International Journal of Soft Computing 10 (6): 383-390, 2015
ISSN: 1816-9503

Use of Universal Coding with Binary Data Thirds for Information Compression and its Security

V.A. Shurigin, V.V. Makarov, A.B. Vavrenyuk and A.V. Starikovskiy
National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute),
Kashirskoe Highway 31, 115409 Moscow, Russian Federation

Abstract: The following study discusses the problem of linking data compression function and information security. There is a definition for coding and its modification, mathematical apparatus for its realization is presented including code words numeration, diagrams of superabundance decrease when increasing source blocks length are shown, five groups of coding parameters are shown (which population is a key) without which decoding is impossible.

Key words: Binary data compression, binary data thirds coding, universal coding, information security, blocks

INTRODUCTION

Information encryption is a process of signal modification from the form convenient for information usage into the form convenient for transfer, storage and automatic processing.

According to the information theory, universal coding is a coding which removes superabundance in the source data stream without knowing the statical characteristics of the source by increasing blocks length. The source sequenceis divided into these blocks. The following coding is called asymptomatically optimal because its superabundance goes to zero when source blocks length goes to infinity.

In the present article, the case of specific Universal Coding Method is presented binary data Thirds Coding (TC) (Alexandrovich *et al*., 2011; Kozlov *et al*., 2011). The goal is to link data compression function and information protection.

This question is discussed in many scientific study, e.g., Mohamed *et al*. (2014) propose a compression and encryption technique on securing Trivial File Transfer Protocol packet. Haleem *et al*. (2006) present a joint distributed data compression and encryption scheme suitable for wireless sensor networks. By Lv and Zhao (2007), a way to increase the information security through integration of data compression and cryptographyis explained. These discussions and many others have advantages and disadvantages.

MATERIALS AND METHODS

Let us move to goal setting. It should be assumed that the source initiates the sequence of statistically independent symbols "1" and "0" with unknown to us p and q possibilities. Let us divide the sequence of binary symbols, initiated by the source into blocks with n length (n-blocks). The following value will be taken as superabundance on source sequence symbol:

$$R_n(p) = n_{av}/n\text{-}H(p) \quad (1)$$

Where:
n_{av} = Average code word length
H (p) = Source entropy

Coding quality will be determined by R_n value-coding superabundance:

$$R_n = \text{Sup}\, R_n(p) \quad (2)$$

where, $0<p<1$. The following code is universal, if the following condition is fulfilled:

$$\lim_{n\to\infty} R_n = 0 \quad (3)$$

Let us move to the coding definition: examine the multitude N which elements are all possible binary digits with n capacity, 2^n elements on the whole. In Fig. 1, there are multitude N elements for n = 5 in ascending order.

Corresponding Author: V.A. Shurigin, National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute), Kashirskoe Highway 31, 115409 Moscow, Russian Federation

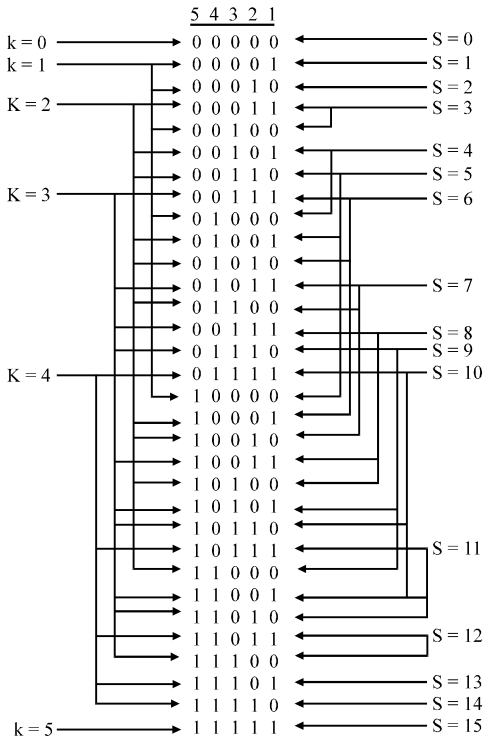

Fig. 1: Multitude N elements for n = 5 in ascending order

Divide N into sub-multitudes and mark them with M_k, L_s. K_κ elements are all N multitude elements which have k units ($0 \le k \le n$). L_s elements are all N multitude elements which units positions sum is equal to S ($0 \le S \le n(n+1)/2$). In Fig. 1 sub-multitudes for n = 5 are situated.

Determine $R_{k,s}$ as crossing of M_k, L_s. Number of $R_{k,s}$ multitude elements will be marked as r (n, k, s). For each $R_{k,s}$ multitude element a number b (n, k, s) will be put where $0 \le b(n,k,s) \le r(n,k,s)-1$.

Determine the code word w corresponding to n-block, like arranged third of binary sets (k, s, b (n, k, s)). Code word length (in bits) will be equal to:

$$L =]\log_2(n+1)\,[+]\log_2(n(n+1)/2+1[+]\log_2 r(n,k,s)[\quad (4)$$

The code will be determined as W multitude where all w code words are is elements. According to the scheme, W is a prefix multitude, therefore, biunique correspondence exists between n-blocks and w code words. The following coding will be determined as TC.

Theorem 1 (TC is universal for bernoulli sources)

Proof: Let us set $\eta(n, k, s)$ the possibility of n-block appearing with k units and with units position number sum equal to S.

$$\eta(n,k,s) = r(n,k,s)\,p$$

Let us write down the formula for code word average length-n_{av}:

$$n_{av} =]\log_2(n+1)\,[+]\log_2(n(n+1)/2+1[+ \sum_{K=0}^{n}\sum_{s=s_{min(k)}}^{S_{max(k)}}\eta(n,k,s)]\log_2 r(n,k,s)[\quad (5)$$

$$S_{min(k)} = k(k+1)/2;\ S_{max(k)} = kn-k(k+1)/2 \quad (6)$$

Further:

$$n_{av} \le \log_2(n+1) + \log_2(n(n+1)/2+1) + \sum_K\sum_s \eta(n,k,s)\log_2 r(n,k,s) + 2 \quad (7)$$

Taking into consideration that:

$$n(n+1)/2+1 = (n^2+n+2)/2 \le$$
$$(n^2+2n+1)/2 = (n+1)2/2 \text{ for every } n$$

Then:

$$n_{av} \le 3\log_2(n+1) + \sum_K\sum_s \eta(n,k,s)\log_2 r(n,k,s) + 2$$

Let us make some identical transformations:

$$\sum\sum\eta(n,k,s)\log_2 r(n,k,s) = \sum_K\sum_s\eta(n,k,s)\log_2\eta(n,k,s) - \sum_K\sum_s\eta(n,k,s)\log_2 p^k q^{n-k}$$
$$= H' - \sum_K\sum_s\eta(n,k,s)\,k\log_2 p - \sum_K\sum_s\eta(n,k,s)(n-k)\log_2 q$$

From the combinatory concepts, it follows that (Fig. 1):

$$\sum_s r(n,k,s) = C^k{}_n$$

Then:

$$\sum_K\sum_s\eta(n,k,s)\,k\log_2 p = \sum_K C^k{}_n p^k q^{n-k}\,k\log_2 p \le n\log_2 p$$

By a similar way:

$$\sum_K\sum_s\eta(n,k,s)(n-k)\log_2 q = \sum_K C^k{}_n p^k q^{n-k}(n-k)\log_2 q \le n\log_2 q$$

Therefore:

$$n_{av} \le 3\log_2(n+1) + H' + nH(p) + 2$$
$$R_n(p) = n_{av}/n - H(p) \le (3\log_2(n+1))/n + H'/n + H(p) + 2/n - H(p)$$

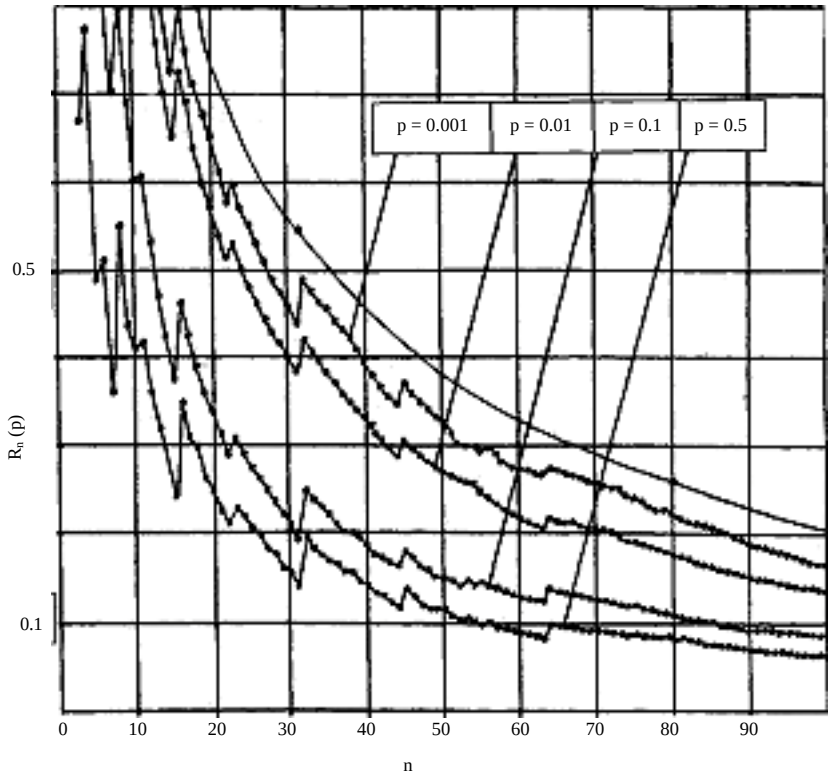

Fig. 2: Coding superabundance dependence

As:

$$H' = 0, R_n(p) \leq (3\log_2 (n+1))/n + 2/n \quad (8)$$

It follows that: $\lim_{n\to\infty} R_n = 0$. The theorem is proved. In Fig. 2, there are diagrams which illustrate the set coding method features. Coding superabundance values were calculated on the computer according to Eq. 1-7 accurate to three symbols. The upper superabundance board is shown, calculated according to Eq. 8. Calculations were made for four values of P unit appearance possibility.

The p-values were chosen to have the possibility of seeing coding superabundance behavior for different sources. In case p = 0.5, there is no superabundance in binary streams, therefore usage of coding for source sequence will result in transferring data volume increase. In case p = 0.001, there is one unit for every 1000 bit of the source sequence, i.e., there is a superabundance. Calculations showed that if we use p = 0 in a code word of middle length (Eq. 7) then superabundance value will differ from the cause with p = 0.001 in the fourth symbol from the dot. Therefore, for each p = 0.001 superabundance value (accurate to three symbols) will be equal to superabundance with p = 0.001 and in this case p = 0.001 is the best for coding usage. Causes with p = 0.1 and p = 0.01 are intermediate.

The necessity to study different causes is caused by the primary proposal about prior uncertainty of units' appearance possibility in the source binary sequence.

Because R_n (p) is a discreet function, determined on the multitude of natural numbers, diagrams are separate dots corresponding to whole n. In Fig. 2, they are connected with direct lines, it is made for better visualization.

It is obvious that for set n superabundance is increasing the more p differs from 0.5. Superabundance goes to zero when n grows but not monotonely and not proportionally through P.

According to coding definition its symmetry against p arises means superabundance curves for p will be equal to 1-P curves. In other words, the source sequence with p = 0.1 will be compressed in the same amount by using coding like when p = 0.9 when p = 0.01, it will be the same for p = 0.99, etc.

Let us return to formula for code word length (Eq. 4). The second summand in this formula determines the number of units needed for S parameter record. Provided that this number is permanent for each n definition and is calculated for S maximum value.

On the other side, for K fixed parameter there are set values of $s_{min(k)}$ and $s_{max(k)}$, calculated with Eq. 6. Because there is a permanent (for the set n value) number of digits

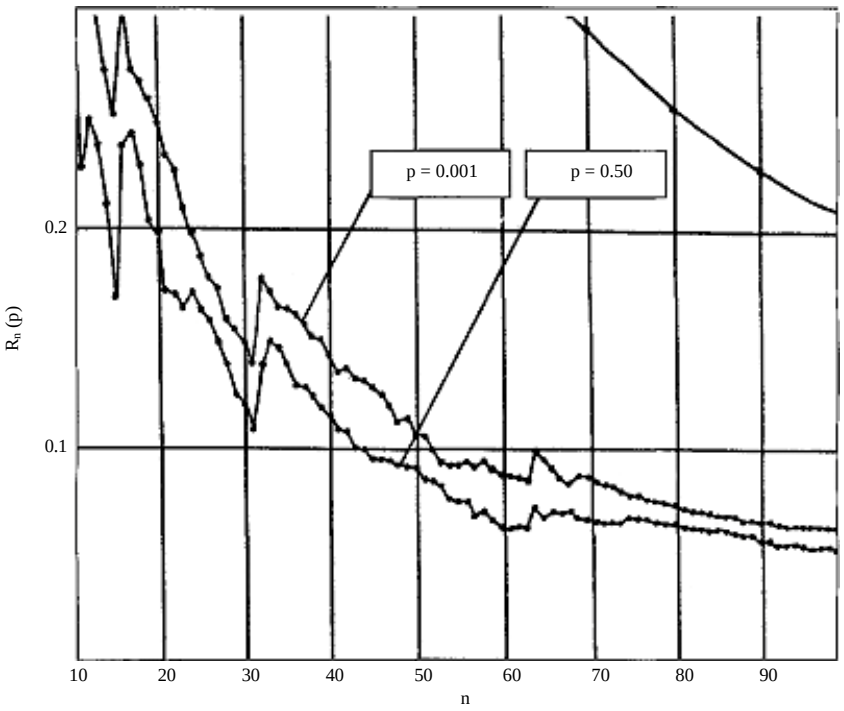

Fig. 3: Superabundance dependence from TC modified method

(the first summand in the Eq. 4) for k record, k is a prefix, i.e., the first parameter determined while coding. However, when K is determined, the corresponding S values can be calculated with Eq. 6, means the second summand in (Eq. 4) can be changed into the equation:

$$]\log_2 (s_{max} (k)-s_{min} (k)+1)[$$

which helps to reduce L code word length. Equation for L will be:

$$L =]\log_2 (n+1)[+]\log_2 (\kappa(n-\kappa)+1)[+]\log_2 r (n, k, s)[\quad (9)$$

n_{av} equation will also change:

$$n_{av} =][\log_2(n+1)[+\sum_{K=0}^{n} C^k{}_n p^k q^{n-k}]\log_2(\kappa(n-\kappa)+1)[+$$

$$\sum_{K=0}^{ns_{max}(k)} \sum_{s=s_{min}(k)} \eta (n, k, s)]\log_2 r(n, k, s)[\quad (10)$$

In Fig. 3, the corresponding TC modified method dependences are presented: by comparing these diagrams with the diagrams in Fig. 2 it can be stated that superabundance values for fix n have decreased 2 times. Moreover, R_n (p) going to 0 for different p become more proportional which means R_n (p) values "corridor" for different p has narrowed. Calculations have shown that superabundance values for p = 0.5; 0.1 are equal till the third symbol after the dot. For p = 0.01 difference from the case with p = 0.5 is less 0.004. That is why curves for p = 0.1 and 0.01 in the figure are not shown. For p = 0.001, beginning with n = 75 this definition does not exceed 0.008.

Hence, TC modification saves not only code word digits but more smooth (accurate to the third symbol after dot) going to zero for R_n (p) for different p values.

RESULTS

To realize TC and to estimate its effectiveness and labor intensity, it is necessary to calculate r(n, k, s) coefficients values. During combinatory literature analysis there was no information on coefficients that have the same meaning.

Two closest results to our case can be underlined below. In Vilenkin (1969), there is a recurrent relation for the number of partition of a natural number into unequal summands. Each summand is less than some n digit. In addition, there is also a generating function for such partition but the number of summands is not limited. In Special functions reference book (1979), there is an obvious formula for a number of partitions but without limitation for summands values.

According to this theorems proved and correlations for calculating r(n, k, s) coefficients are written and proved which are a mathematical apparatus for coding practical realization. Proofs and intermediate formulas are not shown while developing the formulas because of their awkwardness.

Theorem 2: For r(n, k, s) there is a recurrence equation:

$$r(n, k, s) = r(n-1, k, s) + r(n-1, k-1, s-n) \quad (11)$$

Result 1:

$$r(n, k, s) = \sum_{j=0}^{n-1} r(j, k-1, s-(j+1)) + r(0, k, s) \quad (12)$$

Result 2:

$$r(n, k, s) = \sum_{j=0}^{k} r(n-(j+1), k-j, s-jn+j(j+1)/2) \quad (13)$$

Theorem 3: For each n, k, s the following is correct:

$$r(n, k, s) = r(n, n-k, n(n+1)/2-s) \quad (14)$$

Theorem 4: For each n, k, s the following is correct:

$$r(n, k, s) = r(n, k, k(n+1)-s) \quad (15)$$

Theorem 5: For even values of nand each k, s the following is correct:

$$r(n, k, s) = \sum_{K1} \sum_{s1} r(n/2, k1, s1)\, r(n/2, k-k1, s-s1-(k-k1)n/2) \quad (16)$$

where, $0 \le k1 \le k$, $s1_{min} \le s1 \le s1_{max}$, wherein:

$$s1_{min} = \max\{k1(k1+1)/2, s-(k-k1)(2n-(k-k1-1))/2\}$$
$$s1_{max} = \min\{nk1/2-k1(k1-1)/2, s-(k-k1)(n+k-k1+1)/2\}$$

Comment: For odd n and each non-equal partition of n-block into (n1, n2), the formula is transformed into the Eq. 16:

$$r(n, k, s) = \sum_{K1} \sum_{s1} r(n1, k1, s1)\, r(n2, k-k1, s-s1-(k-k1)n1) \quad (17)$$

In the formula for $s1_{max}$ n/2 is changed into n1 and n into 2n1.

Theorem 6: For each n, k, s the following is correct:

$$r(n, k, s) = \sum_{i=1}^{n-(k-1)} r(n-i, k-1, s-ik) \quad (18)$$

Result 3: For each r(n, k, s) the following recurrence equation is used:

$$r(n, k, s) = r(n-1, k-1, s-k) + r(n-1, k, s-k) \quad (19)$$

During the work the explicit expressions for r(n, k, s) were tried to be derived. It turned out that in general without using special functions these expressions are impossible to be derived. In case Gaussian function is used, expressions turn out to be too hard and impractical, means they have no practical interest.

Nevertheless, it was possible to derive explicit expressions for the specific cases (k = 2, k = 3, k = 4). They can be used for calculating the source values r(n, k, s) with the following usage of recurrence equations:

$$r(n, 2, s) = \min\{s-2; n-1\} - \max\{\,](s-1)/2]; 1\} + 1 \quad (20)$$

$$r(n, 3, s) = (2]s/3] + 1-s)x + x^2 + (s-]s/3[-5/2)\xi - \xi^2/2 - y(4](s-]s/3[)/2[-y-\varphi-7)/4-\psi \quad (21)$$

Where:

$$x = [s/2-]s/3[],\ y = s-]s/3[-4$$

$$\psi = \begin{cases} y-1-\left[\dfrac{(y-1)}{2}\right] \\ 0 \end{cases}$$

if (s-]s/3[)/2) is integral; for other cases:

$$\varphi = y-1-2\,[(y-1)/2]$$

$$\xi = \min\{s-4; n-1\} - \max\{\,]s/3[; 2\} + 1$$

The explicit expression for k = 4 is not presented in the study because of its awkwardness but it is can be used.

For the purpose of memory and calculations volume evaluation needed for coding realization it is necessary to know with which k and s, r(n, k, s) coefficients have the largest values. Therefore, the following theorems are formulated and proved.

Theorem 7: For each k = n, r(n, k, s) takes the largest value if s =]k (n+1)/2[.

Theorem 8: For each n, r(n, k, s) takes the largest value if k =]n/2], s =]]n/2[(n+1)/2[. On the basis of these equations, the algorithms which realize TC are derived as

well as method for calculation of r(n, k, s) which provides the minimum labor intensity of TC realization with the set n value.

Expressions (Eq. 11 and 12) were used for calculating R_n (p) for the above mentioned Fig. 1 and 2. Equation 13 is used for numeration algorithm development.

Equations 14 and 15 set symmetry properties for coefficients. Symmetry evaluation helps to save memory needed for coefficients storage during coding realization by at least 4 times. Theorems 7 and 8 state the memory volume for TC realization.

Other equations are used to calculate coefficients values in different coding realization variants. In such a way, mathematical apparatus is formed needed for TC realization.

Previously, it was determined that w code word which corresponds to n-block is an ordered binary data third (k, s, b (n, k, s)). Therefore, during the coding it is necessary to add corresponding number b (n, k, s) (where in $0 = b (n, k, s) = r (n, k, s)-1$) to each element of $R_{k, s}$ set.

By using correspondence tables of source and code blocks,coding labor intensity increases like block length exponential function. Obviously, coding effective realization is quite hard with such labor intensity even taking into consideration the development of modern microelectronics and flash-memory.

The alternative variant is to use numeration algorithms when b (n, k, s) is determined like integral function of such block parameters like number of units in the block and its positions numbers.

Let us go to numeration algorithm development and take one n-block and fix the number of k-unit i_k. In this case, the other digits from 1 till (i_k-1) will be placed within the (i_k-1) word size and the number of such possible k combinations with s sum will be equal to $r (i_k-1, k, s)$.

Further, we shall fix number i_k^{-1}. The other numbers from 1 till i_{k-2} will be placed within the $(i_{k-1}-1)$ word size. In this case, the number of all possible combinations (k-1) numbers in $(i_{k-1}-1)$ word size with $s-i_k$ sum will be equal to $r (i_{k-1}-1, k-1, s-i_k)$. The same variant will be used for all other numbers tilli_1 including. As the result there is a value $r (i_j-1, j, s-i_k-i_{k-1}-...-i_{j+1})$ for each i_j. It is logical to suppose the following:

$$\begin{aligned} b (n, k, s) = {} & r(i_k-1, k, s) + r (i_{k-1}-1, k-1, s-i_k) + \\ & r(i_{k-2}-1, k-2, s-i_k-i_{k-1}) + ... + \\ & r(i_2-1, 2, s-i_k-i_{k-1}-...-i_3) + \\ & r(i_1-1, 1, s-i_k-i_{k-1}-...-i_2) \end{aligned}$$

Being aware that $s-i_k-i_{k-1}-...-i_2 = i_1$, therefore, $r (i_1-1, 1, s-i_k-i_{k-1}-...-i_2) \in 0$. Additionally, we shall take into consideration that $s-i_k-i_{k-1}-...-i_3 = i_2+i_1$, $s-i_k-i_{k-1}-...-i_4 = i_3+i_2+i_1$, etc. Let us write down our hypothesis the following way:

$$\begin{aligned} b (n, k, s) = {} & r (i_k-1, k, i_k + i_{k-1}+...+i_1) + \\ & r (i_{k-1}-1, k-1, i_{k-1}+i_{k-2}...+i_1) + \\ & r (i_2-1, 2, i_2+i_1) \\ = {} & \sum_{j=2}^{k} r (i_j-1, j, \sum_{m=1}^{j} i_m) \end{aligned} \quad (22)$$

During the work, two theorems were formulated and proved. These theorems prove the hypothesis validity (Eq. 22). In this study proves are missed to save the place.

Theorem 9: Every number b (n, k, s) calculated according to Eq. 22 satisfy the inequation: $0 \le b (n, k, s) \le r (n, k, s)-1$.

Theorem 10: Between n-blocks and ordered number thirds (k, s, b (n, k, s)) where b (n, k, s) is determined with the Eq. 22 there is a unique correspondence. As an example we will analyze the numeration when n = 11, k = 4, s = 25. In this case, Eq. 22 will be:

$$\begin{aligned} b (11, 4, 25) = {} & r (i_4-1, 4, 25) + r (i_3-1, 3, 25-i_4) + \\ & r (i_2-1, 2, 25-i_4-i_3) \end{aligned}$$

Let us write down all possible combinations for four unit position numbers which add up to 25 in 11 bit word size (Table 1). It can be determined that each combination has its own number distinguished from every other.

Based on the presented mathematical apparatus, coding and decoding algorithms were developed as well as their practical realization. These materials are presented by Alexandrovich *et al*. (2011) and Kozlov *et al*. (2011).

Apart of data compression, universal coding with binary Thirds Coding (TC) can be used for information security.

Coding algorithm was published and is well-known. However, to decode code words sequence in a right way, it is necessary to know the following (the complex of these ideas is a key):

n-source block length, i.e., the number of binary units in the block on which the source binary sequence is divided.

It is the main parameter without which it is impossible to make a decoding. Knowledge of n is necessary but insufficient condition for right decoding because it is necessary to know other parameters, stated below.

TC general or modified coding: In case of general coding, the number of code word bits for units' positions sum is permanent and calculated as the maximum sum for the

Table 1: Numeration sample

I_4	I_3	I_2	I_1	Formula summands (22)	Number b (n, k, s)
11	10	3	1	r (10, 4, 25)+r (9, 3, 14)+r (2, 2, 4) = 14+8+0	22
11	9	4	1	r (10, 4, 25)+r (8, 3, 14)+r (3, 2, 5) = 14+6+1	21
11	9	3	2	r (10, 4, 25)+r (8, 3, 14)+r (2, 2, 5) = 14+6+0	20
11	8	5	1	r (10, 4, 25)+r (7, 3, 14)+r (4, 2, 6) = 14+4+1	19
11	8	4	2	r (10, 4, 25)+r (7, 3, 14)+r (3, 2, 6) = 14+4+0	18
11	7	6	1	r (10, 4, 25)+r (6, 3, 14)+r (5, 2, 7) = 14+1+2	17
11	7	5	2	r (10, 4, 25)+r (6, 3, 14)+r (4, 2, 7) = 14+1+1	16
11	7	4	3	r (10, 4, 25)+r (6, 3, 14)+r (3, 2, 7) = 14+1+0	15
11	6	5	3	r (10, 4, 25)+r (5, 3, 14)+r (4, 2, 8) = 14+0+0	14
10	9	5	1	r (9, 4, 25)+r (8, 3, 15)+r (4, 2, 6) = 6+6+1	13
10	9	4	2	r (9, 4, 25)+r (8, 3, 15)+r (3, 2, 6) = 6+6+0	12
10	8	6	1	r (9, 4, 25)+r (7, 3, 15)+r (5, 2, 7) = 6+3+2	11
10	8	5	2	r (9, 4, 25)+r (7, 3, 15)+r (4, 2, 7) = 6+3+1	10
10	8	4	3	r (9, 4, 25)+r (7, 3, 15)+r (3, 2, 7) = 6+3+0	9
10	7	6	2	r (9, 4, 25)+r (6, 3, 15)+r (5, 2, 8) = 6+1+1	8
10	7	5	3	r (9, 4, 25)+r (6, 3, 15)+r (4, 2, 8) = 6+1+0	7
10	6	5	4	r (9, 4, 25)+r (5, 3, 15)+r (4, 2, 9) = 6+0+0	6
9	8	7	1	r (8, 4, 25)+r (7, 3, 16)+r (6, 2, 8) = 1+2+2	5
9	8	6	2	r (8, 4, 25)+r (7, 3, 16)+r (5, 2, 8) = 1+2+1	4
9	8	5	3	r (8, 4, 25)+r (7, 3, 16)+r (4, 2, 8) = 1+2+0	3
9	7	6	3	r (8, 4, 25)+r (6, 3, 16)+r (5, 2, 9) = 1+0+1	2
9	7	5	4	r (8, 4, 25)+r (6, 3, 16)+r (4, 2, 9) = 1+0+0	1
8	7	6	4	r (7, 4, 25)+r (6, 3, 17)+r (5, 2, 10) = 0+0+0	0

stated source block length, the second summand in the Eq. 4. In case of modified coding the number of code word bits for units positions sum is permanent and depends on the number of units in the block, the second summand in the Eq. 9.

It is obvious that from the point of data compression effectiveness, modified coding is more preferable but general coding can also be used for information security. Without knowing which coding is used, it is impossible to decode the information, even if the block source length is known.

If the adaptation procedure was used, what k_{max} is equivalent to?: In case adaptation was used, coding is applied to k values which are less or equal to k_{max} for other values block is transferred without coding with the corresponding tag in the form of code word. The following adaptation variant is possible: coding is used for k values which are great of equal to n-k_{max} taking into consideration coding symmetry, Eq. 14 and 15, i.e., compression for k is equal to that of n-k.

When the inversion is used: It was mentioned before that coding has symmetry property meaning for p = 0.1 it is equal to that for p = 0.9. Therefore, not only binary block can be coded but its inversion with the tag in the form of code word.

Tag location within the code word: To make decoding more complicated, it is possible to place tags, i.e., adaptation and inversion presence bits can be placed not at the beginning of code word format but for example adaptation presence in the 3rd bit and inversion presence in the 5th bit or vice versa. Of course, these positions should be within the minimum possible code word length. Moreover, it is obvious that there can be variants when adaptation and inversion are not used at all or only one of them is used.

It should be highlighted that the full set cannot be used in this case because of the great amount of different correlation variants, 5 types of key parameters andbecause of nothing to compare with.

Thus, it can be stated that TC will not only remove statistical superabundance but also will protect information.

DISCUSSION

In the following study, the attempt to unite two issues was made:

- Compression without the loss of binary data when source statistics is unknown
- Information security, encryption will help to exclude unauthorized access to the source binary sequence

On one side, these tasks are opposite because data compression is an exclusion of superabundance from the source binary sequence and leads to noise immunity decrease. As it is known from the information theory after excluding statistical superabundance, the number of zeros and ones in the code sequence and possibility of their appearance will be almost equal, pseudo-random. Moreover, it is almost impossible to track conformity.

Thus, it is almost impossible to make the right decoding with the known coding algorithm and unknown key which consists of 5 random independent parameters and if there is no “human factor” leak.

Hereafter, researchers will pay their main attention to "breaking" issues. In total, the proposed method deemed to have potential.

CONCLUSION

Researchers of the study propose their own solution for data compression issue and information security increase.

REFERENCES

Alexandrovich, A.Y., V.A. Shurigin and I.M. Yadikin, 2011. Universal coding method for binary data. J. Radio Electron., 2: 94-115.

Haleem, M.A., C.N. Mathur and K.P. Subbalakshmi, 2006. Joint distributed compression and encryption of correlated data in sensor networks. Proccedings of the IEEE Military Communications Conference, October 23-25, 2006, Washington, DC., pp: 1-7.

Kozlov, N.A., D.M. Mikhaylov and V.A. Shurigin, 2011. Method of image compression for wireless capsule endoscopy. J. Specialized Mach. Commun., 6: 34-37.

Lv, C.F. and Q.F. Zhao, 2007. Integration of data compression and cryptography: Another way to increase the information security. Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, Volume 2, May 21-23, 2007, Niagara Falls, Ont., pp: 543-547.

Mohamed, N.N., H. Hashim, Y.M. Yussoff, M.A.M. Isa and S.F.S. Adnan, 2014. Compression and encryption technique on securing TFTP packet. Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics, April 7-8, 2014, Penang, pp: 198-202.

Vilenkin, N.Y., 1969. Combinatorics. Nauka, Moscow, Pages: 328, (In Russian).