cryptography allows to present a plaintext as several ciphertexts, each of them or an incomplete set of ciphertexts cannot be decrypted into a plaintext. A number and length of these harmed texts are manageable and depend on a specific application. There are three common enough models of building systems using multichannel cryptography:

1. Systems where a part of ciphertexts of a small length is encrypted by a perfect secrecy system or is steganographically concealed in a conscious text container. In this case such a system has an additional long key of the unavailable (concealed) part of a message.
2. Systems where a part of information is sent to the addressee (for instance, for storage), and the other part is not sent at all. In this case an information keeper cannot use or send the information being stored to someone else.

3. Systems where information is spatially separated due to using other transmission channels.

We shall consider different variants of this protective technology in some applications which are stated in this chapter.

5.2 Telecommunications

Today an electronic mail is the most popular means of communication among millions of people. But it has at least two fundamental disadvantages. The first one is connected with the fact that your correspondence can be read by both the provider and a third person (even if it is encrypted by commercial ciphers, it is the matter of the decryption price), the second one – you can receive messages from unwanted people.

Besides, there are mail systems which:

- allow the possibility of interception and substitution of the transmitted information;
- allow the possibility of substituting the sender's address;
- don't react to a spam, and as a consequence are subject to virus spreading through a spam;
- require using additional encryption means and very often presence of corresponding training and knowledge that majority of the users doesn't possess;
- open for erroneous sending of unencrypted mail.

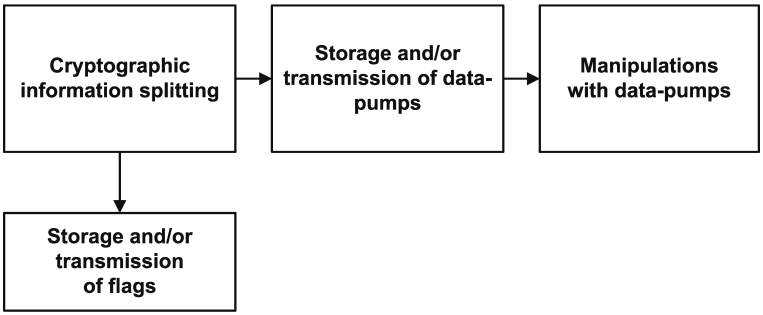

Fig. 5.1: *A structure of the SeMail system based on the MV2 algorithm*

One of the solutions of email security in the simplest case is a two-channel email based on the MV2 (Fig. 5.1).

Users A and B have at their disposal the MV2 algorithm with the key K_{MV2}, which allows obtaining two harmed ciphertexts – a core's ciphertext and a flags' ciphertext. A harmed ciphertext of the core is sent directly via a mail service, and the additionally encrypted ciphertext of the core of a small size is sent via the other channel – the system's server. To solve cryptographic problems (message authentication, sender identification, integrity of the transmitted message, and access of an authorized user only to the correspondence) additional encryption of the core's ciphertext is used.

This additional encryption can be carried out by both a symmetric system, for example, AES (at that the provider himself gives keys to every user $K_{AP\ AES}$, $K_{BP\ AES}$), and an asymmetric encryption system, for example, RSA with the key k_{Ppb}. In the latter case the MV2-RSA. system can be used. This combination structure of the type "star" became possible for thousands of users thanks to small sizes of the core's ciphertext only. Otherwise the system would be inoperative.

The peculiarity of this system is that it guarantees

protection of transmitted messages from non-authorized reading and conscious corruption by other persons including the provider, integrity control of transmitted data, a univocal identification of the sender and message authentication.

The system's provider is situated behind the firewall which blocks all incoming connections, therefore it sets connection with the server according to the set schedule and executes the necessary processing of the data being accumulated there. All the system's users who need some data processing by the server first form a request for data processing and send it to the server to their box. When the provider finishes processing, he sends the result to the same box, after that a user can take it. Thus, safety and inviolability of the data stored in the server is provided.

The system is protected by the patents [20] and [19].

5.3 Information storage

Information can be stored in the place of its use (locally), remotely or in a combined way. In all the cases it's necessary to do backup and parallel storage due to possible information loss. First of all we shall be interested in storage of confidential information. The owner of such information is confronted with the dilemma: should I keep the information personally, or should I entrust it to a special depository? In the first case it can happen that he doesn't possess the necessary conditions for secure storage, and in the second case he is not sure that the depository won't use his information even if it's encrypted. In the context of this book this requirement becomes clear: there's everything in the ciphertext, and if it's worth breaking it, it will be broken. Moreover encrypted information can be anonymously sold by the depository, and a legitimate user will

never know about that.

As the reader might have already suspected, this problem can be easily solved by multi-channel cryptography. If multi-channel cryptography creates several harmed texts having no meaning, their storage is the key to solve the problem. Such storage must satisfy a number of requirements:

- storage costs some money;
- in case of corruption or loss of stored data the depository guarantees payment of a stipulated compensation;
- stored data must be returned to a legitimate owner at any moment of the time and uncorrupted (requirement of storage reliability);
- storage should be anonymous;
- a mechanism of proving correctness of depository or owner's actions should be provided in case an adverse party behaves dishonestly.

Here we shall consider several protocols necessary at such storage.

Let Alice be an owner of the data, Bob is a remote depository. Alice has the MV2-RSA system, Bob -RSA only. First it's necessary to go through mutual authentication of participants according to the protocol p. 4.2.

1. The protocol of mutual authentication.

1. A: A;
   $r \in_R Z_p$;
   $\{F_r = E_1(r, K_{MV2})\} \to B$;
   $C_r = E_2(r, K_{MV2})$;
   $\{Y_r = E_3(C_r, k_{Apr})\} \to B$;

2. B: B;
   $C_r = E_4(Y_r, k_{Apb})$;
   $r = E_{12}^{-1}(F_r, C_r, K_{MV2})$;
   $e \in_R \{0, ...2^t - 1\}$;
   $\{F_e = E_1(e, K_{MV2})\} \to A$;
   $C_e = E_2(e, K_{MV2})$;
   $\{Y_e = E_3(C_e, k_{Bpr})\} \to A$;
3. A: A;
   $C_e = E_4(Y_e, k_{Bpb})$;
   $e = E_{12}^{-1}(F_e, C_e, K_{MV2})$;
   $s = e + r$;
   $\{F_s = E_1(s, K_{MV2})\} \to B$;
   $C_s = E_2(s, K_{MV2})$;
   $\{Y_s = E_3(C_s, k_{Apr})\} \to B$;
4. B:
   $C_s = E_4(Y_s, k_{Apb})$;
   $s' = E_{12}^{-1}(F_s, C_s, K_{MV2})$;
   $e' = s' - r$;
   $e' = e$?!

Then, if authentication went successfully, you should go directly to the protocol of storage.

2. The protocol of remote storage based on the MV2-RSA algorithm.

1. A: A;
   $F = E_1(M, K_{MV2})$;
   $C = E_2(M, K_{MV2})$;
   $Y_c = h(C)$;
   $\{Y_{AC} = E_3(C, k_{Bpb})\} \to B$;
2. B: B;
   $C = E_4(Y_{AC}, k_{Bpr})$;
   $Y_c' = h(C)$;
   $\{Y_{BC} = E_3(Y_c', k_{Apb})\} \to A$;

3. A: A;
   $Y'_c = E_4(Y_{BC}, k_{Apr})$;
   $Y_c = Y'_c$?!;
   $\{Y'_{AC} = E_3(Y_c, k_{Apr})\} \to B$;
4. B: B;
   Y_c" $= E_4(Y'_{AC}, k_{Apb})$;
   $Y'_c = Y_c$"?!
   $\{Y_{BC} = E_3(Y'_c, k_{Bpr})\} \to A$;

In the result of such a protocol the depository has a core C (it is an object of storage) and Y'_{AC} (it's document proving that a person A really deposited the object C).

The owner of information A in his turn keeps Y_{BC} as a prove that a depository B took exactly the core C for storage.

Thus, the owner of the information stores remotely the core C, and in his computer he has flags F and keys only. The core C and the flags F can be interchanged, and the flags will be stored remotely in this case, but the storage will cost more.

Storage in two remote places is possible: in one place you keep a core, in the other – flags.

The following item is of practical interest: separate storage of information of large contents in form of files and folders on compact carriers of low capacity like smart-cards or tokens. In this case every file is divided into a core and flags, and then an integral (possibly large enough) summary core is formed. Then an MV2 transformation is used for a second time, in the result of this we get a core of small sized cores which is stored separately from the flags in a data carrier of low capacity.

5.4 Trade

Today high technologies are widely presented on the

market: stock-taking and movement of goods, cash-registers based on PCs, credit cards and other services.

But today one question remains open: what does the buyer purchase: an original certified product or a forgery? This question is very important when it deals with medicine, foodstuff, toys and everything that can harm human health. According to estimations of specialists the world of forgery takes up to about 10% from the world's production. Even presence of a certificate doesn't help, a certificate can turn out to be false. There's only one person in the chain "producer – seller – buyer" who can answer the question about goods' authenticity – it is a producer. But the producer can't evaluate every goods' item which the buyer holds. In fact he can, if he himself gives an *authentic certificate*. It's just necessary to comply with one rule: *goods without a producer's certificate are declared forged*. This problem is based on technical, informational and organizational measures where the two latter take the main place.

Let's remember the following trick from espionage films: somebody takes a photo or a banknote and tear it arbitrary into two parts. And the more deckle the edges of the tearing the better. One of the parts is given to one person, the other – to another one. Presenters recognize each other if the edges coincide. We won't discuss disadvantages of this method here (interception of a part of this password by another person and so on), we shall use the idea itself. Let's complicate this trick: imagine you have a ciphertext on a sheet of paper. You tore it into two (or more if necessary) parts and not only gave it to two people to recognize each other, but told them the contents of this message. Then, during authentication it's necessary not to only put together the document, but to know its contents before decryption to compare with the contents of the decrypted document. Such a document can be used

only once; if somebody presents such a document repeatedly it means that one of the presenters is a cheat.

This concept leads us to the following model.

The producer forms a decryption of every item of the goods (its name and characteristics) with an individual number with the help of the MV2 algorithm.

At that he leaves the flags F in his database, and places the short core C on the label of the goods or directly to the goods in form of a bar-code. During scanning the information is sent to the producer's server where it is reunited with the information F, and we get a description of the goods with an individual number. This description is transmitted to the shop to a cash register in form of a receipt. The buyer (or a machine) evaluates identity of the purchase and the description. Here is what depicted on the label in form of a bar-code:

- Country code IC.
- Producer's ID IDP.
- Producer's UID
  UIDP=IC||IDP.
- ID of goods/ or a batch of goods IG.
- UID of goods (a batch of goods)
  UIDG = UIDP||UIDL||IG.
- Data of producer DP – any data added by the producer to characterize goods
- Checksum KS – a test value added to control data integrity.
- Label's data DTAG = UIDG||DP.

- Unique label
  UTAG = UIDG||C(DTAG||KS1)||KS2.

An image corresponding to a unique label:
PUTAG = UIDG||F(DTAG|| KS1).
Object: IC IDP UIDL IG C(DTAG||KS1) KS2
Minimal number of bits

10
20
32
32
64
10
Total: 168

168 bits are presented as 30,9 43-digit symbols, 21 bytes or 25,3 100-digit symbols.

In other words to present an object 21 bytes of numerical information is required, or 31 symbol in the bar-code CODE39 or 26 symbols in the bar-code CODE128.

Such systems are highly recommended to associations on protection of buyers and producers against fraud. The buyer will be able to check a purchase in the terminal of the association out of the shop.

The same systems can successfully struggle against infringing goods on borders of countries as they possess hierarchical properties: protection of a container, pallet, package, an item Below there are possible types of attacks:

1. A repeated sale from the same shop.
2. A repeated sale from different shops.
3. A sale of an unregistered product.

4. A check-up of an unregistered product.
5. A check-up of a blocked product.
6. Using a blocked certificate of the terminal.
7. Using an invalid certificate of the terminal.
8. Using an unregistered certificate of the terminal.
9. Using an invalid certificate of the producer.

Necessary reaction of the system:

1. A repeated sale from the same shop – it is detected by a checking terminal of a shop. Indication of the event is displayed with a special color. The terminal denies the sale; message about the address and time of the previous sale.
2. A repeated sale from different shops – it is detected by a checking terminal of a shop at a repeated sale. Indication of the event is displayed with a special color. The terminal denies the sale; message about the address and time of the previous sale.
3. A sale of an unregistered product – a sale of an unregistered product. The terminal sends a message "The product is not registered". The terminal denies the sale.
4. A check-up of an unregistered product (of an arbitrary generated forged label) – it is detected by an independent terminal. Message "The label doesn't exist".
5. Using a blocked certificate of the terminal – the system denies access.

6. Using an invalid certificate of the terminal – the system denies access.

7. Using an unregistered certificate of the terminal – the system denies access.

8. Using an invalid certificate of the producer – it is detected by checking terminals. A checked label doesn't exist.

9. Shifting a label from one product to another before the sale. The attack cannot be detected. Protection from the attack should be provided by the technology of attaching a label to goods. Detaching a label from a product should lead to label destruction or to impossibility of attaching it to another product.

5.5 Protection of documents on paper carriers

Paper documents belong to the class of documents with visually spread information. Historically they are the most wide spread kind of documents, though only paper plays the role of carrier here. From this point of view we shall call all documents on which information for visual perception is placed and which are equivalent to a paper carrier (plastic, fabric and so on) as paper documents.

Distinctive marks for paper documents can be watermarks, paper work, more often – a stamp and a signature of an official. A combination of these attributes is possible. All these authenticity indications can be easily forged.

Here we shall consider a technology of creating authentic paper documents and checking them by methods of multichannel cryptography.

A producer of a document makes it visual copy with all authenticity signs and encrypts this visual image with the MV2 algorithm. He places the flags to the database, and the core (in form of a bar-code) is placed on the document.

During checking for authenticity a controller reads a bar-code and sends it to the database. The corresponding flags are detected according to the read core, and a visual image of the document is restored. This image is sent to an askable person where comparison of the presented and authentic document takes place.

A certification center can be created in this technology which will guarantee incorruptibility of persons who created a document. The tree-channel algorithm MV3 can be used for these purposes where the third channel belongs to the certification center.

5.6 Protection of corporeal property

We shall consider only the corporeal property which has paper documents where the owner is registered and there's a description of the subject. We shall consider a well-known problem of an autotheft. First of all this problem is interesting because it requires juridical, technical, organizational and informational solutions. It hasn't been solved up to now only because of unsolved informational problems that can be solved by multichannel cryptography.

Here is a typical scheme of an autotheft in Western Europe with driving cars to former USSR countries. Criminal elements come to an agreement with a car owner who gets

the insurance and bribe from the criminals and doesn't inform police about a theft for 2 to 3 days. He may declare to be away or ill for 2-3 days. The criminals make all the necessary forged documents and within this period drive the car to another country. Then they will register it with other fake documents and changed numbers. Annual European autotheft according to this or another scheme is 20-30 thousands cars per year minimally.

Here we shall consider a technical, organizational and informational aspects of this problem based on multichannel cryptography.

An informational aspect – a car is provided with an authentic passport with a marked data-pump in form of a bar-code in compliance with 5.5. At that the database of the flags for all issued car passports is maintained. Distributed information storage and the two-channel encryption algorithm MV2 are used here.

An organizational aspect – if a car changes its owner the passport of the car should be changed as well. Checking authenticity if a log book takes place during maintaining service, crossing borders and if necessary.

A technical aspect – checking authenticity of a passport should be implemented via the Internet with the help of hardware. (PC).

During checking a passport for authenticity a controller reads a bar-code of the core which is sent to the database. The core together with flags decrypts the received information, and a restored passport is resent to the controller.

A verifiable passport is compared with a received document. Any forgery is excluded. Using such a technology a car can be stolen and taken to spare pieces, but it cannot be driven abroad or registered (provided there are no criminal groups or people violating this technology in the corresponding

organizations).

5.7 Protection of intellectual property in multimedia

This problem is of current importance and it hasn't been solved up to now firstly, because the object of protection is physically intangible. Available mechanisms of a copyright don't work properly, are not very effective, require high expenses and became out of date long ago. Multichannel cryptography lets easily solve this problem. Here we shall only consider the idea of such protection which doesn't lead to prohibition of copying, but makes the process useless. In general it isn't lawfully to prohibit copying if a buyer purchased a disc and would like to have a copy in case he loses the original. He doesn't have the right to distribute copies the contents of which may be intellectual property. Below there's the main idea of such protection.

A plaintext is encrypted by the MV2 algorithm by the key of a disk producer; only flags are placed on the disks. Note that all disks contain the same information (that is the necessary condition at mass duplication). During purchasing the cores of tracks are encrypted by an open key of a disk buyer on a smart-card or a token. Thus, we have a universal record on a disk and an individual record on a portable carrier. The latter can be sent via the Internet. To play the disk it's necessary to know a secret key of an asymmetric system. For a universal record-player this technology requires an additional chip for decoder implementation. There's a special MV3 algorithm to realize this idea.

5.8 Summary

Practically all mass technologies require some standard methods which are provided by multichannel cryptography: cryptographic information splitting into two or more parts, presenting it in form of harmed ciphertexts, distributed storage of harmed texts and transmission of short texts (data-pumps) (Fig. 5.2). A type of a manipulation is determined by this or that mass technology.

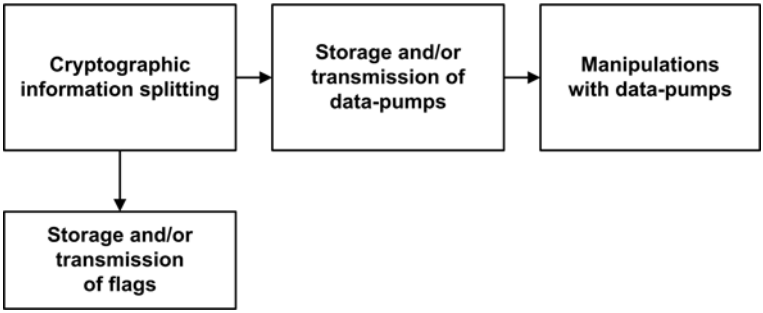

Fig. 5.2: *Standard methods of mass technologies*

Distributed storage of cores and flags makes a ciphertext unavailable for criminal elements, and consequently provides impossibility of organizing such attacks. In case when flags are not subject to transmission (due to the main point of a technology), such a system is close to the ideal one due to a very long unicity distance. If at this transmission of short cores takes place secretly, the attacker won't get any ciphertext at all. The cores can also be additionally encrypted in case of open transmission.

Conclusion

In this book we consider a new direction in cryptography – a multichannel cryptography based on generation and use of several harmed texts each of them having no sense either in a plaintext alphabet or in a ciphertext alphabet. Harmed texts generation occurs in compliance with the idea of secret splitting, when it's necessary to know the whole total of harmed texts to restore the ciphertext. We suggest a universal algorithm of harming a text irrespective of text's nature. The peculiarity of our approach is that a set of harmed texts contains elements of manageable length. This allows some elements of the set to successfully use steganographic methods of information concealment which are used for data of a small length. In such a formulation a cryptanalysis faces new, at times irresistible difficulties: a cryptanalyst has at his disposal a corrupted ciphertext which doesn't correspond to an open plaintext.

The peculiarity of multi-channel cryptography in comparison with the single-channel one (an input text – a ciphertext) is an increased number of degrees of freedom during engineering different applications (an input text – several harmed ciphertexts). This will allow manipulating harmed ciphertexts of various length to solve different problems: information distribution, distributed storage, partial sending, concealment or additional encryption. These possibilities allow modernizing

old protocols and creating new ones, synthesizing new cryptosystems with amazing properties.

These peculiarities of multi-channel cryptography find a direct use in mass technologies of the modern market protecting these technologies from criminal elements. A classical one-channel cryptography can't do it because it doesn't have the necessary degrees of freedom. Just because of this informational aspects in these applications were unsolvable up to now.

In the scope of harmed texts the cryptanalysis faces a new class of problems solutions of which it doesn't have even a theoretical approach today. It is caused, first of all, by harming every alphabet symbol along the whole text length.

This mechanism makes a cryptanalysis use only the brute force for off-stage harmed ciphertexts which in this case play the role of unknown long slave keys. These extra keys is a function of a plaintext, of encryption keys and mechanism of harming. In the result we have a sharp increase in Shannon's unicity distance.

The theory of harmed texts and the multi-channel cryptography based on it just start their way to cryptography. This book is designed to draw attention of specialists to those principally new possibilities this direction creates.

The peculiarity of the concidered approach is the fact that multi channel cryptogrphy organically blends with already reached and practical results. It allows to combine the universal mechanism of harming with known cryptogaraphic systems and protocols braodaning their possibilitiesboth in resistance and in solvable problems. The new cryptographic "brick" MV2-RSA allows building new systems which unite advantages of the systems with secret and open keys, and possess high quick-action and cryptographic resistance.

Appendix A
Terminology and basic definitions

In books about cryptography a reader can come across a number of terms and definitions which are interpreted in different ways by different authors. The known summary of the terminology and definitions accepted today was given in [24]. In this application we place materials of this summary.

Cryptographic algorithm

An algorithm implementing computation of one of *cryptographic functions*.

Encryption algorithm

An algorithm combining an *encryption algorithm* and a *decryption algorithm*, implementing a basic encryption algorithm in a certain *encryption mode*, and performing a transformation of a set of open *messages* into a set of *encrypted messages*, depending on a *key*.

A basic block encryption algorithm

An encryption algorithm which realizes the same reversible transformation which depends on a *key* for each block of a fixed length plaintext (as a rule a block length is divisible by the length of a machine word).

Quantum cryptography

A subsection of *cryptography* devoted to use of quantum physics methods for a synthesis and analysis of *cryptographic systems*.

Cryptography

An area of science, applied technical engineering researches and practice which studies development, analysis and ground of security of *cryptographic means* of information protection from threats from the side of an *adversary (attacker)*. Thus it solves the following main tasks: it ensures *confidentiality, integrity, authentication, non-repudiation, non-accountability*. Unlike physical and organized information securities under cryptographic means we also understand such hardware which uses mathematical transformation methods of protected information. Cryptography is conventionally divided into two parts: a *cryptosynthesis* and a *cryptanalysis*, note that cryptography includes *cryptology*.

Cryptology

A branch of mathematics and mathematical cybernetics which studies mathematical models of *cryptographic* systems. Like *cryptography* it is also conventionally divided into two parts: a *cryptosynthesis* and a *cryptanalysis*.

Cryptographic system

Safety system of a protected network which uses *cryptomaterial*. As subsystems it can include systems of *encryption, identification, simulative protection* (continuity test), *digital signature* (protection from non-repudiation), etc., as well as a

key system which provides functioning of remaining systems. At the heart of choosing and building a *cryptosystem* there's a condition of providing *cryptographic resistance*. Depending on a key system they distinguish *symmetric and asymmetric cryptosystems*.

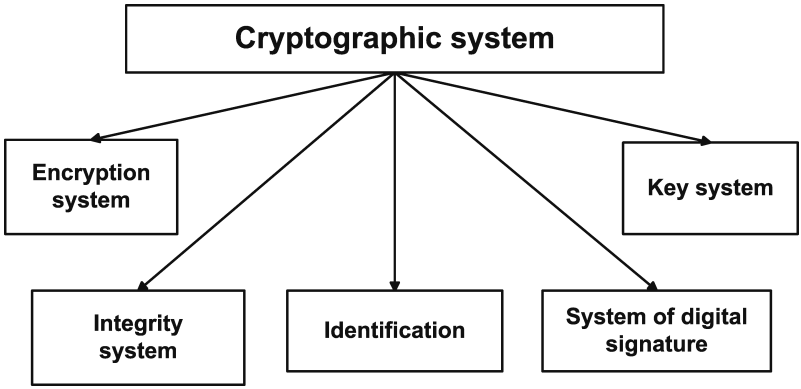

Fig. A.1: *Cryptosystem*

Method of a cryptographic analysis (cryptanalysis)

A set of methods aimed at research of *cryptosystem security*, united by the same idea (mathematical, technical or other). Some methods of analysis lead to building decryption algorithms and are aimed at obtaining estimations of *practical security*; other methods don't lead to building *decryption* algorithms and are aimed at obtaining estimations of *provable security*. We can assume that both authors of *cryptosystems* and an *adversary (attacker)* use the same set of methods of cryptanalysis. *Work content* and *reliability* are usually considered as the most important features of methods of cryptographic analysis.

Adversary (Internal adversary, Dishonest party)
A protocol party, breaking actions set by the *protocol*.

Cryptographic operation

The term accepting in *computer cryptography* and introduced into *the standard* ISO/IEC 15408-99 to designate *cryptographic algorithms and protocols*. Under cryptographic operations we understand the following: data or *key encryption and decryption*, forming and checking a *digital signature* or *Message Authentication Code*, computing the value of a *hash-function* and the protocol of key generation and others. In earlier standards the term *cryptographic mechanism* was used.

Cryptographic primitive

A function (a family of functions), possessing a certain cryptographic property. Cryptographic primitives are used at building *cryptosystems* and *cryptographic protocols*. The most important examples are: *one-way function, hash-function, a pseudorandom sequence generator, a family of pseudorandom functions*.

A cryptographic property is unique for every primitive and serves as basis for its determination: for a one-way function, this is computing complexity of its inversion, for hash-functions this is computing complexity of the task of collision search and so on. Sometimes they call cryptographic primitives such objects as *digital signature, electronic money*, certificate of *open key* etc, provided that they are used as construction elements of a *cryptographic protocol*.

Adversary (External adversary)

An outside (regarding *protocol parties*) subject (or coalition of subjects) watching *protocol* messages being transmitted and having the possibility of intercepting work of the protocol by distortion (modification), insertion (creation of new), repeating and readdressing messages, blocking transmission

and so on with the purpose of violating one or several security functions. It can form a coalition with an inside *adversary*.

Cryptographic protocol

A protocol designed to execute functions of a *cryptographic system*, in the process of which parties use *cryptographic algorithms*.

Quantum cryptographic protocol

A *cryptographic protocol* which uses a *quantum communication channel*.

Applied cryptographic protocol

A cryptographic protocol designed to solve real-world problems with providing *safety function* with the help of *cryptosystems*. Examples: *a protocol of confidential data transmission, a protocol of digital signature, a voting protocol, a contract signing protocol* and others.

Primitive cryptographic protocol

A protocol which doesn't possess an independent applied significance, but is used as a component at building more complex *applied cryptographic protocols*. Examples: a protocol of bit salt, *a protocol of coin flipping*.

Property of meshing

Not a strictly formalizable property of a *cryptographic transformation* that lies in a considerable complication of dependences between a *key* and an *encryption text*. The term was introduced by C. Shannon.

Mixing property

Not a strictly formalized property of a *cryptographic transformation* , that lies in a considerable complication of correlation between static and analytic characteristics of *plaintexts* and *encrypted texts*. The term was introduced by C. Shannon and borrowed from the ergodic theory.

Property of dispersion

Not a strictly formalized property of a *cryptographic transformation* that lies in distribution of influence of every symbol of a *plaintext* on a large number of symbols of an *encrypted text*. The term was introduced by C. Shannon.

Identification system

A cryptographic system, performing the function of *parts' authentication* in the process of information interaction. A mathematical model of the system includes an *authentication protocol* and a *key system*.

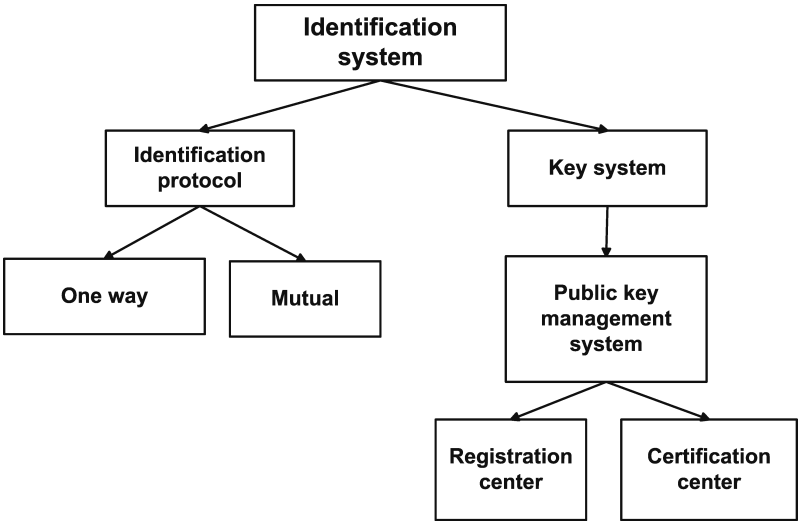

Fig. A.2: *Identification system*

Integrity system

A cryptographic system, performing the function of *message authentication* and is designed to protect from non-authorized change of information or from obtrusion of false information. A mathematical model of the integrity system includes a *cryptographic algorithm of integrity encoding* of information

(in can be *a cryptoalgorithm*, an *Message Authentication Code* or another transformation) and decision-making algorithm for validity of received information and a *key system*.

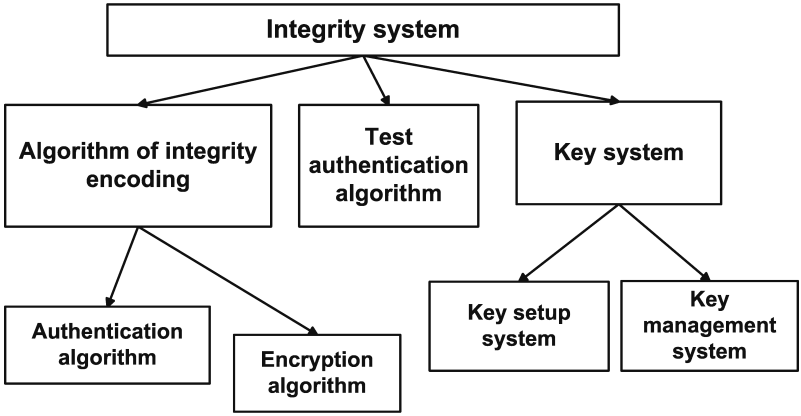

Fig. A.3: *System of simulation protection*

Key system

Consists of the *key set* and key *setup and management subsystems*.

System of digital signatures

A cryptographic system, performing the function of *information source authentication* or *messages* and is designed to protect from non-repudiation. For instance, a sender can repudiate the fact of message transmission by claiming that an addressee himself created this message, and the addressee can easily modify, substitute or create a new message and then claim that he received it from the sender. A mathematical model of the system of *digital signature* includes a *a scheme of digital signature a key system*.

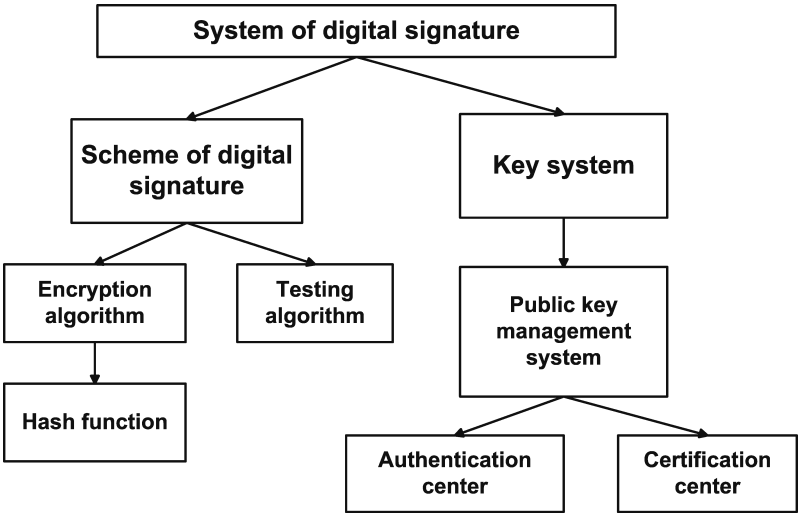

Fig. A.4: *Scheme of digital signature*

Practical security of a cryptosystem

Time complexity of performing a successful *cryptosystem attack* by the fastest from the known *cryptanalysis methods* for the real assumptions about cryptosystem properties and its use, about computers on which the *attack* will be implemented.

Security of cryptographic system (cryptographic protocol)

Capability of a *cryptosystem (cryptoprotocol)* to resist attacks of an *adversary (attacker)*. The concept of security is individual for every type of *cryptosystem and cryptoprotocol* (and as a rule for a variety inside a given type) and can be defined in regard to a certain pair (attack, danger) only. There are two main approaches to the definition of resistance in *cryptography: information-theoretical* and *theoretically-complex*.

Security information-theoretical (shannon, absolute)

Capability of a *cryptosystem (cryptoprotocol)* to resist pressure of an *adversary (attacker)*, who can use an arbitrary algorithm (without limitations to computing resources) to achieve his objectives (implementation of a *threat*).

Synonym: *Perfect cryptographic security (by C. Shannon).*

Theoretically-complex security

Security of a cryptographic system (protocol) against *threats* from the side of an *adversary (attacker)*possessing limited computing resources. Usually, time limitations for algorithm execution are considered. Theoretically-complex security is always based on some *cryptographic assumption.*

Scheme of digital signature

It consists of two *algorithms*, one is to *form*, the other is to *check a signature.* Security of the digital signature scheme is determined by complexity of the following three tasks for a person who doesn't possess a secret key: *signature falsification*, i.e. computing signature value under a set document; *creation of a signature message*, i.e. finding at least one message with a correct value of the signature; *message substitution*, i.e. selecting two different messages with the same values of the signature.

Cryptographic function

A function necessary to implement a cryptographic system. For instance, generating *keys, pseudorandom sequences, an encryption function, a one-way function*, computing and checking values of a *Message Authentication Code* and an *electronic digital signature*, computing values of *hash-functions* and so on; possess certain cryptographic features which influence cryptographic properties: key dependence, usage complexity and so on.

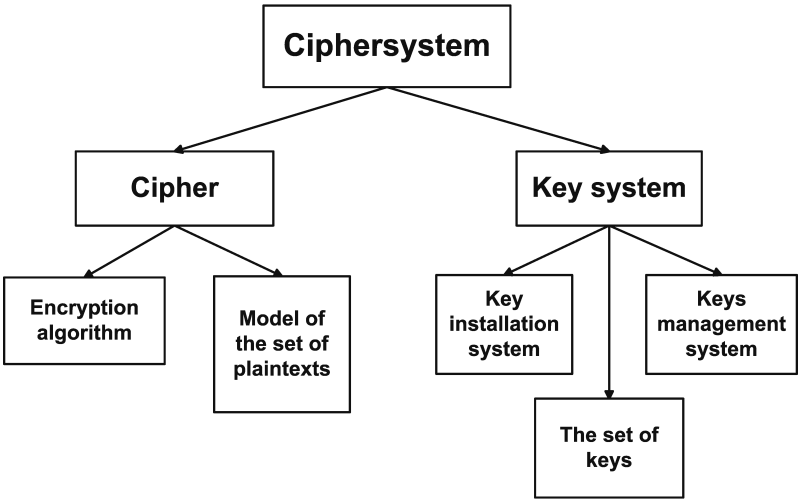

Fig. A.5: *Ciphersystems*

Cryptographic hash-function

An effectively calculable function compressing input strings for which *collision* search problem is computationally difficult. It formalizes as a *one-way hash-function* or as a *hash-function* with *difficult-to-locate collisions*.

Ciphersystem

A cryptographic system is designed to protect information from reading by a non-authorized person by using information *encryption means*. A mathematical model of a ciphersystem includes encoding of input and output information *a cipher* and *a key system*.

Cipher

A family of invertible mappings (encryption mappings) of a set of *open texts (messages)) blocks* into a set of *encrypted texts (messages) blocks* and back, each of them is determined by a certain parameter called "key"and is described by some

encryption algorithm, which implements one of *encryption modes*. A mathematical model of a cipher includes two algorithms: *encryption* and *decryption* (altogether referred to as – *encryption algorithm*), defining an algorithm and encryption mode and a set of open messages. Depending on means of presentation of *plaintexts (messages)* they distinguish block, stream and other types of ciphers. The main requirements defining the quality of a cipher are: *cryptographic resistance, simulation resistance, noise immunity* and others.

Appendix B
Basic implementation of the MV2 algorithm

B.1 Description

In this application one of the implementations of the general scheme of harming (see 3.2) is described. It is one of the implementations of the MV2 algorithm (see 3.3), which we shall further call as *basic*.

This implementation has the following design features:

- a key is a short ($128 \div 2048$ bits) sequence (master key), from which 32 tables are generated that set an MV2-transformation with the parameters $r = 3$ and $n = 7$;
- a stream cipher RC4 is used for whitening;
- a 128-bit linear transformer with a high degree of diffusion is used as a permutation transformation;
- at least 16 rounds are performed.

We shall describe input and output parameters of the basic implementation.

Encryption

Input:	plaintext	M ($8 \times L$ bits)
	secret key	K (128, 256, 512, 1024, 2048 bits)
	number of rounds:	m
	or maximal core length:	L_c
Output:	ciphertext	$\Big(C\,,\,F\Big)$

Decryption

Input:	ciphertext	$\Big(C\,,\,F\Big)$
	secret key	K (128, 256, 512, 1024, 2048 bits)
Output:	error message or a plaintext	M ($8 \times L$ bits)

To ensure the ciphertext could be decrypted back to the message, the encryption transformation has to be invertible, but it's not necessary to use identical algorithms for encryption and decryption. In MV2 encryption and decryption are made by different algorithms. As mentioned in 3.2, a global structure of the encryption algorithm of the MV2 cipher may be shown as an SPN (see Fig. 3.5).

The whole encryption process made by the cryptographic algorithm MV2 could be divided into some rounds, with interleaving of linear and non-linear transformations. Each round consists of a *linear layer* and *non-linear layer*. Mappings with images of various lengths are used to implement nonlinear transformations. These mappings are set with secret tables which constitute key data.

The number of rounds in the basic implementation of the MV2 algorithm can be set directly or indirectly by setting the upper bound for the core length. In any case there will be

no less than 16 transformation rounds performed. Besides, at a set remainder of the upper margin a number of rounds is defined automatically at reaching the set core length.

A plaintext is whitened by the stream encryption cipher RC4 before performing main rounds. One of the permutations is used as an RC4 key. This permutation sets an MV2-transformation and is generated from the key.

A round of the basic implementation

Each round consists of two transformations: a substitution transformation and a permutation transformation. The permutation transformation is made locally. The processed message is divided into 128-bit blocks, each subjected to the same transformation, which rearranges its bits. If the text length is not multiple 128, the last incomplete block is not processed. The permutation transformation is followed by the substitution one set by the selected table for this round.

The permutation transformation is the linear ensuring high degree of local diffusion and it is similar to that one described in [34]. This transformation permutes 128 bits recorded in four 32-digit words.

At each round one of 32 key substitution transformations is performed. The transformation is selected using values of R from the GPS.

A substitution transformation is an MV2-transformation (see 3.1.4).

This transformation maps an n-bit string x into a pair (c, f), consisting of two variable length strings. In the basic realization $n = 8$ and $r = 3$, therefore each byte of the input text is mapped into a pair of bit strings, one of which (remainder) is 3 to 7 bits long, the second one is a value code in the range from 1 to 6. This transformation can be set by the table, where a permutation $x_0, \ldots x_{255}$ of values 0 to 255

is in the left part and images, consisting of the "remainder" and "flag" parts, are stored in the right part (see 3.1.2 and Tables B.1 3.2).

Tabl. B.1: *Task of a substitution transformation*

Symbol	Image length	Remainder	Flag
s_1	3	000	00001
s_2	3	001	00001
...	...	...	...
s_8	3	111	00001
s_9	4	0000	0001
...	...	...	...
s_{24}	4	1111	0001
s_{25}	5	00000	001
...	...	...	...
s_{56}	5	11111	001
s_{57}	6	000000	01
...	...	...	...
s_{120}	6	111111	01
s_{121}	7	000000	1
...	...	...	...
s_{248}	7	1111111	1
s_{249}	3	000	00000
...	...	...	...
s_{256}	3	111	00000

Key

The key is an arbitrary binary string which can have a length of 128, 256, 512, 1024 or 2048 bits. In the algorithm a special key transformation is used, which maps a received key into a set of substitution tables. A set of transformations is

set by a string having the following format: the first 256 bytes contain permutation of all the numbers from 0 to 255 that define a bijective mapping, and 4 standby bytes. At using, the permutation is displayed into a table which sets an MV2-transformation, as shown in the Table B.1.

The algorithm of substitution transformation generation is resistant to a linear and differential cryptanalysis.

In the basic realization of the algorithm the number of substitution tables in the key is limited to 32 (in this case to give a transformation number 5 binary bits are required).

Random number generator

The basic implementation of the algorithm is an iterative probabilistic cipher.

The generator is used to randomize the cipher, i.e. a random number of a table is generated and used at each current encryption round. For this purpose a random number generator (GPS) built on the basis of the affine transformation is used:

$$x_{t+1} = 2^{13}(x_t + x_{t-1} + x_{t-2}) \bmod (2^{32} - 5).$$

The initial state of the GPS is reset by the timer during initialization of the device.

Presentation of output data

Algorithm's output

Output data consist of two binary sequences we call a core and string of flags. The core is a remainder obtained at the last round. A string of flags is a concatenation of output flags obtained at all transformation rounds. For convenience of decryption, the flags round outputs are presented in a reverse order (the last flag round output, the one before, ..., the first flag round output) without separators.

Output of each round

As substitution transformations have bit outputs, while the minimal data storage unit in modern computer systems is byte, in the general case, the algorithm's outputs have to be complemented to a byte value. At the same time we have to know the true length in bits for decryption. Therefore, 1 byte of service data is fore-added to the obtained remainder, this byte comprises the table number (5 bits) and the number of real bits in the last byte (3 bits). The obtained Flags are added to the previous ones.

For such a presentation, strings of flags practically have no redundancy.

B.2 Statistical estimations of output data and resistance

Evaluation of output lengths

As it was mentioned it's important to know lengths of output data for practical implementation.

Let M be a plaintext of the length $L(M)$ bytes, and (C, F) be a ciphertext after m rounds of transformations.

In the basic realization of the MV2 algorithm one service byte is added to each result of a round substitution transformation. If $\mathbf{E}(L_m(C))$ is the expectation of the length of the core and $\mathbf{E}(L_m(F))$ is the expectation of the length of the string of flags then

$$\mathbf{E}\Big(L_m(C)\Big) \approx {\mathrm{K_c}}^m \cdot \Big(L(M)+1\Big) + \frac{159}{128} \cdot \frac{1-{\mathrm{K_c}}^{m+1}}{\mathrm{K_f}}, \quad \text{(B.1)}$$

$$\mathbf{E}\Big(L_m(F)\Big) \approx (1-{\mathrm{K_c}}^{m+1}) \cdot (L(M)+1) + \\ + \frac{225}{128} \cdot m - 1 - \frac{1-{\mathrm{K_c}}^{m+1}}{\mathrm{K_f}}, \quad \text{(B.2)}$$

where $\mathrm{K_c} = \dfrac{97}{128} = 0.7578125$ and $\mathrm{K_f} = \dfrac{31}{128} = 0.2421875$ (see expressions (3.43) and (3.44) from the section 3.2.2).

If the core length shouldn't exceed some length L_c, then

$$m \approx 1 + \frac{\log L_c - \log L(M)}{\log \mathrm{K_c}} \tag{B.3}$$

and the total length of the flags

$$\mathbf{E}(L_m(F)) \approx L(M) - L_c + 2 \cdot \frac{\log L_c - \log L(M)}{\log \mathrm{K_c}}. \tag{B.4}$$

The total output length:

$$\mathbf{E}(L(C) + L(F)) \approx L(M) + 2 \cdot \frac{\log L_c - \log L(M)}{\log \mathrm{K_c}}. \tag{B.5}$$

For random L-bytes input text for one round of transformation from the statement 3.3 follows that a number of bytes in the output of the remainder will be restricted to:

$$(\mathrm{K_c} - \sigma_c/8) \cdot L \;\le\; |C|/8 \;\le\; (\mathrm{K_c} + \sigma_c/8) \cdot L,$$

and a number of bytes in the output flags will be restricted to:

$$(\mathrm{K_f} - \sigma_f/8) \cdot L \;\le\; |F|/8 \;\le\; (\mathrm{K_f} + \sigma_f/n) \cdot L,$$

with the probability no less than $1 - L^{-2}$. In these inequalities σ_c and σ_f are standard deviations of the output text length from average values. For a uniformly distributed input text $\sigma_c \approx 1,2$ and $\sigma_f \approx 4,2$.

Note that the value of a standard deviation of the remainder output length is not large and is about 1 bit per plaintext symbol. At the same time the value of a standard deviation of the flag output length is more than $1/2$ byte per 1 byte of a plaintext. On the other hand at each round the flag

output length is defined by the length of a remainder obtained at a previous round, therefore with the probability $1 - L^{-2}$ after executing the m-th round the remainder length $L_C^{(m)}$ (in bytes) won't exceed the value

$$L_c^{(m)} \leq (\mathrm{K_c} + \sigma_c/8)^m \cdot L + 10,$$

and the flag output length $L_F^{(m)}$ (in bytes) won't exceed the value

$$L_F^{(m)} \leq \frac{3}{4}(\mathrm{K_c} + \sigma_c/8)^m \cdot L + 10.$$

About resistance

Note that a simple attack meet-in-the-middle cannot be implemented for MV2. The well-known methods of differential and linear cryptanalysis cannot be applied either. Authors have not found a better attack on MV2 with 16 rounds other than a brute force attack.

There are probably faster attacks, but they should require an unreal amount of selected open texts and memory volume.

Unlike other cryptographic transformations, in our case we may consider not only variants of unknown keys, but other variants as well – an unknown core (a part of encryption result) or an unknown string of flags (the other part of encryption result) at known or unknown keys.

Further in this section the plaintext M shall be considered as a uniform sequence of symbols $x \in \{0, 1\}^n$. Such consideration is justified as, from one side, in the basic realization before implementation of main rounds, the plaintext is being processed by a stream cipher (noise), from the other side, as the result of a certain number of rounds, a randomized text goes to the input of a byte substitution transformation.

Note that for the MV2 cipher the complexity of attacks grows together with the length of the plaintext.

Evaluation of number of texts having the known core and unknown flags

If only the core is known and there's no limitations for a number of rounds, then even if the keys are known, there's an infinite set of texts that give such a core. At a limited number of rounds a set of plaintexts corresponding to the given core is finite.

As a round permutation is fixed, a set of texts having the same remainder is determined by the substitution transformation $T = \left(c, f\right) \in \mathfrak{F}_8^3$. If L_c is a number of bytes in the output of the core and m is a number of performed rounds, then N_C is a number of possible plaintexts corresponding to the given core will be no less than (see the formula (3.70)):

$$N_C \geq 2^{\left(\frac{128}{31}\left(\frac{128}{97}\right)^m - \frac{1}{31}\right)\cdot L_c}$$

For example, if the known core has the length of 1032 bits (128 bytes + 1 byte of service information), $n = 8$, $r = 3$ and $m = 10$ rounds was executed, then no less than 2^{67657} variants of a plaintext (if the key is known) is possible.

Evaluations of number of texts having known flags and an unknown remainder

As it has already been discussed in 3.2, if only the flags output is unknown and the plaintext's length and the number of performed rounds are unknown, this sharply decreases probability to select the plaintext.

Assume in the result of encryption of the plaintext M by the MV2 cipher with the key K, we obtained a cryptotext $(C, F) = MV2(K, M)$. As the MV2 cipher is a pseudorandom function, the task of finding M using known K and F is $2^{H(C)}$ hard. Note that in the real applications where file sizes are bigger than 1024 bytes, after encryption we get $|C| > 128$

bits. Hence, $H(C) > 128$, therefore, complexity of finding X by the known K and F corresponds to modern requirements.

If the number of rounds m is known, then, from (3.61) the number N_F of plintexts which have the string of flags F, is

$$N_F \approx 2^{\frac{\mathrm{K_c}^m}{1-\mathrm{K_c}^{m+1}}\cdot|F|},$$

where $|F|$ is a length of the string of flags (bits) and $\mathrm{K_c} = \frac{97}{128}$.

About inherited properties of the plaintext

If a cryptoanalyst has no a single pair "message-cryptogram", the only thing he might use, would be analysis of properties of the open text that are being inherited by cryptograms. I.e., the real plaintext is replaced by its model, reflecting its most important properties. Then the cryptoanalysis may be built, for example, on the statistical solutions theory. During such an approach, the most important features of the plaintext model are its frequency characteristics. It's practically not possible to reveal a correlation between frequency characteristics of the chosen plaintext model and those ones of flags due to the following:

There are 6 different digits in the alphabet of flags. For each key mapping $T = (c, f) \in \mathcal{F}_8^3$, there are 8 images with flags 5 and 6, and 2^{8-j} images with the flag j, $1 \le j \le 4$. Therefore, if the language model has n symbols, then, for the random mapping T, expectations of the numbers t_j symbols having an image with the flag j are $\mathbf{E}(t_1) = n/2$, $\mathbf{E}(t_2) = n/4$, $\mathbf{E}(t_3) = n/8$, $\mathbf{E}(t_4) = n/16$, $\mathbf{E}(t_5) = n/32$, $\mathbf{E}(t_6) = n/32$. I.e., practically all the symbols cannot be identified using the first flag output. So the frequency characteristics of the first flag output does not correlate with the frequency characteristics of the model language.

Evaluations of the number of texts at an unknown key

As we mentioned before, the cardinality of a set is of substitution transformation $\left|\mathcal{F}_n^r\right| = 2^n!$ (for the basic realization $\left|\mathcal{F}_n^r\right| = 2^8! \approx 2^{1684}$).

If (C, F) is an image of a plaintext M at performed an unknown transformation $T \in \mathcal{F}_8^3$, then the number of different transformations giving the same flags outputs for long enough plaintexts is determined by the formula (3.62).

If the plaintext M contains all the values from $\{0,1\}^8$, then, the exact equality is performed and

$$\prod_{i=3}^{7} 2^i! \approx 2^{1190}.$$

At the known outputs (c, F) the number of performed rounds m can be determined from the rations (B.3), (B.1) and (B.2). Thus, for a long enough core C at an unknown key the number of possible plaintexts N_K will be:

$$N_K \approx 2^{1190 \cdot m}.$$

Security of the cascade

Security of a cascade cipher is characterized by the following theorem of Maurer-Massey ([62]):

Theorem B.1 *A cascade cipher has at least the same security as the first cipher in the cascade.*

A basic implementation of the MV2 algorithms is a cascade of ciphers. In such system, as in any other, the plaintext M and the secret key K are random values.

Statistics of M depend on the nature of the source of plaintexts, while the statistics of K is controlled by the cryptographer. In usual ciphers the encryption process is deterministic, i.e., the cryptotext Y is uniquely determined by the plaintext M and the key K.

A cipher has *property of non-expanding* if there is an ascending sequence of positive integer $n_1, n_2, \ldots$, such, that the first n_i digits $Y_1, Y_2, ...Y_{n_i}$ of the cryptotext together with the secret key uniquely determine the first n_i digits $X_1, X_2, ...X_{n_i}$ of the open text for $i = 1, 2, \ldots$. Ciphers with the property of non-expanding are called non-expanding. A single-round MV2 cipher is a non-expanding cipher, we may simply assume $n_i = \sum_{j=1}^{i} f_i$, where $f_1, f_2, ...$ are flags, and $Y_1, Y_2, ...Y_{n_i}$ are substrings of the remainder, such, that $|Y_i| = f_i$. The following fact is well known (see. [61]).

Property "random input – random output" for non-expanding ciphers: For each selection of k of the secret key K, the cascade consisting of binary-symmetric source (BSS) and a non-expanding cipher creates a BSS as well. Moreover, for any probabilistic key distribution, this cascade shall generate a cryptotext sequence $Y_1, Y_2, ...$ which is statistically independent from the secret key K.

As defined by Shanon, a cipher is *perfectly secure* if the key K is statistically independent from the cryptotext sequence $Y_1, Y_2, ...$ At any attacks using a known ciphertext against a perfectly secure cipher, the attacker cannot obtain any information on the secret key K, irrespective of the amount of ciphertexts his checking on. I.e., the cipher's security is not diminishing at increase of the total size of the encrypted plaintext prior to the secret key change. It follows from the feature "random input - random output" that each non-expanding cipher becomes perfectly secure if BSS is the source

of plaintexts. This means, for such sources MV2 is a perfectly secure cipher.

B.3 Testing of the algorithm

Testing on correspondence to dependence criteria

For testing a basic implementation of the MV2 algorithm on correspondence to dependence criteria the formulae (3.85), (3.86) and (3.87) were used, and also (3.88) and (3.89).

The lengths of outputs of MV2 are dependent on a plintext M, a key K and a randomizer R. As it was mentioned in 3.7, in this case the normalizing coefficient is not completely correct in the expressions (3.86), (3.87), (3.88) and (3.89). This normalization understates values of the degree of avalanche effect and the degree of strict avalanche criterion. But this criteria can be used for the testing of MV2.

The method of testing on corresponding to dependence criteria

The MV2 was tested with four variants of size of an input block. The plaintext sizes of 16, 32, 64 and 128 bytes were considered. For each variant, the test set consisted of 5000 randomly chosen inputs encrypted under a single randomly chosen MV2 key. These examinations are carried out for varying numbers of rounds, from 1 round to 16 rounds. Initial data were taken from a file containing a data generated by a physical random number generator.

We've determined the maximal length of outputs of the core (L_c) and flags (L_f), average length of the core (L_c^*) and string of flags (L_f^*), the average number of output bits changed when changing 1 input bit, and separately for the core and the string of flags, the degree of completeness, the degree of avalanche effect (d_a^c and d_a^f) and the degree of the strict avalanche criterion (d_{sa}^c and d_{sa}^f) were computed.

The degree of completeness, except for one special modification of MV2 has always been ≈ 1.0

The algorithm's output is the core and the string of flags. The lengths of the core and the string of flags are changed at every round. The following expressions were used as the degree of avalanche effect and the degree of the strict avalanche criterion:

$$d_a = \frac{d_a^c \cdot L_c^* + d_a^f \cdot L_f^*}{L_c^* + L_f^*},$$

$$d_{sa} = \frac{d_{sa}^c \cdot L_c^* + d_{sa}^f \cdot L_f^*}{L_c^* + L_f^*}.$$

Dependence of d_a and d_{sa} on size of input data

A substitution transformation (see section 3.1.4) for each entry byte assigns a reminder, which has the length from 3 to 7 bits. If a plaintext is short, only some part of a substitution table is used for the text transformation on every round; besides, the 128-bit fixed permutation is used, which doesn't work for texts less than 16 bytes long, therefore,we may assume that values d_a and d_{sa} must depend on the size of entry data.

Fig. B.1 and B.2 show values of d_a and d_{sa} at different input lengths. It's obvious that with the growth of input

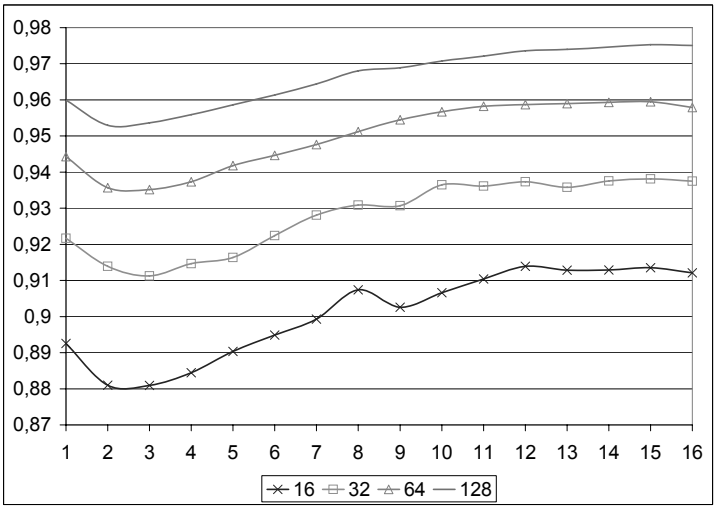

Fig. B.1: *Comparison of criteria d_a for 128, 64, 32 and 16 -byte inputs*

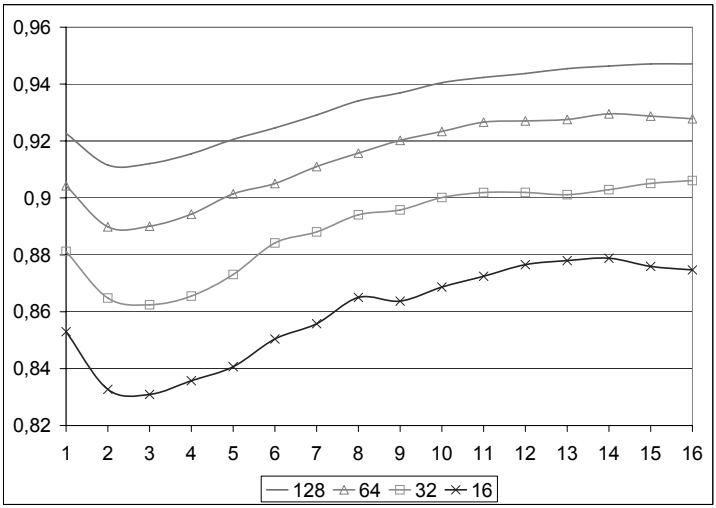

Fig. B.2: *Comparison of criteria d_{sa}(strict avalanche effect) for 128, 64, 32 and 16 -byte inputs*

lengths, the values of d_a and d_{sa} are growing as well for all rounds.

On the choice of a permutation transformation

There are several mechanisms for providing diffusion in the round function. In the basic realization of the MV2 algorithm a 128-bit linear transformer is used to ensure dispersion. Further we'll call it as a basic one.

To determine the influence of a permutation transformation, we made changes to the source code of the algorithm and computed values d_a and d_{sa} for different permutation transformations. We've considered the following variants:

1) with a basic permutation transformation;

2) with the affine byte permutation transformation;

3) with "armenian" shuffle [60].

4) without a permutation transformation;

As the charts in Fig. B.3 and B.4 show, different permutations practically have no influence on the values d_a and d_{sa}.

We conclude that this effect is a corollary of the pseudorandom change of substitution transformations at each round. To check the hypothesis, the random number generator has been turned off in the algorithm's source code and a fixed sequence of values was used for selection of the round substitution transformation. Thus, during all tests the same substitution transformations were made for the same rounds. For these conditions, degrees of avalanche, completeness and strict avalanche for 128-byte inputs were computed.

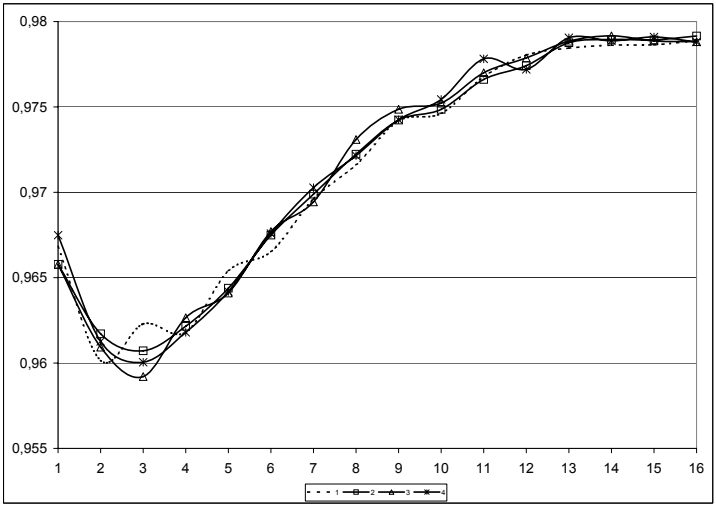

Fig. B.3: *Comparison of criteria of avalanche effect d_a for 128-bit random inputs at different permutations. 1 – "armenian", 2 – basic, 3 – affine permutation, 4 – without a permutation transformation;*

At these conditions degrees of completeness, avalanche and strict avalanche were computed for 128-bit inputs. The tests results are displayed in the charts in the Fig. B.5 and B.6. Upper charts (MV2) in Fig. B.5, B.6 and B.7, B.8 correspond to a usual (pseudorandom) table shuffle without a linear transformation before the substitution.

We can see from these charts, that a pseudorandom choice of substitution transformation at each round has greater influence on the values d_a and d_{sa}, than a permutation transformation.

As the difference between the values d_a and d_{sa} till the 4th round and after it is not very big (Fig. B.5, B.6), in Fig. B.7 and B.8 the values d_a and d_{sa},were concluded beginning with the 4th round.

One can see from the charts in Fig. B.7 and B.8, that the

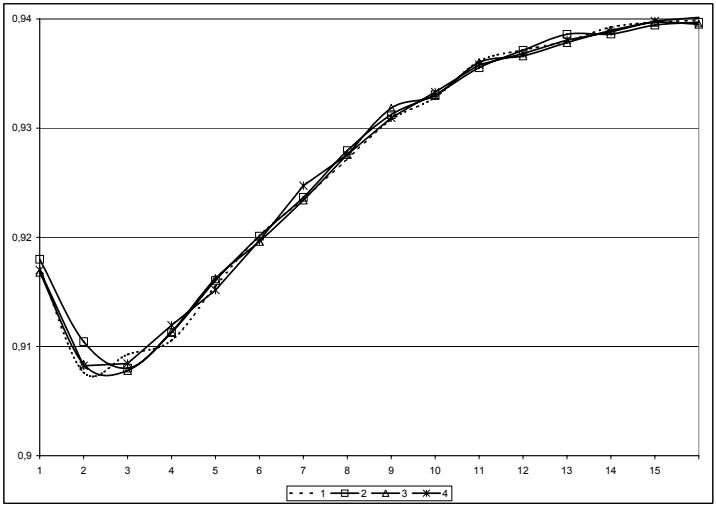

Fig. B.4: *Comparison of criteria of s strict avalanche effect d_{sa} for 128-bit random inputs at different permutations. 1 – "armenian", 2 – basic, 3 – affine permutation, 4 – without a permutation transformation;*

basic permutation has a better influence on the values d_a and d_{sa}.

The results of testing (Fig. B.3 – B.8) allow drawing conclusions about the following: firstly, a pseudorandom change of permutation transformation has bigger influence on values of dependence criteria, than that one of a linear transformation, and, secondly, a chosen in the basic realization permutation transformation is better than others considered ones.

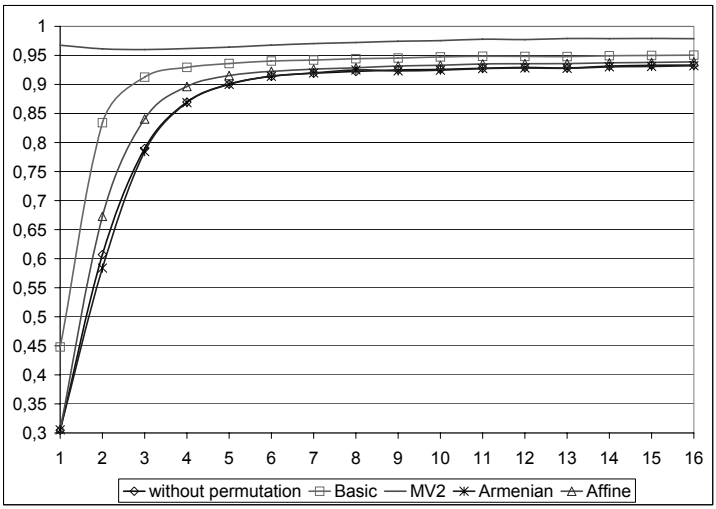

Fig. B.5: *Comparison of the criteria d_a for different permutations at the fixed substitution transformations*

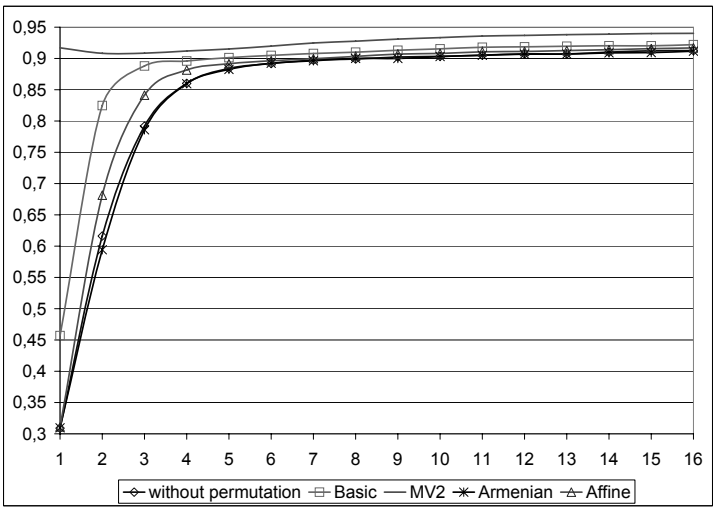

Fig. B.6: *Comparison of the criteria d_{sa} for different permutations at the fixed substitution transformations*

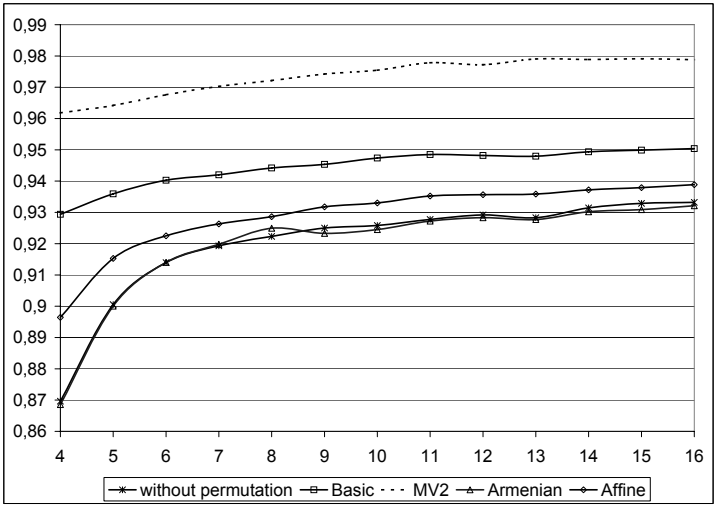

Fig. B.7: *Comparison of the criteria d_a for different permutation transformations at the fixed substitutions beginning with the 4th round*

Influence of whitening

Let's remember, the MV2 encryption algorithm is a cascade of a stream cipher and the general scheme of harming. Application of a stream cipher ensures whitening of a plaintext.

To check its impact on the degree of completeness, degrees d_a and d_{sa} we carry out tests for the MV2 with whitening and MV2 without whitening (Fig. B.9, B.10). The first test consists of 5000 randomly selected 16- and 128-byte bytes plaintexts. The second test consists of 256 128-byte homogeneous plaintexts $M = x^{128}$ where $x \in \{0,1\}^8$. Under a homogeneous input here we understand a sequence consisting of the same bytes. The following designations are used in these charts: 1a, 2a and 1b, 2b – correspondingly 16- (1), 128-byte (2) random inputs at presence (a) or absence (b) of whitening, 3a, 4a and 3b, 4b – correspondingly 16- (3), 128-

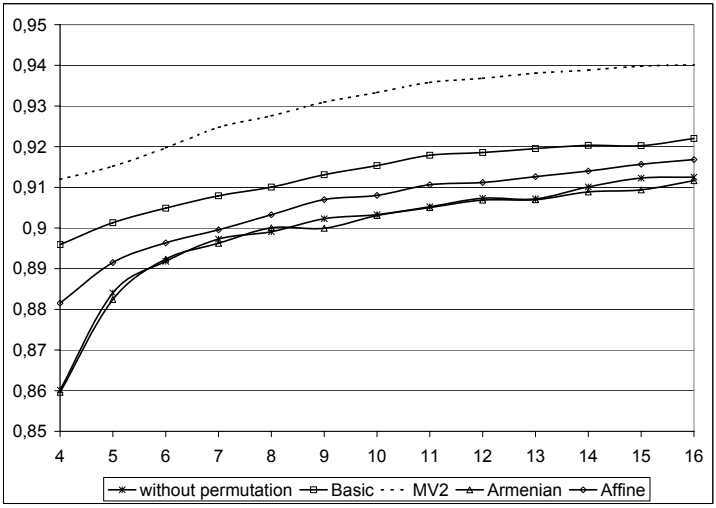

Fig. B.8: *Comparison of the criteria d_{sa} for different permutation transformations at the fixed substitutions beginning with the 4th round*

byte (4) homogeneous inputs at presence (a) or absence (b) of whitening.

From the charts displayed in Fig. B.9 and Fig. B.10 it follows, that whitening has a significant impact on values of degrees of avalanche criteria and a strict avalanche criteria in case of homogeneous inputs. This impact grows at increase of the input length.

Whitening substantially increased the difficulty of attacking the cipher, by hiding from an attacker the specific inputs to the first round.

Whitening and the flag output

We also carried out testing to evaluate impact of whitening on the flag output. As the mappings satisfying SAC, satisfy other criteria as well, then, the flag output was tested for correspondence on a strict avalanche criterion. The degree of a

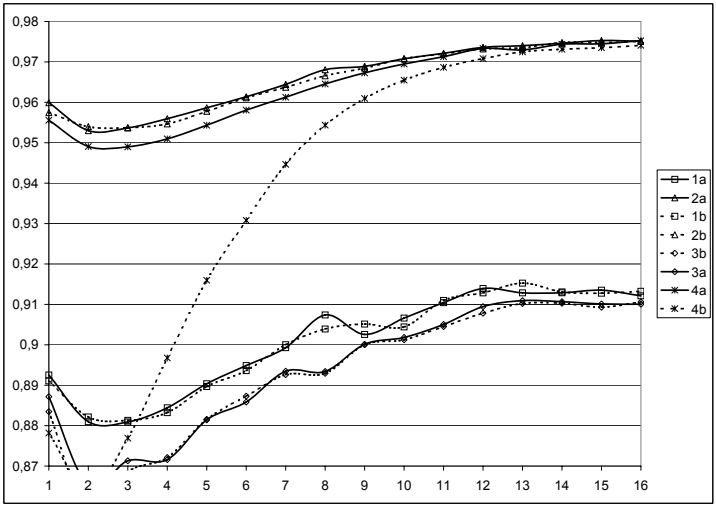

Fig. B.9: *Dependence $d_a$on presence of whitening and type of input data at 16- and 128-byte inputs:*

strict avalanche criteria was defined by the formula (3.89). At computing according to this formula it's possible to estimate an error occurred due to a variable input length.

The charts of degree of a SAC of the flag output on the number of rounds for different input lengths and tests are showed in Fig. B.11.

Sequences consisting of 16, 32, 64, 128 and 256 bytes went to the input. For each input length three groups of tests of computing the degree of a strict avalanche criterion were carried out:

1. A test for homogeneous inputs without whitening (Fig. B.11, charts 1A – 1E);

2. A test for homogeneous inputs with whitening (Fig. B.11, charts 2A – 2E);

3. A test for random inputs without whitening (Fig. B.11,

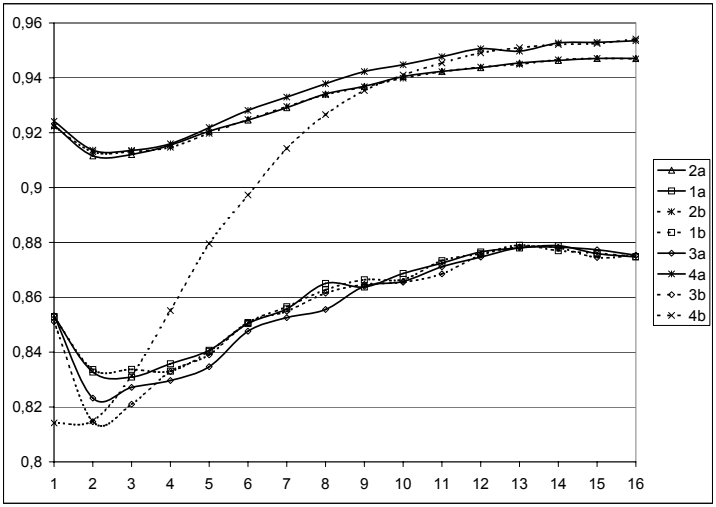

Fig. B.10: *Dependence d_{sa} on presence of whitening and type of input data at 16- and 128 byte inputs*

charts 3A – 3E).

From the comparison of charts displayed in Fig. B.11, we can draw conclusions about the following:

- values of the SAC degree grow at increasing an input length;

- at increasing a number of rounds the values of a SAC degree start decreasing for short inputs (16, 32, 64 byte).

Usually, the MV2 algorithm is used to encrypt data of large capacity, therefore the most interesting case is to consider values of SAC degrees for 256-byte inputs. The corresponding charts are represented in Fig. B.12.

We can see from the charts displayed in Fig. B.12, that:

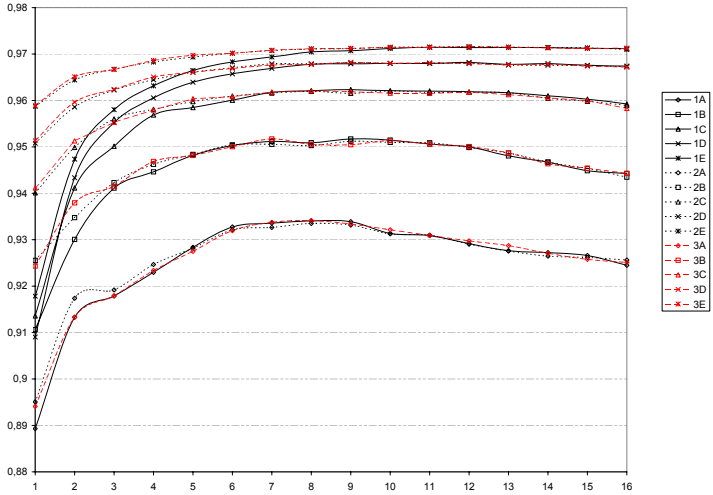

Fig. B.11: *Charts of dependence degree of a SAC of the flag output on the number of rounds for different output lengths: 1 – homogeneous inputs without whitening; 2 – homogeneous inputs with whitening; 3 – random inputs without whitening; A – 16, B – 32, C – 64, D – 128, E – 256-byte inputs*

- values of a SAC degree at homogeneous inputs and whitening behave in the same way as at a random input without whitening;

- at first rounds if there's no whitening, the values of a SAC degree are considerably smaller than in the case with whitening.

From the charts (Fig. B.11 and B.12) we can see that whitening has a considerable impact on values of degree of an avalanche and a strict avalanche of the flag output in case of homogeneous inputs. Whitening considerably increases the difficulty of attacking by known flags by concealing from an attacker the specific inputs at the first round.

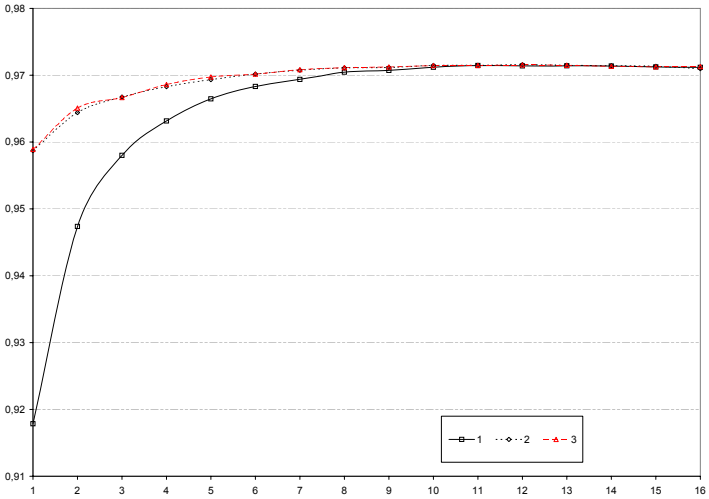

Fig. B.12: *Charts of dependence of a SAC degree of the flag output on a number of flags for 256-byte inputs: 1 – homogeneous inputs without whitening; 2 – homogeneous inputs with whitening; 3 – random inputs without whitening*

For 256-byte inputs degrees of a SAC of various texts began to coincide after 7 transformation rounds. Consequently, we can draw a conclusion that for long input texts it's recommended to perform no less than 7 encryption rounds. For short input texts (less than 128 byte), on the contrary, it's not recommended to perform more than 10 encryption rounds. Consequently implementation of the algorithm should be built in such a way that wouldn't allow short (less that 16 bytes) remainder outputs, at that no less than 8 transformation rounds should be performed.

The length of the flag output

In the table. B.2 there are average values of flags output lengths. The values at which the average output length is bigger than an input length are in bold type. The correlation

of the table with the charts in Fig. B.12 allows drawing the conclusion that the optimal number of rounds depends on a length of an input text.

Tabl. B.2: *Tentative and calculated (in the parenthesis) average lengths of the flag output (in bytes) at different input lengths for varying number of rounds*

	16	*32*	*64*	*128*	*256*
1	5,0(6,2)	8,9(13,0)	16,6(26,7)	32,1(53,9)	63,0(108,4)
2	8,9(9,8)	15,7(18,8)	29,3(36,9)	56,5 (73,0)	110,9(145,3)
3	12,3(12,9)	21,3(23,6)	39,4(45,1)	75,4 (88,0)	147,6(173,7)
4	15,3(15,7)	26,0(27,7)	47,4(51,7)	90,2 (99,7)	175,9(195,7)
5	**18,1**(18,2)	30,1(31,2)	54,0(57,1)	101,9(109,0)	197,7(212,8)
6	20,6(20,6)	**33,57**(34,3)	59,4(61,7)	111,2(116,5)	214,8(226,1)
7	23,0(22,8)	36,7(37,0)	**64,0**(65,6)	118,7(122,6)	228,1(236,7)
8	25,2(24,9)	39,5(39,6)	67,9(68,9)	124,9(127,6)	238,7(245,1)
9	27,4(26,9)	42,1(41,9)	71,4(71,9)	**130,0**(131,9)	247,2(251,9)
10	29,5(28,8)	44,5(44,1)	74,4(74,6)	134,3(135,5)	254,1(257,5)
11	31,6(30,7)	46,8(46,2)	77,18(77,0)	138,0(138,7)	**259,7**(262,1)
12	33,6(32,6)	48,9(48, 2)	79,7(79, 3)	141,3(141,6)	264,5(266,1)
13	35,6(34,5)	51,1(50,1)	82,1(81,5)	144,3(144,2)	268,5(269,5)
14	37,5(36,3)	53,1(52,0)	84,4(83,5)	146,9(146,5)	272,0(272,5)
15	39,5(38,1)	55,1(53,9)	86,5(85,5)	149,4(148,8)	275,2(275,2)
16	41,4(39,9)	57,1(55,7)	88,7(87,5)	151,7(150,9)	278,0(277,7)

The criterion of completeness

The dependency tests show that the MV2 satisfies the completeness criterion.

During examinations tests in which at each round a certain substitution transformation was performed were carried out. In these tests we found the index of dependence of the degree of completeness on a number of performed rounds. In Fig. B.13 there are charts of values of the index of the degree of completeness at the fixed at each round number of substitution transformation for various types of permutation transformation. We can see from this chart that at the

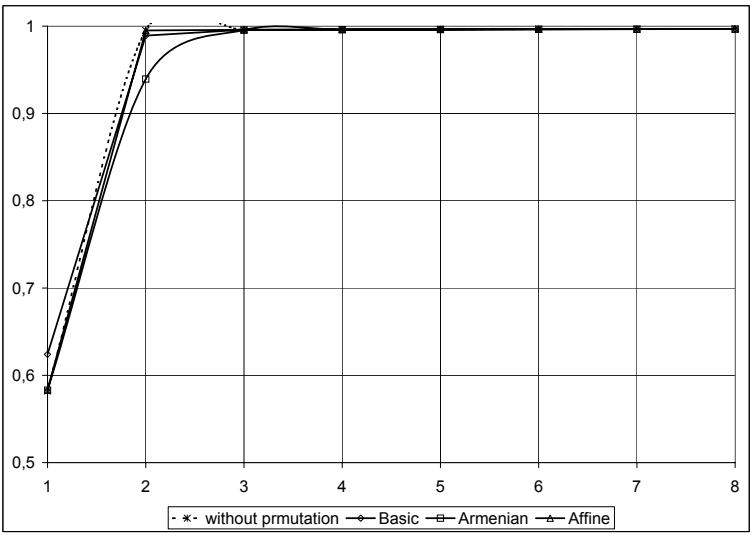

Fig. B.13: *Comparison of degree of completeness for different linear transformations at the fixed set of substitution transformations and 128-byte random inputs*

fixed extract of substitution transformation the criterion of completeness is carried out beginning with the 3d round.

Analysis of cores' output

The core is a harmed ciphertext.

In the table. B.3 and B.4 there are experimental data obtained during testing the core output on accordance to dependence criteria. In these tables the following designations are used: *Rnd* is a number of a round, $\overline{L}_C$ is the maximal length of the core output (in bits), $\widetilde{L}_C$ is the average length of the core output (in bits), Δ – the average number of changed bits, d_c is degrees of completeness, d_a and $\overline{d}_{sa}$ – correspondingly degrees of avalanche and strict avalanche,

computed by (3.86) and (3.87), $\widetilde{d}_{sa}$ is the degree of a strict avalanche, calculated by (3.87), σ is an experimental value of a standard deviation of the core length from the average value and σ/m – evaluation of the error of the degree of a strict avalanche calculated by (3.89).

Evaluations of errors

Normalization factor in the expressions (3.86), (3.88) and (3.87),(3.89) is not used entirely correctly. Let's evaluate an error occurred due to incorrect evaluation of "tails" in (3.89). Denote the average output length through μ. Then from (3.89) we have

$$\widetilde{d}_{sa} = 1 - \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{\mu}\left|\frac{2a_{ij}}{\#\mathcal{X}} - 1\right| - \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=\mu+1}^{\widetilde{m}_i}\left|\frac{2a_{ij}}{\#\mathcal{X}} - 1\right| \tag{B.6}$$

For the component

$$\overline{d_{sa}} = 1 - \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{\mu}\left|\frac{2a_{ij}}{|X|} - 1\right|$$

the value a_{ij} can be considered close to true, and the second one in (B.6) is computed with an error.

Then, the real value will be:

$$d^*_{sa} = \overline{d_{sa}} + \delta,$$

where δ – is an added error.

Let's consider a random value τ, which equals the output length. Let μ be an expectation and σ will be standard deviation of this random value. As $a_{ij} = 0, j > \mu$, then, we can assume that

$$\delta \approx \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=\mu+1}^{\overline{m}_i} 1.$$

Tabl. B.3: *Results of testing dependence criteria for 16-byte inputs*

Rnd	$\overline{L}_C$	$\widetilde{L}_C$	Δ	d_c	d_a	$\overline{d}_{sa}$	$\widetilde{d}_{sa}$	σ	σ/m
1	127	109,9	60,83	1	0,96	0,88	0,95	7,68	0,07
2	119	93,65	53,48	1	0,94	0,86	0,94	8,08	0,09
3	111	81,30	47,55	1	0,92	0,83	0,94	8,11	0,1
4	103	71,94	43,03	1	0,91	0,81	0,93	7,93	0,11
5	95	65	39,20	1	0,9	0,8	0,93	7,74	0,12
6	90	59,61	36,67	1	0,9	0,8	0,92	7,52	0,13
7	85	55,59	34,48	1	0,88	0,78	0,92	7,37	0,13
8	79	52,49	32,72	1	0,89	0,78	0,92	7,16	0,14
9	79	50,27	31,57	1	0,89	0,78	0,92	7,0	0,14
10	75	48,41	30,33	1	0,88	0,77	0,92	6,81	0,14
11	73	47,13	29,73	1	0,89	0,78	0,91	6,7	0,14
12	71	45,92	28,99	1	0,9	0,8	0,92	6,56	0,14
13	71	45,17	28,45	1	0,9	0,79	0,91	6,45	0,14
14	70	44,61	28,15	1	0,89	0,78	0,91	6,39	0,14
15	64	44,16	27,77	1	0,89	0,76	0,91	6,3	0,14
16	63	43,75	27,64	1	0,89	0,75	0,91	6,29	0,14

Tabl. B.4: *Results of testing dependence criteria for 64-byte inputs*

Rnd	$\overline{L}_C$	$\widetilde{L}_C$	Δ	d_c	d_a	$\overline{d}_{sa}$	$\widetilde{d}_{sa}$	σ	σ/m
1	447	400,3	209,2	1	0,97	0,93	0,97	20,6	0,05
2	366	313,8	167,2	1	0,95	0,9	0,96	18,4	0,06
3	303	248,0	134,7	1	0,94	0,89	0,96	16,3	0,07
4	249	198,5	109,7	1	0,93	0,87	0,95	14,7	0,07
5	207	160,8	90,33	1	0,92	0,85	0,95	13,3	0,08
6	178	132,1	75,58	1	0,91	0,84	0,94	11,9	0,09
7	157	110,7	64,2	1	0,9	0,83	0,94	10,9	0,1
8	134	94,06	55,62	1	0,9	0,82	0,93	10,0	0,11
9	119	81,73	48,7	1	0,89	0,81	0,93	9,24	0,11
10	109	72,32	43,47	1	0,89	0,8	0,93	8,64	0,12
11	103	65,06	39,71	1	0,89	0,8	0,92	8,16	0,13
12	94	59,73	36,74	1	0,89	0,8	0,92	7,79	0,13
13	87	55,7	34,57	1	0,88	0,79	0,92	7,5	0,13
14	85	52,46	32,76	1	0,88	0,8	0,92	7,26	0,14
15	79	50,33	31,56	1	0,88	0,78	0,92	7,03	0,14
16	79	48,57	30,44	1	0,87	0,78	0,92	6,86	0,14

Assume, that deviations from the average length by the value which is bigger than standard deviation are unlikely, then $\overline{m}_i - \mu \approx \sigma$ and $\delta \approx \sigma/m$.

From the given tables B.3 and B.4 it follows that the value of degree of a strict avalanche is close to 1. Thus, the MV2 algorithm satisfies dependence criteria.

Statistical testing

Using the battery of Diehard tests, flags of long files have been tested. The tests results allow for interpretation of flags as "random" sequences.

Core length

Deviation of expectation value of the core length from the real one is less than 1 byte for short plaintext and 1% for the long ones.

Flag length

Deviation of expectation value of the flags length from the real one is less than 1 byte for short plaintext and 1% for the long ones.

Performance

During testing we made 100 encryptions of 200000 byte pseudorandom sequence with different keys. The text module has been compiled by MS VC 7.0 compiler. Testing was made

on a PC with a Pentium 4 processor, 1700MHz, RAM 256 MB, 266 MHz DDR, Windows XP. Encryption rate achieved was ≈ 5 MB/sec.

A compiler significantly impacts the algorithm's implementation speed.

Appendix C
Some mathematical facts

The number of combinations from n to m equals

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}.$$

The following agreements are used throughout this work

$$0 \log 0 = 0 \quad \text{and} \quad 0 \log \frac{1}{0} = 0.$$

These agreements are justified as

$$\lim_{x \to 0}(x \log x) = 0 \text{ and } \log(1/x) = -\log x.$$

In [15] there's an identity :

$$\sum_{k=1}^{n} k \cdot x^k = \frac{nx^{n+2} - (n+1)x^{n+1} + x}{(x-1)^2}. \qquad \text{(C.1)}$$

In the present work a special case of this expression is frequently used at $x = 2$:

$$\sum_{k=1}^{n} k \cdot 2^k = (n-1)2^{n+1} + 2. \qquad \text{(C.2)}$$

From (C.2) we get another important equation:

$$\sum_{k=r}^{n} k \cdot 2^k = (n-1)2^{n+1} - (r-2)2^r. \qquad \text{(C.3)}$$

For the real $x \neq 1$ the following identity is true:

$$\sum_{k=1}^{n} k^2 \cdot x^k = \frac{n^2 \cdot x^{n+3} - (2n^2+2n-1) \cdot x^{n+2} + (n+1)^2 \cdot x^{n+1} - x^2 - x}{(x-1)^2}. \qquad \text{(C.4)}$$

The proof (C.4) follows from the identity

$$x\sum_{k=1}^{n} k^2 x^k + 2x\sum_{k=1}^{n} kx^k + \sum_{k=1}^{n} x^k = \sum_{k=1}^{n} k^2 x^k - x + (n+1)^2 \cdot x^{n+1}$$

и (C.1).

From (C.4) for $x = 2$ it follows that:

$$\sum_{k=1}^{n} k^2 \cdot 2^k = (n^2 - 2n + 3) \cdot 2^{n+1} - 6, \qquad \text{(C.5)}$$

And assuming $x = 1/2$, we have:

$$\sum_{k=1}^{n} k^2 \cdot 2^{-k} = 6 - (n^2 + 4n + 6) \cdot 2^{-n}. \qquad \text{(C.6)}$$

From (C.5) directly follows the identity

$$\sum_{k=r}^{n} k^2 \cdot 2^k = (n^2 - 2n + 3) \cdot 2^{n+1} - (r^2 - 4r + 6) \cdot 2^r. \qquad \text{(C.7)}$$

The expressions (C.2) – (C.7) are used in the work during computations of probabilities, expectations and dispersions.

According to [25, p. 57, Theorem 2] the following is true

Theorem C.1 *Let $r_1, r_2, \ldots, r_k$ be a nonnegative integer, and*

$$r_1 + r_2 + \ldots + r_k = n, r_i \geq 0. \tag{C.8}$$

Then, the number of ways by which n elements can be divided into k groups, the first one of which containing exactly r_1 elements, the second one - r_2 elements and so on equals

$$\frac{n!}{r_1.!r_2! \cdot \ldots \cdot r_k!} \tag{C.9}$$

Lemma C.1 *The number of different solutions of the equation C.8 is determined by the following formula*

$$A_{r,n} = \binom{n-r-1}{r} = \binom{n-r-1}{n-1} \tag{C.10}$$

Jensen inequality

Lemma C.2 *(Jensen inequality [14, c. 254]) If $f : (a,b) \to \mathbb{R}$ is a bump (down) function, $x_1, \ldots, x_n$ are the periods of the interval (a,b), $\alpha_1, \ldots, \alpha_n$ are nonnegative numbers such that $\sum_{i=1}^{n} \alpha_i = 1$, then, the following inequality is true*

$$\sum_{i=1}^{n} \alpha_i f(x_i) \geq f\big(\sum_{i=1}^{n} \alpha_i x_i\big). \tag{C.11}$$

Note [14], that if f is a strictly convex function and there are at least two different from 0, among α_i then, the equality (C.11) can take place if and only if $x_1 = \ldots = x_n$.

Appendix D
Table of ASCII codes

Code	00000000	00000001	00000010	00000011
Symbol	NUL	SOH	STX	ETX
Code	00000100	00000101	00000110	00000111
Symbol	EOT	ENQ	ACK	BEL
Code	00001000	00001001	00001010	00001011
Symbol	BS	HT	LF	VT
Code	00001100	00001101	00001110	00001111
Symbol	FF	CR	SO	SI
Code	00010000	00010001	00010010	00010011
Symbol	DLE	DC1	DC2	DC3
Code	00010100	00010101	00010110	00010111
Symbol	DC4	NAK	SYN	ETB

Code	00011000	00011001	00011010	00011011
Symbol	CAN	EM	SUB	ESC
Code	00011100	00011101	00011110	00011111
Symbol	FS	GS	RS	US
Code	00100000	00100001	00100010	00100011
Symbol	SP	!	"	#
Code	00100100	00100101	00100110	00100111
Symbol	$	%	&	'
Code	00101000	00101001	00101010	00101011
Symbol	(	)	*	+
Code	00101100	00101101	00101110	00101111
Symbol	,	-	.	/
Code	00110000	00110001	00110010	00110011
Symbol	0	1	2	3
Code	00110100	00110101	00110110	00110111
Symbol	4	5	6	7
Code	00111000	00111001	00111010	00111011
Symbol	8	9	:	;
Code	00111100	00111101	00111110	00111111
Symbol	<	=	>	?
Code	01000000	01000001	01000010	01000011
Symbol	@	A	B	C
Code	01000100	01000101	01000110	01000111

Symbol	D	E	F	G
Code	01001000	01001001	01001010	01001011
Symbol	H	I	J	K
Code	01001100	01001101	01001110	01001111
Symbol	L	M	N	O
Code	01010000	01010001	01010010	01010011
Symbol	P	Q	R	S
Code	01010100	01010101	01010110	01010111
Symbol	T	U	V	W
Code	01011000	01011001	01011010	01011011
Symbol	X	Y	Z	[
Code	01011100	01011101	01011110	01011111
Symbol	\	]	^	_
Code	01100000	01100001	01100010	01100011
Symbol	`	a	b	c
Code	01100100	01100101	01100110	01100111
Symbol	d	e	f	g
Code	01101000	01101001	01101010	01101011
Symbol	h	i	j	k
Code	01101100	01101101	01101110	01101111
Symbol	l	m	n	o
Code	01110000	01110001	01110010	01110011
Symbol	p	q	r	s

Code	01110100	01110101	01110110	01110111
Symbol	t	u	v	w
Code	01111000	01111001	01111010	01111011
Symbol	x	y	z	{
Code	01111100	01111101	01111110	01111111
Symbol	|	}	~	DEL
Code	10000000	10000001	10000010	10000011
Symbol	Ђ	Ѓ	‚	ѓ
Code	10000100	10000101	10000110	10000111
Symbol	„	…	†	‡
Code	10001000	10001001	10001010	10001011
Symbol	€	‰	Љ	‹
Code	10001100	10001101	10001110	10001111
Symbol	Њ	Ќ	Ћ	Џ
Code	10010000	10010001	10010010	10010011
Symbol	ђ	‘	’	“
Code	10010100	10010101	10010110	10010111
Symbol	”	•	–	—
Code	10011000	10011001	10011010	10011011
Symbol	□	™	љ	›
Code	10011100	10011101	10011110	10011111
Symbol	њ	ќ	ћ	џ
Code	10100000	10100001	10100011	10100010

Symbol		Ў	ў	J
Code	10100100	10100101	10100110	10100111
Symbol	¤	Ґ	¦	§
Code	10101000	10101001	10101010	10101011
Symbol	Ё	©	Є	«
Code	10101100	10101101	10101110	10101111
Symbol	¬	-	®	Ї
Code	10110000	10110001	10110010	10110011
Symbol	°	±	I	i
Code	10110100	10110101	10110110	10110111
Symbol	ґ	µ	¶	·
Code	10111000	10111001	10111010	10111111
Symbol	ё	№	є	ї
Code	11000000	11000001	11000010	11000011
Symbol	А	Б	В	Г
Code	11000100	11000101	11000110	11000111
Symbol	Д	Е	Ж	З
Code	11001000	11001001	11001010	11001011
Symbol	И	Й	К	Л
Code	11001100	11001101	11001110	11001111
Symbol	М	Н	О	П
Code	11010000	11010001	11010010	11010011
Symbol	Р	С	Т	У

Code	11010100	11010101	11010110	11010111
Symbol	Ф	Х	Ц	Ч
Code	11011000	11011001	11011010	11011011
Symbol	Ш	Щ	Ъ	Ы
Code	11011100	11011101	11011110	11011111
Symbol	Ь	Э	Ю	Я
Code	11100000	11100001	11100010	11100011
Symbol	а	б	в	г
Code	11100100	11100101	11100110	11100111
Symbol	д	е	ж	з
Code	11101000	11101001	11101010	11101011
Symbol	и	й	к	л
Code	11101100	11101101	11101110	11101111
Symbol	м	н	о	п
Code	11110000	11110001	11110010	11110011
Symbol	р	с	т	у
Code	11110100	11110101	11110110	11110111
Symbol	ф	х	ц	ч
Code	11111000	11111001	11111010	11111011
Symbol	ш	щ	ъ	ы
Code	11111100	11111101	11111110	11111111
Symbol	ь	э	ю	я

Bibliography

[1] *Алферов А.П., Зубов А.Ю., Кузмин А.С., Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2001.

[2] *Бабаш А.В., Шанкин Г.П.* Криптография. М.: СОЛОН-Р, 2002. С. 511.

[3] Большая советская энциклопедия, Т. 23. М.: Советская энциклопедия, 1976. с. 628.

[4] *Бриллюэн Л.* Наука и теория информациию М. Физматгиз, 1960. С. 392.

[5] *Ватутин А.Е., Виланский Ю.В.* Защита компьютерных данных методом скрытия // Вести Института современных знаний. Специальный выпуск. 2000. №1. С. 46 – 50.

[6] *Виланский Ю.В., Мищенко В.А.* Кодирование информации на основе алгоритма универсального сжатия // Вести Института современных знаний. Специальный выпуск. 2000. №1. С. 36 – 39.

[7] *Виланский Ю.В.* Алгоритм шифрования MV2 – дверь с двумя замками // Вести Института современных знаний. 2001. №2. С. 84–89.

[8] *Виланский Ю.В., Лепин В.В.* Информационные утечки в отображениях с образами различной длины // Весці НАН Беларусі. Сер. фіз.-мат. навук. 2004. №3. С. 47 – 53.

[9] *Виланский Ю.В., Лепин В.В., Мищенко В.А.* Двухканальный алгоритм шифрования MV2 // Вести Института современных знаний. №3-4. 2003. С. 113 – 121.

[10] *Виланский Ю.В., Лепин В.В., Мищенко В.А.* Двухканальный алгоритм шифрования MV2 (продолжение) // Вести Института современных знаний. 2004. №1. С. 77 – 88.

[11] *Галлагер Р.* Теория информации и надежная связь. М. Связь. 1974.

[12] *Грехем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики. М.: Мир, 1998.

[13] *Грушо А.А., Тимонина Е.Е., Применко Э.А.* Анализ и синтез криптоалгоритмов. Йошкар-Ола: Марийский филиал Московского открытого социального университета. 2000. С. 110.

[14] *Зорич В.А.* Математический анализ. Ч. 1. М.: Наука. 1981. с. 254

[15] *Кнут Дональд Э.* Искусство программирования Т.1. Основные алгоритмы. "Вильямс". М. – СПб. – Киев, 2000. С. 712.

[16] *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. М.: МЦНМО, 2002.

[17] *Месси Дж. Л.* Введение в современную криптологию // М.: ТИИЭР. 1988, Т. 76. №5. С. 24-42.

[18] *Мищенко В.А.* Криптография и массовые технологии современного рынка. // Успехи современного естествознания. 2004. №5. Приложение №1. С. 146 – 149.

[19] *Мищенко В.А., Захаров В.В., Виланский Ю.В.* // Патент Евразийский № 004904, МКИ H04L9/06. Способ шифрования, передачи, хранения конфиденциальных сообщений и система для осуществления способа // Заявка № 200200467; Заявлено 15.10.1999; Дата выдачи 26.08.2004. – С. 90.

[20] *Мищенко В.А., Захаров В.В., Виланский Ю.В., Вержбалович Д.И.* Способ шифрования и дешифрования и устройство для его осуществления // Евразийский патент № 003679, 2003.

[21] *Молдовян А.А, Молдовян Н.А, Гуц Н.Д., Изотов Б.В.* Криптография: Скоростные шифры. СПб., БХВ-Петербург, 2002.

[22] *Молдовян А.А., Молдовян Н.А., Советов Б.Я.* Крипография. СПб.: Издательство Лань, 2000. С. 224.

[23] *Сачков В.Н.* Введение в комбинаторные методы дискретной математики. М. Наука. 1982 г.

[24] Труды конференции "Математика и безопасность информационных технологий" МаБИТ-04, Москва, 28-29 октября 2004 г.

[25] *Феллер В.* Введение в теорию вероятностей и ее приложения. Т.1. М.: Мир. 1984.

[26] *Фомичев В.М.*, "Дискретная математика и криптология". М.: ДИАЛОГ-МИФИ, 2003.

[27] *Харин Ю.С., Берник В.И., Матвеев Г.В.* Математические основы криптологии. Минск.: БГУ, 1999. С. 320.

[28] *Шеннон К.* Работы по теории информации и кибернетике. М. ИЛ. 1963. С. 333–369.

[29] *Шеннон К.* Теория связи в секретных системах. В кн. Работы по теории информации и кибернетики. М.: ИЛ, 1963.

[30] *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002.

[31] *Adams C.M.* On Immunity Against Biham and Shamir's Differential Cryptanalysis // Information Processing Letters. V. 41, 14 Feb 1992, P. 77–80.

[32] *Adams C.M. and Tavares S.E.* The Structured Design of Cryptographically Good SBoxes. // Journal of Cryptology 1990. V.3, N.1. P. 27 –41.

[33] *Adams C.M. and Tavares S.E.* Designing S-Boxes for Ciphers Resistant to Differential Cryptanalysis. // Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography Rome, Italy. 15-16 Feb 1993. P. 181–190.

[34] *Andersen R., Biham E. and Knudsen L.* Serpent: A Proposal for Advanced Encryption Standard // First Advanced Encryption Standard (AES) Conference, Ventura, (CA), 1998.

[35] *Beauchemin P. and Brassard G.* A Generalization of Hellman's Extension to Shannon's Approach to Cryptography // Journal of Cryptology. 1988. V.1, N.2. P. 129-132.

[36] *Bellare M., Goldwasser S., Micciancio D.* "Pseudo-Random" Number Generation within Cryptographic Algorithms: the DSS Case // Advances in Cryptology – Crypto 97 Proceedings, Lecture Notes in Computer Science, V. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[37] *Den Boer B. and Bosselaers A.* Collisions for the Compression Function of MD5 // Advances in Cryptology EUROCRYPT 93 Proceedings. Springer-Verlag. 1994. P. 293

[38] *Chaum D.* Online cash cheks // Proc. EUROCRYPT'89, Lect. Notes in Comput. Sci., 1990. V. 434. P. 288 – 293.

[39] *Chaum D.* Blind Signature Systems // US Patent #4759063, 1988.

[40] *Cover T.M. and King R.C.*, A Convergent Gambling Estimate of the Entropy of English // IEEE Transactions on Information Theory, V. IT-24, N.4 Jul 1978, P. 329 – 421.

[41] . *Davies D.W. and Price W.L.* The Application of Digital Signatures Based on Public-Key Cryptosystems // Proceedings of the Fifth International Computer Communications Conference, Oct 1980. P. 525 – 530.

[42] *Dawson M.H. and Tavares S.E.* An Expanded Set of S-Box Design Criteria Based on Information Theory and

Its Relation to Differential-like Attacks // Advances in Cryptology EUROCRYPT '91 Proceedings, Springer-Verlag, 1991. P. 352–367.

[43] *Dawson M.H. and Tavares S.E.*, An Expanded Set of Design Criteria for Substitution Boxes and Their Use in Strengthening DES-Like Cryptosystems // IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, Victoria, BC, Canada, 9–10 May 1991. P. 191 –195.

[44] *Deavours C.A.* Unicity Points in Cryptanalysis // Cryptologia, 1977. V.1, N.1. P. 46 – 68.

[45] *Desmedt Yvo.* Some Research Aspects of Threshold Cryptography // Information Security First International Workshop, ISW'97, Tatsunokuchi, Ishikawa Japan, September 17–19, 1997, Proceedings Series: Lecture Notes in Computer Science, V. 1396 Okamoto, Eiji; Davida, George; Mambo, Masahiro (Eds.) 1998, XII. P. 357. Softcover.

[46] *Diffie W. and Hellman M.E.* Multiuser Cryptographic Techniques // Proceedings of AFIPS National Computer Conference. 1976. P. 109–112.

[47] *Diffie W. and Hellman M.E.* New Directions in Cryptography // IEEE Transactions on Information Theory. V. IT-22, N.6. Nov 1976. P. 644 –654.

[48] *Feistel H.* Cryptography and Computer Privacy // Scientific American, May 1973. V. 228, N.5. P. 15 –23.

[49] *Fiat A., Shamir A.* How to prove yourself: practical solutions to identification and signature problems //

Proc. Crypto'86, Lect. Notes in Comput. Sci. V. 263. 1987. P. 186 – 194.

[50] *Forre R.* Methods and Instruments for Designing S-boxes // Journal of Cryptology. 1990. V.2, N.3. P.115 – 130.

[51] *Hellman M.E.* An Extension of the Shannon Theory Approach to Cryptography // IEEE Transactions on Information Theory. V.IT-23, N.3. May 1977. P. 289– 294.

[52] *Heys H.M. and Tavares S.E.* The Design of Substitution-Permutation Networks Resistant to Differential and Linear Cryptanalysis // Proceedings of the 2nd Annual ACM Conference on Computer and Communications Security, ACM Press. 1994. P. 148–155.

[53] *Huffman D.A.* A method for the construction of minimum redundancy codes. // In Proceedings of the Institute of Electrical and Radio Engineers 40. 9(Sept.). 1952. P. 1098 –1101.

[54] *Impagliazzo R., Luby M.* One-way functions are essential for complexity based cryptography // Proc. 30th Annu. Symp. On Found. Of Comput. Sci. 1989. P. 230–235.

[55] *K. Kim.* A Stady of the constraction and Analysis of substetution Boxes for Symmetric Cryptosystems: Dissert. ... Doct. Ph., Yokohama National Univeristy. Division of Electrical and Computer Engineering. 1990.

[56] *K. Kim* Construction of DES-Like S-Boxes Based on Boolean Functions Satisfying the SAC // Advances

in Cryptology, ASIACRYPT 91 Proceedings. Springer-Verlag. 1993. P. 59–72.

[57] *Lenstra A.K., Verheul E.R.* Selecting Cryptographic Key Sizes // Journal of Cryptology. 2001. V. 14, N. 4. P. 255 –293.

[58] *Kothari S.C.* Generalized Linear Threshold Scheme // Advances in Cryptology: Proceedings of CRYPTO 84. Springer-Verlag. 1985. P. 231 –241. p. 884.

[59] *Lloyd S.* Counting functions satisfying A Higher order strict avalanche criterion // EUROCRYPT 1989: P. 63 – 74

[60] *Massey J.L.* On the Optimality of SAFER+ Diffusion // Proceedings of the Second AES Candidate Conference. NIST. Mar. 1999.

[61] *Massey J.L.* Some applications of source coding in cryptography // European Trans. On Telecomm. V. 5. P. 421-429. 1994.

[62] *Maurer U.L., Massey J.L.* Casade Ciphers: The Importance of Being First. // Jornal of Cryptology. 1993. V.6, N. 1. P. 55 –61.

[63] *Memnezes A., Van Oorshot P., Vanstone S.* Handbook of applied cryptography. CRC Press, 1996.

[64] *Merkle R.C.*, Secure Communication Over Insecure Channels // Communications of the ACM. 1978. V. 21, N. 4. P. 294 –299.

[65] *Micali S.* Fair Public-Key Cryptosystems //Advances in Cryptology -– CRYPTO'92 Procveedings. Springer-Vertag. 1993. P.113 – 138.

[66] *Micali S.* Fair Cryptosystems, MIT/LCS/TR-579.b, MIT Laboratory for Computer Science, Nov. 1993.

[67] *Micali S.* Fair Cryptosystems and Methods for Use, U.S.Patent #5,276,737, 4 Jan 1994.

[68] *Micali S.* Fair Cryptosystems and Methods for Use, U.S.Patent #5,315,658, 24 May 1994.

[69] *Mischenko V.A.* Method for authorized displaying information distributed through public communication media // International Application Number: PCT/BY01/00013. International ublication Number: WO 03/017566. International Publication Date: 27.02.2003. Int. Filing Date: 20.08.2001.

[70] *Michtchenko V.A.* Method of Encryption, Transmitting and Decrypting of Information in Public Networks // Application WO 02/091667 A1 14 November 2002.

[71] *Mischenko V.A., Zakharau U.U., Vilansky Y.V., Verzhbalovich D.I.* Method for encrypting information and device for realization of the method. International Application Number: PCT/BY99/00005. International Publication Number: WO 00/65767. International Publication Date: 02 November 2000. Int. Filing Date: 16 Mart 1999. http://pctgazette.wipo.int/

[72] *Mischenko V.A., Zakharau U.U., Vilansky Y.V.* Methods for encoding, decoding, transferring, storage and control of information, systems for carrying out the methods // International Application Number: PCT/BY99/00008. International Publication Number: WO 01/30017. International Publication Date: 26 April 2001. Int. Filing Date: 15 October 1999. http://pctgazette.wipo.int/. 105 c. (35 c.)

[73] National Institute of Standards and Technology, NIST FIPS PVB 185 Escrowed encryption standart. U.S. Department of Comerce, Feb. 1994.

[74] National Institute of Standards and Technology, NIST FIPS PUB 186, Digital Signature Standard. U.S. Department of Commerce, May 1994.

[75] National Institute of Standards and Technology, NIST FIPS PUB 197, Advanced Encryption Standard. U.S. Department of Commerce. 2001.

[76] *Nyberg K.* Perfect Nonlinear S-Boxes. // Advances in Cryptology EUROCRYPT '91 Proceedings. Springcr-Verlag. 1991. P. 378 –386.

[77] *Nyberg K.* On the Construction of Highly Nonlinear Permutations. // Advances in Cryptology EUROCRYPT '92 Proceedings, Springer-Verlag. 1991. P. 92 –98.

[78] *O'Connor L.* On the Distribution of Characteristics in Bijective Mappings // Advances in Cryptology EUROCRYPT '93 Proceedings, Springer-Verlag. 1994. P. 360 –370.

[79] *O'Connor L.* On the Distribution of Characteristics in Composite Permutations // Advances in Cryptology CRYPTO '93 Proceedings. Springer-Verlag. 1994. P. 403-412.

[80] *O'Connor L. and Klapper A.*, Algebraic Nonlinearity and Its Application to Cryptography. // Journal of Cryptology. 1994. V. 7, N.3. P. 133 –151.

[81] *B. Preneel* Analysis and Design of Cryptographic Hash Functions // Ph. D. dissertation, Katholieke Universiteit Leuven. Jan 1993.

[82] *B. Preneel, A. Bosselaers, V. Rijmen, B. Van Rompay L. Granboulan, J. Stern, S. Murphy, M. Dichtl, P. Serf E. Biham, O. Dunkelman, V. Furman F. Koeune, G. Piret, J-J. Quisquater, L. Knudsen, H. Raddum* Comments by the NESSIE Project on the AES Finalists. [Electronic resource]. 2000. Mode of access: http:// csrc.nist.gov/ CryptoToolkit/ aes/round2/comments/20000524-bpreneel.pdf

[83] *Preneel B., Van Leekwijck W., Van Linden L., Govaerts R., Vandewalle J.,* Propagation Characteristics of Boolean Functions // Advances in Cryptology EUROCRYPT '90 Proceedings. Springer-Verlag. 1991. P. 161 –173.

[84] *Rissanen J.J. and Langdon G.G.*, Universal modeling and coding. // IEEE Trans. Inf. Theory IT-27, 1(Jan.). 1981. P. 12 –23.

[85] *Rivest R.L.* All-Or-Nothing Encryption and The Package Transform // Fast Software Encryption 1997. LNCS 1267. Springer-Verlag. 1997. P. 210 –218.

[86] *Rivest R.L.* The MD5 Message Digest Algorithm. RFC 1321. 1992.

[87] *Robshaw M.J.B.* MD2, MD4, MD5, SHA, and Other Hash Functions // Technical Report TR-101, Version 3.0, RSA Laboratories, Jul 1994.

[88] *Robshaw M.J.B.* Implementations of the Search for Pseudo-Collisions in MD5 // Technical Report TR-103, Version 2.0, RSA Laboratories, Nov 1993.

[89] *Robshaw M.J.B.* On Pseudo-Collisions in MD5 Technical Report TR-102, Version 1.1, RSA Laboratories, Jul 1994.

[90] *Schnorr C.P.*, Efficient identification and signatures for smart cards // Proc. Crypto'89, Lect. Notes in Comput. Sci. V. 435. 1990. P. 239-252.

[91] *Shamir A.* How to Share a Secret // Communications of the ACM. V. 24, N. 11. Nov 1979. P. 612 –613.

[92] *Shannon C.E.* A mathematical theory of communication // The Bell System Technical Journal. 1948. V. 27. P. 379 –423.

[93] *Shannon C.E.* Communication theory of secrecy systems // The Bell System Technical Journal, 28 (1949), P. 656 –715.

[94] . *Simmons G.J.* The Prisoner's Problem and the Subliminal Channel // Advances in Cryptology: Proceedings of CRYPTO'83. Plenum Press. 1984. P. 51 – 67.

[95] *Simmons G.J.* The Subliminal Channel and Digital Signatures // Proceedings of EUROCRYPT 84. Springer-Verlag. 1985. P. 364 – 378.

[96] *Simmons G.J.* A Secure Subliminal Channel(?) // Advances in Cryptology – CRYPTO'85 Proceedings. Springer-Verlag. 1986. P. 33 – 41.

[97] *Sivabalan M., Tavares S.E., and Peppard L.E.* On the Design of SP Networks from an Information Theoretic Point of View. // Advances in Cryptology, Proceedings of Crypto'92, Springer-Verlag, 1993. P. 260 – 279.

[98] *Webster A.F. and Tavares S.E.* On the design of S-boxes // Advances in Cryptology, Proceedings CRYPTO85, Spinger-Verlag, Heidelberg, 1986. P. 523 –534.

[99] *Youssef A.M. and Tavares S.E.* Information Leakage of Randomly Selected Functions // Proceedings of the 4th Canadian Workshop On Information Theory. May 28, 1995. Lecture Notes in Computer Science (LNCS 1133). Springer-Verlag. 1996. P. 41 –52.

[100] *Zhang M., Tavares S.E., and Campbell L.L.* Information Leakage of Boolean unctions and its Ralationship to Other Cryptographic Criteria // Proceedings of 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, 1994. P. 156 – 165.

[101] *Ziv J. and Lempel A.* A universal algorithm for sequential data compression. // IEEE Trans. Inf. Theory IT-23,3, 3 (May). 1977. P. 337-343.

Contents

Introduction **5**

Chapter 1. Harmed texts **13**
1.1 Meaning of texts and information theory . . . 13
1.2 Cipher attacks. The concept of harmed texts . 16
1.3 The concept of harmed texts 18
1.4 Cryptographically harmed texts and multichannel cryptography 28
1.5 Concealment and encryption of harmed texts . 39
1.6 Harmed texts and data-cores. Unicity distance for harmed ciphertexts 47
1.7 Nine structural encryption schemes based on harmed texts 49
1.8 Summary 60

Multi-channel cryptography **61**
2.1 Two-channel symmetric encryption algorithm MV2 (first familiarity) 61
2.2 Multi-channel cryptographic transformations . 62
2.2.1 Variant of little harm 65
2.2.2 Variant of repeated use of the MV2 algorithm to the text HC_1 66
2.2.3 Variant of repeated use of the MV2 algorithm to the united text $\{HC_i\}$ . . 66

2.3 Real symmetric-asymmetric system: MV2 and asymmetric system with an open key 67
2.4 Three-channel symmetric-asymmetric MV3 system and a system with an open key 70
2.5 Combined cipher: MV2 and a stream cipher . 72
2.6 Concealed channel of information transmission. Information transmission with the help of MV2 algorithm keys. 74
2.7 Misleading with the help of the MV2 algorithm 76
2.8 Multi-channel quantum cryptography 77
2.9 Summary 84

Chapter 3. Universal mechanism of harming 86
3.1 Substitution transformation for obtaining harmed texts 86
3.1.1 Mappings with variable length of an image 86
3.1.2 Definition of the MV2-transformation . 89
3.1.3 Information and statistical estimations for an MV2-transformation 94
3.1.4 MV2-transformation for obtaining harmed texts 102
3.1.5 Composition of two MV2-transformations 114
3.2 A general scheme of harming 116
3.2.1 The device for harming 116
3.2.2 A method of harming based on MV2-transformation 119
3.2.3 Preliminary analysis of the general scheme which uses MV2-transformations . . . 122
3.2.4 Examination of the general scheme . . 125
3.3 Two channel encryption algorithm MV2 . . . 130
3.4 Shannon's security model for two-channel ciphers 133

3.4.1 Shannon's security model 133
3.4.2 General information ratios for two-channel systems 138
3.5 Analysis of security of using a universal algorithm of harming 140
3.5.1 Analysis of the general scheme security at unknown flag output 140
3.5.2 Analysis of the general scheme security at unknown core output 146
3.6 Usage modes of two-channel systems 152
3.7 Statistical testing 153
3.7.1 Dependence criteria 153
3.7.2 Dependence criteria for substitution transformations with a variable length output 155
3.8 Summary 158

Chapter 4. Protocols and multi-channel cryptography 160
4.1 Concept of protocol. Language of protocols . . 160
4.2 Authentication of a communication participant 161
4.3 Message authentication (digital signature) . . 166
4.4 Non-accountability. Electronic money 171
4.5 The problem of key deposition and the MV2 algorithm 175
4.6 Summary 179

Chapter 5. Mass technologies and multi-channel cryptography 181
5.1 Concept of mass technologies 181
5.2 Telecommunications 188
5.3 Information storage 190
5.4 Trade 193
5.5 Protection of documents on paper carriers . . 198

5.6 Protection of corporeal property 199
5.7 Protection of intellectual property in multimedia 201
5.8 Summary 202

Conclusion **203**

Appendix A. Terminology and basic definitions **205**

Appendix B. Basic implementation of the MV2 algorithm **216**
B.1 Description 216
B.2 Statistical estimations of output data and resistance 221
B.3 Testing of the algorithm 228

Appendix C. Some mathematical facts **247**

Appendix D. Table of ASCII codes **250**

Bibliography **256**