Protection of Numerical Information Based on Permutations

Oleksiy A. Borysenko [1], Oleksii Y. Horiachev [2], Viktor V. Serdyuk [3], Andriy O. Horyshnyak [4], Oleksandr M. Kobyakov [5] and Olga V. Berezhna [6]

[1-6] Sumy State University, st. Rimsky-Korsakov, 2, Sumy, Ukraine

Abstract

The article solves the problem of protecting decimal numbers used in systems of information transmission, processing and storage from unauthorized access with simultaneous correction of single errors in them and detection of error bursts. To protect the decimal number, each of its digits is first converted to a binary-decimal digit, and then, using a special table, into a binary-coded permutation. After that, the digits of the decimal number themselves are mixed. The paper gives estimates of the level of secrecy of decimal numbers encoded in this way. Since each digit of a decimal number can contain one of 10 digits, 10 permutations are required to encode them. To obtain them, at least 4 elements 0, 1, 2, 3 are required. They form 24 permutations, of which 14 are redundant. Specially selected 10 binary-coded permutations out of 24 form a binary-coded permutation code with a minimum code distance equal to 4. This allows correction of any single error and detection of double errors on the set of permutations.

Keywords

Information protection, numerical codes, secrecy, permutations, errors, noise immunity

1. Introduction

In practice, binary-decimal codes have become widespread, with the help of which information from various sensors is extracted and transmitted, for example, information about the amount of consumed thermal and electrical energy, water and other similar indications. Usually, each binary-decimal digit taken from the sensor is transmitted over a communication channel, essentially a telecommunication system, which includes a buffer memory with an encoder, a communication line, an information display device, and a receiver with a decoder [1]. The communication line can be both wired and mobile, using radio communication. In the latter case, information can be transmitted directly to moving objects, such as cars.

However, the transmitted information in some cases must be protected from unauthorized access. To do this, the binary-decimal digits of each decimal number are uniformly mixed using the appropriate tables. At the receiving end, these tables allow to restore the original information. They are, in essence, cipher keys. Moreover, the secrecy of the mixed each binary-decimal place can be significantly increased by additional mixing of the bits of binary-decimal numbers.

However, in addition to protecting against unauthorized access, it is often required to further increase the noise immunity of the transmitted binary-decimal numbers.

Binary-decimal coding protects to a certain extent the transmitted or stored decimal digits from interference due to the redundancy of a binary-decimal code containing sixteen four-bit

III International Scientific And Practical Conference "Information Security And Information Technologies", September 13–19, 2021, Odesa, Ukraine

EMAIL: 5352008@ukr.net (A. 1); alevgor@gmail.com (A. 2); viktman2012@gmail.com (A. 3); a.horishnyak@ias.sumdu.edu.ua (A. 4); o.kobyakov@ekt.sumdu.edu.ua (A. 5); o.berezhna@ekt.sumdu.edu.ua (A. 6).
ORCID: 0000-0001-7466-9135 (A. 1); 0000-0002-4251-0813 (A. 2); 0000-0002-1694-8651 (A. 3); 0000-0002-9799-0180 (A. 4); 0000-0003-4591-2102 (A. 5); 0000-0001-7105-1276 (A. 6)

binary-decimal code words. However, the level of protection against interference is still low, although for a number of practical cases it may be acceptable. Therefore, it became necessary to increase it.

It was proposed to solve this problem in [1-4] using binary-decimal error-correcting codes, which are essentially decimal digits encoded with error-resistant combinations. For this purpose in [1] the coding of binary-decimal digits by equilibrium code combinations was introduced, which significantly increased the ability of the telecommunications system to detect errors [1-4].

To assess the noise immunity of such codes, it was proposed to use formulas for the probabilities of transition of code combinations into classes of correct combinations, allowed erroneous combinations that are not detected and forbidden combinations that can be detected [5]. According to the results of the analysis, it was concluded that the use of equilibrium codes provides the requirements of the reliability class I1 of the international standard IEC 870-5-1-95 in the whole range of failure levels of one bit of information [1]. At the same time, the secrecy of information was increased, since there was no reliable test for unravelling their values, because statistics for decimal digits presented in the form of equilibrium code combinations does not help well, unlike text information, for the decoding of which the statistical probabilities of letters play an essential role.

However, errors in the transmission of decimal digits by equilibrium code combinations are difficult to eliminate, and the implementation of ARQ in mobile communications is sometimes difficult. Therefore, the task arose of developing a telecommunication system that would not only detect errors, but also correct them, using inseparable codes, in order to hide the true value of decimal digits during transmission.

2. Problem statement

The task of this work is to increase the noise immunity of transmitted binary-decimal digits, accompanied by error correction, with sufficient protection against unauthorized access.

For this, it is proposed to enhance the noise immunity of binary-decimal information by using inseparable codes on permutations, since, on the one hand, they allow error detection and correction, and on the other hand, they can hide the true information deeper.

Permutations are widespread in mathematics. Permutations are used in abstract algebra, and they are also used to solve combinatorial optimization problems, for example, the travelling salesman problem [6-8].

In addition to solving mathematical problems, permutations are used in practical problems of protecting information from unauthorized access [9-16]. The area of their possible application is constantly expanding. Along with this, permutations successfully solve the problem of anti-jamming coding, since by their nature they contain redundant information, which makes it relatively easy to find and, which is especially important for small mobile devices, to eliminate errors in messages transmitted with their help [17,18]. In addition, the permutations make it possible to combine solutions to the problems of anti-jamming coding with effective protection of information from unauthorized access.

3. Coding with permutations

Any finite sequence of distinct elements of length n is a permutation. While any symbols can be elements of permutations, most often numbers are used as them. For example, a sequence of four different digits 0123 would be a permutation of length $n = 4$. At the same time, a sequence of 1011 of length $n = 4$ would not be a permutation, since it only consists of two different repeating elements 0 and 1.

The set of n! permutations of length n forms a permutation code. The difference $n \cdot \log_2 n - \log_2 n!$ forms redundant information of this code, which with increasing of n can reach a significant value, determining the high noise immunity of codes on permutations. In addition, permutations do not have repeating elements and, therefore, obtaining their statistics is difficult. It can be obtained, with high effort, only on a large number of permutations, which greatly complicates the deciphering of information hidden in the permutations.

In the tasks of anti-jamming coding and information protection the elements of permutations are represented in binary form. Such their representation will be called binary-coded. The number of binary bits in binary-coded permutations is defined as the whole logarithm of the permutation elements number n:

$$m = \lceil \log_2 n \rceil \qquad (1)$$

10 different binary-coded permutations are required to encode binary-decimal information.

Therefore, the minimum value of n that can provide the required number of permutations will be 4, since 4 × 3 × 2 = 24 > 10. Of these 24 permutations, 10 permutations are used to encode 10 binary-decimal digits. Each of them encodes one of the digits, for example, permutation 0123 is used to encode 0. The remaining 14 possible permutations are redundant. One of the possible variants of representation of binary-decimal digits by permutations is shown in Table 1. Together, binary-decimal digits in Table 1 form a binary-decimal code (2-10 code).

Table 1
Coding with permutations

№	2-10 code	Permutations
0	0000	0123
1	0001	0132
2	0010	0213
3	0011	0231
4	0100	0312
5	0101	0321
6	0110	1023
7	0111	1032
8	1000	1203
9	1001	1230

3.1. Information secrecy

The number of encoding variants of binary-decimal digits by permutations will be equal to the number of combinations 10 out of 24, each of which can be specified by the corresponding table, like Table 1. Each of these variants, in turn, can be represented by one of 10! permutations encoding 10 digits, each of which can also be represented in the form of a table. Each of these tables can act as a cipher key, consisting of $10!\cdot C^{10}_{24}$ permutations for one decimal place.

In addition, the decimal digits, the number of which is equal to k, can also be shuffled in various ways during their transmission. Accordingly, the total number of permutation variants that can be used to encrypt the decimal permutation code will be equal to $M = k!\cdot 10!\cdot \ \cdot C^{10}_{24}$. If k, for example, equals 10, then the number of variants of the cipher $M = 10!\cdot 10\cdot \ \cdot C^{10}_{24} = 2.58\cdot 10^{19}$. This is a fairly large number of brute force options required to break the cipher. It should be borne in mind that the statistics of the numbers in the permutation cipher is poorly expressed, which greatly complicates its disclosure. The dependence of the M value, which characterizes the complexity of the proposed cipher disclosure, from the parameter k is shown in Table 2 and in the graph Figure 1.

Table 2
Number of permutations M

k	M	k	M
1	$7.11\cdot 10^{12}$	5	$8.54\cdot 10^{14}$
2	$1.42\cdot 10^{13}$	6	$5.12\cdot 10^{15}$
3	$4.27\cdot 10^{13}$	7	$3.58\cdot 10^{16}$
4	$1.70\cdot 10^{14}$	8	$2.86\cdot 10^{17}$
k	M	k	M
9	$2.58\cdot 10^{18}$	15	$9.30\cdot 10^{24}$
10	$2.58\cdot 10^{19}$	16	$1.48\cdot 10^{26}$
11	$2.84\cdot 10^{20}$	17	$2.53\cdot 10^{27}$
12	$3.40\cdot 10^{21}$	18	$4.55\cdot 10^{28}$
13	$4.43\cdot 10^{22}$	19	$8.65\cdot 10^{29}$
14	$6.20\cdot 10^{23}$	20	$1.73\cdot 10^{31}$

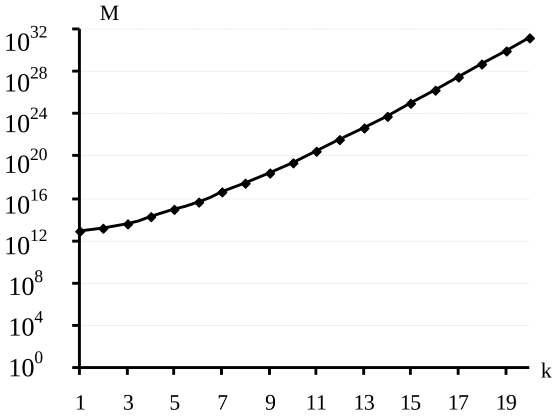

Figure 1: Graph of *M* versus *k*

3.2. Evaluation of the noise immunity of the code on permutations

In addition to secrecy, permutations can significantly increase the noise immunity of the binary-decimal code. This is due to the fact that the binary-coded representation of such permutations according to formula (1) will contain four digits of length m = 2. Permutations P of length n = 4 and their binary-coded representation BCP are presented in Table 3.

Table 3
Binary-coded permutations

P	BCP	P	BCP

0123	00 01 10 11	2013	10 00 01 11
0132	00 01 11 10	2031	10 00 11 01
0213	00 10 01 11	2103	10 01 00 11
0231	00 10 11 01	2130	10 01 11 00
0312	00 11 01 10	2301	10 11 00 01
0321	00 11 10 01	2310	10 11 01 00
1023	01 00 10 11	3012	11 00 01 10
1032	01 00 11 10	3021	11 00 10 01
1203	01 10 00 11	3102	11 01 00 10
1230	01 10 11 00	3120	11 01 10 00
1302	01 11 00 10	3201	11 10 00 01
1320	01 11 10 00	3210	11 10 01 00

Each permutation differs from others by at least two elements, and therefore, the minimum code distance in a binary code on permutations is 2. Such a code distance allows detecting in binary-coded permutations all single errors, as well as all errors of odd multiplicity 1, 3, 5, ...

Increasing the code distance will improve the noise immunity of binary-coded permutations. To achieve this, out of all 24 permutations of length $n = 4$, 10 allowed permutations should be selected, as shown in Table 4, which differ from each other by three elements, and thereby ensure the minimum code distance between their binary representations equal to 4. This allows not only detecting double errors in binary-coded permutations, but also correcting any single error in them.

Table 4
Permutations with minimum code distance 4

P	BCP	P	BCP
0123	00 01 10 11	2013	10 00 01 11
0231	00 10 11 01	2130	10 01 11 00
0312	00 11 01 10	2301	10 11 00 01
1203	01 10 00 11	3021	11 00 10 01
1320	01 11 10 00	3102	11 01 00 10

3.2.1. The fraction of detected errors

The noise immunity of a code on binary-coded permutations can be estimated using a characteristic called the fraction of detected errors D [5, 18]. It shows the probability with which any error translates the permutation into a forbidden combination that can be detected. The D value is defined as the ratio of the number of forbidden combinations Z_f to the total number of combinations $D = Z_f / n^n = 246 / 256 = 0.96$.

4. Error detection

A transmission error can translate a binary-coded permutation into either a forbidden combination that is not a permutation, or into one of the permutations. In the case where an error converts a permutation to a non-permutation combination, it can be easily detected as follows.

First, since all permutations contain the same elements, arranged in a different order, the sum of the binary numbers encoding these elements must remain constant. It forms a checksum, the same for all permutations, equal to

$$S = n\cdot(n - 1) / 2\ . \qquad (2)$$

It can be used to detect erroneous combinations, the checksum of which does not coincide with the value determined by the formula (2) [17]. For the considered code on permutations, such a checksum is equal to $S = 4\cdot(4 - 1) / 2 = 6$.

Example 1. On the receiving side, during permutation transmitting, a sequence of elements 1231 was received, which is not a permutation. Counting the sum of these elements gives the result 1 + 2 + 3 + 1 = 7. This number does not coincide with the checksum value obtained above for the code on permutations $S = 6$. This means that the resulting sequence is not a permutation and contains an error.

Second, the appearance of two or more identical elements in a permutation, during its transmission or storage obviously transforms it into a combination that is not a permutation. Then, by comparing the elements of the transmitted combinations on the receiving side, it is possible to establish whether they are permutations or not.

Example 2. On the receiving side, a sequence of elements 1231 was obtained. As a result of comparing the first element of this sequence with all other elements, it is found that it coincides with the fourth element: $\underline{1}$ 23 $\underline{1}$. Therefore, the resulting sequence is not a permutation and contains an error.

4.1. Double error detection

In the case when a double error occurs during the transmission of a binary-coded permutation, it can translate into one of the 14 forbidden permutations. The fact that the allowed permutation can translate solely into the forbidden permutation is explained by using for the encoding of numerical information only permutations with the minimum code distance 4. Such an error can be detected on the receiving side by comparing the received permutation with all allowed permutations given in Table. 4. If a match of the received permutation with one of the 10 allowed permutations is found, then the decision is made that it is correct; otherwise it is forbidden and contains a double error.

Example 3. Permutation 0123 (00 01 10 11) after the interference translated into permutation 1023 (01 00 10 11). Comparing this permutation with all allowed permutations presented in Table 4, shows no coincidence with any of them and, accordingly, indicates that it is forbidden. Therefore, it contains a double error. Indeed, in the permutation 0123 0 transformed into to 1, and 1 into 0.

4.2. Error correction

Comparing a binary-coded permutation containing an error in any element with all 10 allowed permutations allows a single error to be corrected. All permutations except one will differ from the erroneous sequence by more than one element. Any permitted permutation that differs from a permutation with an error in one element will be considered its corrected value.

Example 4. On the receiving side, a sequence of elements 1231 was received. By calculating the checksum and comparing the elements with each other, it is found that this sequence is not a permutation, which means that it contains an error. Since the minimum coding distance for permutations of Table 4 is 4, it is possible to correct a single error. To correct it, the erroneous sequence 1231 is compared with all allowed permutations in Table 4. As a result of this comparison, it is found that among the allowed permutations only one permutation 0231 differs from the obtained sequence by one element. This permutation is recorded as the correct value of the received sequence: 1231 → 0231.

However, the use of specially selected permutations for detecting double errors and correcting single errors reduces the level of secrecy of information, since the opponent can start breaking the cipher just from the analysis of these permutations. Therefore, it is necessary to weigh what is more important for the transmission of information, its noise immunity or secrecy, and accordingly choose the method of protecting decimal digits from interference.

4.3. Algorithm for detecting and correcting errors

The error detection and correction algorithm contains the following steps.

Step 1. In the received binary combination of 8 bits, the sum of its permutation elements, each of which consists of 2 binary digits, is calculated. If the calculated value equals 6, then it is considered as one of 24 binary-coded permutations, which may be correct or incorrect.

Step 2. The received permutation is compared with 10 allowed binary-coded permutations representing decimal digits. In the case when there is allowed permutation that coincides with the received permutation, then it is written as correct. If it differs from all the allowed permutations by the value of two or more elements, then it is erroneous and can be corrected by ARQ.

Step 3. If the calculated value doesn’t equal 6, then the received binary combination is an erroneous sequence that is not a permutation. In this case, some of its elements have the same value. If the received sequence differs from one of the 10 allowed permutations in only one element, then this one permutation will be the corrected permutation. In other case the error can only be corrected by ARQ.

5. Conclusions

The inseparable code on permutations proposed in the work for encoding digits allows solving the problem of digital information transmission secrecy, and at the same time ensures its noise immunity. Wherein, the secrecy of information can reach acceptable values for many applications due to the special properties of the permutations, which make it possible to hide the statistics of the transmitted decimal digits.

Along with the secrecy the permutations can effectively solve the problem of increasing the noise immunity of the transmitted digits. They allow detection of errors bursts and fix single errors. It is also important that the considered methods of detecting and correcting errors in permutations, used to encode decimal digits, are quite simple to implement.

6. References

[1] O. Borysenko, O. Berezhna, A. Novhorodtsev, V. Serdyuk, M. Yakovlev,

"Information transmission and display system with numerical data protection", Information processing systems. Vol. 2 (157), 2019, pp. 103-108. (in Ukrainian)

[2] O. Borysenko, V. Kalashnikov, Chapter 7: "Description and applications of binomial numeral systems complex" in Security and noise immunity of telecommunication systems: new solutions to the codes and signals design problem: Collective monograph. ASC Academic Publishing, Minden, Nevada, 2017, pp. 147-159.

[3] A. Kuznetsov, R. Serhiienko, D. Prokopovych-Tkachenko, B. Akhmetov, "Chapter 3: Representation of cascade codes in the frequency domain" in Security and noise immunity of telecommunication systems: new solutions to the codes and signals design problem: Collective monograph. ASC Academic Publishing, Minden, Nevada, 2017, pp. 71-101.

[4] A. Kuznetsov, S. Ksvun, Y. Gorbenko, "Chapter 4: The methodology of evaluating the energy gains from coding in channels with grouping errors" in Security and noise immunity of telecommunication systems: new solutions to the codes and signals design problem: Collective monograph. ASC Academic Publishing, Minden, Nevada, USA, 2017, pp. 102-119

[5] N.T. Berezyuk, Information coding (binary codes). Directory. Edited by N.T. Berezyuk, Vishcha shkola, Kharkiv, 1978. (in Russian)

[6] E. Reinhold, J. Nivergelt, N. Deo, Combinatorial algorithms: theory and practice, Mir, Moscow, 1980. (in Russian)

[7] D.Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd ed., Addison-Wesley Professional, 1997.

[8] D.Knuth, The Art of Computer Programming, Vol. 4A: Combinatorial Algorithms, Part 1, 1st ed., Addison-Wesley Professional, 2011.

[9] D. Smith, R. Montemanni, "A new table of permutation code", Designs, Codes and Cryptography, Vol. 63, pp. 241–253, 2012.

[10] W. Stallings, Cryptography and Network Security Principles and Practices, fourth ed., Prentice Hall, 2005.

[11] R. Girija, H. Singh, "A new substitution-permutation network cipher using Walsh Hadamard Transform" in Proceedings of International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 2017, pp. 168 - 172. DOI: 10.1109/IC3TSN.2017.8284470

[12] A. Aryal, S. Imaizumi, T. Horiuchi, H.i Kiya, "Integrated algorithm for block-permutation-based encryption with reversible data hiding" in Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 203 - 208. DOI: 10.1109/APSIPA.2017.8282028

[13] I. Janiszczak, R. Staszewski, "An improved bound for permutation arrays of length 10", [On-line]. Available: http://www.iem.uni-ue.de/preprints/IJRS.pdf [October 16, 2014].

[14] R. Montemanni, J. Barta, D.H. Smith, "Permutation codes: a branch and bound approach" in Proceedings of the International Conference on Pure Mathematics, Applied Mathematics, Computational Methods (PMAMCM), 2014, pp. 86-90.

[15] J. Barta, R. Montemanni, "Hamming Graphs and Permutation Codes" in Proceedings of Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), 2017, pp. 154 -158. DOI: 10.1109/MCSI.2017.35

[16] J. Barta, R. Montemanni, D.H. Smith, "A branch and bound approach to permutation codes" in Proceedings of the IEEE Second International Conference of Information and Communication Technology (ICOICT), 2014, pp. 187–192. DOI: 10.1109/ICoICT.2014.6914063

[17] O. Borysenko, O. Horiachev, "Interference-free transmission of economic information on the basis of permutations", Actual problems of economics. Kyiv, vol. 3 (141), 2013, pp. 156 - 163. (in Russian)

[18] O. Borysenko, O. Horiachev, S.Matsenko, O.Kobiakov, "Noise-immune codes based on permutations" in Proceedings of 9th International IEEE Conference «Dependable Systems, Services and Technologies DESSERT'2018», 2018, pp. 645 - 648. DOI: 10.1109/DESSERT.2018.8409204