Теоретические основы информатики

Theoretical foundations of computer science

УДК 004.032.26:004.056.55

ИСПОЛЬЗОВАНИЕ ГИПЕРКОМПЛЕКСНЫХ ЧИСЕЛ В ПРОТОКОЛЕ СОГЛАСОВАНИЯ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

П. П. УРБАНОВИЧ[1), 2)]***, Н. П. ШУТЬКО***[1)]

[1)]*Белорусский государственный технологический университет, ул. Свердлова, 13а, 220006, г. Минск, Беларусь*
[2)]*Люблинский католический университет им. Иоанна Павла II, ал. Рацлавицке, 14, 20-950, г. Люблин, Польша*

Аннотация. Проанализированы особенности структурной и функциональной организации двух взаимодействующих нейронных сетей на основе известной архитектуры в виде древовидной машины четности (ДМЧ) при использовании алгебр действительных и гиперкомплексных чисел. Такие машины применяются как альтернатива алгоритму Диффи – Хеллмана для генерации двумя абонентами общего тайного криптографического ключа.

Образец цитирования:
Урбанович ПП, Шутько НП. Использование гиперкомплексных чисел в протоколе согласования криптографических ключей на основе нейронных сетей. *Журнал Белорусского государственного университета. Математика. Информатика.* 2024;2:81–92 (на англ.).
EDN: MIPTIK

For citation:
Urbanovich PP, Shutko NP. Usage of hypercomplex numbers in a cryptographic key agreement protocol based on neural networks. *Journal of the Belarusian State University. Mathematics and Informatics.* 2024;2:81–92.
EDN: MIPTIK

Авторы:
Павел Павлович Урбанович – доктор технических наук, профессор; профессор кафедры информационных систем и технологий факультета информационных технологий[1)], приглашенный профессор[2)].
Надежда Павловна Шутько – кандидат технических наук, доцент; доцент кафедры информатики и веб-дизайна факультета информационных технологий.

Authors:
Pavel P. Urbanovich, doctor of science (engineering), full professor; professor at the department of information systems and technologies, faculty of information technologies[a], and visiting professor[b].
p.urbanovich@belstu.by
https://orcid.org/0000-0003-2825-4777
Nadzeya P. Shutko, PhD (engineering), docent; associate professor at the department of informatics and web-design, faculty of information technologies.
shutko_bstu@mail.ru
https://orcid.org/0009-0003-6409-6099

Рассмотрены основные элементы математических моделей ДМЧ, функционирующих на базе перечисленных алгебр. Описаны особенности программной реализации симулятора системы на основе взаимодействующих ДМЧ, а также представлены результаты использования разработанного инструментального средства для анализа динамики процессов в рассматриваемой системе. Взаимное обучение и обмен данными выполнены для двух ДМЧ на основе протокола управления передачей данных и межсетевого протокола (TCP/IP). Наступление состояния синхронизации сетей определяется равенством хешей, которые каждая из сторон вычисляет на базе алгоритма безопасного хеширования. Хеши размером 512 бит генерируются преобразованием строкового представления текущего входного вектора весов нейронов. Приведена оценка устойчивости процесса синхронизации ДМЧ к геометрическим атакам третьей стороны.

Ключевые слова: нейрокриптография; древовидные машины четности; гиперкомплексные числа; взаимное обучение сетей.

Благодарность. Авторы выражают благодарность директору Института математики, информатики и ландшафтного дизайна Люблинского католического университета имени Иоанна Павла II доктору наук М. Плонковскому за предоставленную им информацию, которая позволила улучшить содержание статьи.

USAGE OF HYPERCOMPLEX NUMBERS IN A CRYPTOGRAPHIC KEY AGREEMENT PROTOCOL BASED ON NEURAL NETWORKS

P. P. URBANOVICH[a, b]***, N. P. SHUTKO***[a]

[a]*Belarusian State Technological University, 13a Svyardlova Street, Minsk 220006, Belarus*
[b]*The John Paul II Catholic University of Lublin, 14 Racławickie Alley, Lublin 20-950, Poland*
Corresponding author: P. P. Urbanovich (p.urbanovich@belstu.by)

Abstract. We analyse the features of the structural and functional organisation of two interacting neural networks based on the known architecture in the form of a tree parity machine (TPM) using algebras of real and hypercomplex numbers. Such machines are used as an alternative to the Diffie – Hellman algorithm to generate a shared secret cryptographic key between two parties. The main elements of mathematical models of TPMs, operating on the basis of the listed algebras, are considered. The features of the software implementation of a system simulator based on interacting TPMs are described, and the results of using the developed tool for analysing the dynamics of processes in the system under consideration are presented. Mutual learning and data exchange of two TPMs are realised based on the transmission control and Internet protocols (TCP/IP). The synchronisation state of the networks is determined by the equality of the hashes that each party calculates based on the secure hash algorithm. A hash size of 512 bits are generated by transforming the string representation of the current input vector of neuron weights. The effectiveness of possible attempts by a third party to synchronise with two legitimate TPMs operating on the basis of algebras of hypercomplex numbers is assessed.

Keywords: neural cryptography; tree parity machines; hypercomplex numbers; networks mutual learning.

Acknowledgements. The authors express their gratitude to the director of the Institute of Mathematics, Computer Science and Landscape Design of the John Paul II Catholic University of Lublin doctor of science M. Plonkowski for the help and information provided, which contributed to the improvement of the article content.

Introduction

Two major problems associated with symmetric cryptography (storage and transport or agreement of keys secret to a third party) stimulated the search for a new solution. A solution was found in the form of the famous Diffie – Hellman (D-H) algorithm, which laid the foundation for public key cryptosystems [1]. But the mentioned protocol, when used to agree on a shared secret key between two users, **A** and **B**, also has weaknesses associated with certain types of attacks. As an alternative to the D-H-protocol, I. Kanter and W. Kinzel in 2002 proposed solving the problem of generating and transmitting a shared key using two interacting neural networks (NNs), called a tree parity machine (TPM) [2]. Mutual learning of networks means that the synaptic weights of two TPMs (**A** and **B**: $\boldsymbol{W}^{\mathbf{A}}$ and $\boldsymbol{W}^{\mathbf{B}}$) adapt to input (output) pairs according to certain rules. After the mutual learning procedure, the networks form identical sets of weight parameters, which are accepted as a joint secret key.

However, as it turned out, when implementing the networks based on two TPMs, there is a multiplicity of approaches, solutions and problems. All this is considered and analysed in an increasing stream of publications, the subject area of which combines cryptography and NNs – neural cryptography [3–9] – a relatively new scientific direction, the name of which probably first appeared in paper [10], dedicated to the cryptanalysis of data encryption standard (DES).

TPM is a one-way three-layer network, the architecture of which is described by two parameters (fig. 1): the number of neurons in the hidden layer (at level 2) K ($K \in \mathbb{N}$), and the number of input signals for each neuron N ($N \in \mathbb{N}$) [2–9; 11–13]. Each hidden unit works as a perceptron with independent receptive fields, including N input neurons and one output neuron.

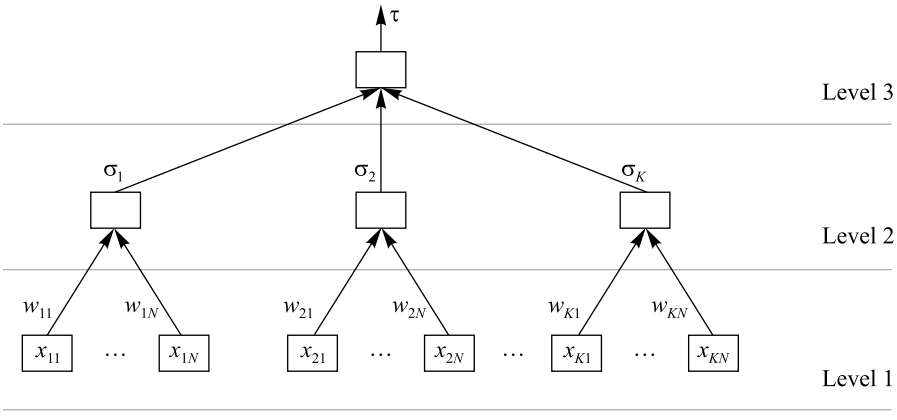

Fig. 1. TPM network architecture

At each step t of mutual learning, two TPMs, **A** and **B**, use a common input vector $\boldsymbol{X} = \{x_{uv}\}$ ($1 \le u \le K$ and $1 \le v \le N$; $\boldsymbol{X} \in \{-1, 1\}^{KN}$) and change (or not change) the eigenvector of weights $\boldsymbol{W} = \{w_{uv}\}$ ($w_{uv} \in \{-L, -L+1, \ldots, L\}$) in accordance with certain learning rules after exchanging the output signals (parameters) generated at this step: $\tau^{\mathbf{A}}$ and $\tau^{\mathbf{B}}$, or $\tau^{\mathbf{A/B}}$; $\tau^{\mathbf{A/B}} \in \{-1, 1\}$ and calculated in accordance with the formula

$$\tau = \prod_{u=1}^{K} \sigma_u, \tag{1}$$

where $\sigma_u \in \{-1, 1\}$ and

$$\sigma_u = \operatorname{sign}(\alpha_u) = \operatorname{sign}\left(\left(\frac{1}{\sqrt{N}}\right)\sum_{v=1}^{N} w_{uv} x_{uv}\right), \tag{2}$$

taking into account that when $\alpha_u = 0$ the function $\operatorname{sign}(\alpha_u)$ can take one of two values: –1 or +1 (in accordance with the accepted rule for training TPMs); in [5] a similar signum function is called «modified signum function».

The used rules and the duration of TPM synchronisation to obtain the same vector of weights ($\boldsymbol{W}^{\mathbf{A}} = \boldsymbol{W}^{\mathbf{B}}$) influence the security of this process, due to the possibility of implementing various types of attacks from the third network (**E**) with a similar architecture. The goal of such attacks is to synchronise the weights of the attacking network ($\boldsymbol{W}^{\mathbf{E}}$) with the weights of one of the legitimate parties of the system (for example, network **A**) [3; 5; 6; 11–17] to ultimately obtain the secret key generated by networks **A** and **B**.

The fundamental model of the TPM is based on the use of the algebra of real integers. At the same time, when solving a number of applied problems, various extensions of real numbers are used – hypercomplex numbers, which make it possible to describe the position of a point in a multidimensional space based on operations on the vectors [18]. As is known, the simplest examples of hypercomplex numbers are imaginary and double numbers, as well as quaternions. All arithmetic operations are performed on these numbers, on the basis of which the TPM model is created. Taking this circumstance into account, in [5; 6; 12] the usage of complex and double complex numbers was justified and analysed to quantitatively determine the parameters of the network synchronisation process based on the TPM architecture. Such architectures are called a tree parity complex machine (TPCM) and TP split-complex machine (TPSCM). In [19] the usage of quaternions (TP quaternion machine (TPQM)) is analysed. The TPM architecture based on complex numbers in [20] is called complex-valued TPM (CVTPM), and the architecture based on quaternions in [21] is called quaternion-valued TPM (QVTPM).

The definitions of the signum function for TPCM and TPQM are justified and described in [5; 19–21].

There is a number of applied tasks that are solved based on NNs using octonion algebra [22–24]. Many properties of octonions are similar to the properties of quaternions and complex numbers. But there is one significant difference between these systems: while the multiplication of complex numbers and quaternions has an associative property, this law does not hold for the multiplication of octonions. If we apply a weakened version of the associativity of multiplication [18, p. 45], then octonions can also be used to TPM modelling.

Similarity of the formal representation of hypercomplex numbers is presented in the form of expression

$$a_0 + a_1 i_1 + a_2 i_2 + \ldots + a_b i_b, \tag{3}$$

where $a_0, a_1, \ldots, a_b$ are arbitrary real numbers; $i_1, i_2, \ldots, i_b$ are imaginary units and $b \in \mathbb{N}$, as well as the commonality of mathematical operations on complex numbers ($b = 1$), quaternions ($b = 3$) and octonions ($b = 7$), make it possible to supplement the known TPCM and TPQM models with the TP octonion machine (TPOM) model based on a common methodological position.

The presented paper describes the TPOM model, obtained on the basis of a generalisation of the known results of using hypercomplex numbers for modelling a system of two TPMs, and also presents new results regarding the simulation of the TPMs interaction based on hypercomplex numbers and the transmission control and Internet protocols (TCP/IP protocol).

Materials and research methods

Some features of the TPM models development based on hypercomplex numbers. The TPM architecture, operating on the basis of a hypercomplex system of dimension $(b+1)$ in accordance with (3), is similar to the architecture based on real numbers (see fig. 1). The changes are related to the methods for applying the learning rule for networks **A** and **B** and the signum function modifying. These changes are due to the fact that all parameters of the network model (input and output quantities on fig. 1) are complex numbers (two-dimensional system), quaternions (four-dimensional, R^4) and octonions (eight-dimensional, R^8). The indicated dimensions correspond to the number of parts (1 – real plus b – imaginary) in the input vector $\boldsymbol{X}$, the values of the weight vector $\boldsymbol{W}$, the hidden layer neuron output σ and the network output τ.

For example, when using the complex number C (we will represent the number C in the canonical form, slightly modifying (3): $C = a_0 + ia_1$) the perceptrons input consists of K N-element vectors $\boldsymbol{X}_C = \{x_{u1}, x_{u2}, \ldots, x_{uN}\}_C$:

$$(x_{uv})_C = \left[(a_0)_{x,uv} + i(a_1)_{x,uv} \right]_C,$$

often identified with a single KN-element vector $\boldsymbol{X}_C = \{x_1, x_2, \ldots, x_{KN}\}_C$ of tetravalent complex numbers chosen from the set $\{(1,1), (-1,1), (-1,-1), (1,-1)\}$. And elements of the levels 1 and 2 (see fig. 1) are perceptrons having N-element weights $\boldsymbol{W}_C = \{w_{u1}, w_{u2}, \ldots, w_{uN}\}_C$:

$$(w_{uv})_C = \left[(a_0)_{w,uv} + i(a_1)_{w,uv} \right]_C,$$

where $1 \le u \le K$; $(a_0)_{w,uv}, (a_1)_{w,uv} \in \{-L, L+1, \ldots, L-1, L\}$. These elements are limited by the range $[-L, L] \times [-L, L]$ or $[-L, L]^2$, which is a natural extension of the $[-L, L]$-range associated with classical TPM.

Following the above reasoning, we can write down the remaining mathematical expressions that fully describe the architecture of the TPCM, as well as the modification of weights $\left(w_{uv}^{\mathbf{A/B}}\right)_C$ in the process of the **A** and **B** networks mutual learning. Using this approach, we shall now proceed to consider the TROM model.

TPOM architecture model. By analogy with a complex number, we write the octonion O in the canonical form:

$$O = a_0 + a_1 i + a_2 j + a_3 k + a_4 l + a_5 m + a_6 n + a_7 p, \tag{4}$$

here $a_0 - a_7 \in \mathbb{R}$.

Note that an octonion of form (4) is formally the sum of a real number a_0 with a vector (imaginary part): $a_1 i + a_2 j + a_3 k + a_4 l + a_5 m + a_6 n + a_7 p$. From the algebraic point of view, the Cayley – Dickson numbers are a set of eight-dimensional linear space over the field of real numbers. It follows that they can be represented as eight-element vectors with real coefficients.

The elements of the input vector of the TPOM network, like the elements of the weight vector, are octonions $\boldsymbol{X}, \boldsymbol{W}^{\mathbf{A/B}} \in O$:

$$\begin{aligned}(x_{uv})_O = \Big[&(a_0)_{x,uv} + i(a_1)_{x,uv} + j(a_2)_{x,uv} + k(a_3)_{x,uv} + \\ &+ l(a_4)_{x,uv} + m(a_5)_{x,uv} + n(a_6)_{x,uv} + p(a_7)_{x,uv} \Big]_O,\end{aligned} \tag{5}$$

$$\begin{aligned}(w_{uv})_O = \Big[&(a_0)_{w,uv} + i(a_1)_{w,uv} + j(a_2)_{w,uv} + k(a_3)_{w,uv} + \\ &+ l(a_4)_{w,uv} + m(a_5)_{w,uv} + n(a_6)_{w,uv} + p(a_7)_{w,uv} \Big]_O.\end{aligned} \tag{6}$$

In (5) $(a_0)_{x,uv}, (a_1)_{x,uv}, (a_2)_{x,uv}, (a_3)_{x,uv}, (a_4)_{x,uv}, (a_5)_{x,uv}, (a_6)_{x,uv}, (a_7)_{x,uv} \in \{-1, 1\}$. Each of the *KN*-elements of the input vector $\boldsymbol{X}_O$ is an element of a set consisting of 2^8 different quantities:

$$\{(1, 1, 1, 1, 1, 1, 1, 1), (-1, 1, 1, 1, 1, 1, 1, 1), \ldots, (1, -1, -1, -1, -1, -1, -1, -1), (-1, -1, -1, -1, -1, -1, -1, -1)\}.$$

The constraint imposed on the values of the weight vector $\boldsymbol{W}_O$ will be expanded to a set from the space $[-L, L]^8$ and in (6): $(a_0)_{w,uv}, \ldots, (a_7)_{w,uv} \in [-L, L]$. This representation is similar to the previous constructions due to the fact that there is a one-to-one mapping between octonions and points of the space R^8. The signum function divides the plane into 16 disjoint subsets.

The normalised value $(\alpha_u)_O$, corresponding to the inputs of the neurons *u* of the hidden layer, is calculated by the formula (see formula (2) above, as well as [21, p. 2–3]):

$$(\alpha_u)_O = \left\{ \left(\frac{1}{\sqrt{N}} \right) \left(\sum_{v=1}^{N} \left((a_0)_{w,uv} \right) \left((a_0)_{x,uv} \right) + i \sum_{v=1}^{N} \left((a_1)_{w,uv} \right) \left((a_1)_{x,uv} \right) + \right. \right.$$

$$\left. \left. + j \sum_{v=1}^{N} \left((a_2)_{w,uv} \right) \left((a_2)_{x,uv} \right) + \ldots + p \sum_{v=1}^{N} \left((a_7)_{w,uv} \right) \left((a_7)_{x,uv} \right) \right) \right\}_O . \quad (7)$$

The input of the neuron *u* of the hidden layer $(\sigma_u)_O$ determined via $\{1, -1, i, -i, j, -j, k, -k, l, -l, m, -m, n, -n, p, -p\}$:

$$(\sigma_u)_O = \left[(a_0)_{\sigma_u} + (a_1)_{\sigma_u} i + (a_2)_{\sigma_u} j + (a_3)_{\sigma_u} k + (a_4)_{\sigma_u} l + (a_5)_{\sigma_u} m + (a_6)_{\sigma_u} n + (a_7)_{\sigma_u} p \right]_O . \quad (8)$$

In (8) $(a_0)_{\sigma_u} = \operatorname{sign}\left(\operatorname{Re}(\alpha_u)_O \right) = \operatorname{sign}\left(\boldsymbol{R}(\alpha_u)_O \right)$; $(a_1)_{\sigma_u} = \operatorname{sign}\left(\boldsymbol{I}(\alpha_u)_O \right), \ldots, (a_7)_{\sigma_u} = \operatorname{sign}\left(\boldsymbol{P}(\alpha_u)_O \right)$ are seven imaginary parts; all of the listed functions are defined in the same way as similar functions are defined for the TPQM (QVTPM) architecture [21].

Following the reasoning used in [5; 20] for the analysis of TPM based on complex numbers (TPCM) and on quaternions (TRQM), we conclude that the signum function for TPOM will divide the space R^8 of possible values into 16 ($2 \cdot 8$) disjoint subspaces. So, the function is defined in an eight-dimensional space, where only 16 ($8 \cdot 2$) finite quantities occur. These quantities play the role of attractors, attracting the closest points to them. And this function can be defined like this:

$$\sigma_u(O) = \begin{cases} (1, 0, 0, 0, 0, 0, 0, 0),\ a_0 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_0 \ge 0, \\ (-1, 0, 0, 0, 0, 0, 0, 0),\ a_0 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_0 < 0, \\ (0, 1, 0, 0, 0, 0, 0, 0),\ a_1 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_1 \ge 0, \\ (0, -1, 0, 0, 0, 0, 0, 0),\ a_1 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_1 < 0, \\ (0, 0, 1, 0, 0, 0, 0, 0),\ a_2 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_2 \ge 0, \\ (0, 0, -1, 0, 0, 0, 0, 0),\ a_2 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_2 < 0, \\ (0, 0, 0, 1, 0, 0, 0, 0),\ a_3 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_3 \ge 0, \\ (0, 0, 0, -1, 0, 0, 0, 0),\ a_3 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_3 < 0, \\ (0, 0, 0, 0, 1, 0, 0, 0),\ a_4 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_4 \ge 0, \\ (0, 0, 0, 0, -1, 0, 0, 0),\ a_4 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_4 < 0, \\ (0, 0, 0, 0, 0, 1, 0, 0),\ a_5 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_5 \ge 0, \\ (0, 0, 0, 0, 0, -1, 0, 0),\ a_5 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_5 < 0, \\ (0, 0, 0, 0, 0, 0, 1, 0),\ a_6 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_6 \ge 0, \\ (0, 0, 0, 0, 0, 0, -1, 0),\ a_6 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_6 < 0, \\ (0, 0, 0, 0, 0, 0, 0, 1),\ a_7 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_7 \ge 0, \\ (0, 0, 0, 0, 0, 0, 0, -1),\ a_7 = \max\left(\{a_i : 0 \le i \le 7\} \right) \wedge a_7 < 0. \end{cases}$$

Taking this into account, the output of each of the perceptrons (level 2 in fig. 1) can be one of 16 Cayley numbers:

$$\{(1, 0, 0, 0, 0, 0, 0, 0), (-1, 0, 0, 0, 0, 0, 0, 0), \ldots, (0, 0, 0, 0, 0, 0, 0, 1), (0, 0, 0, 0, 0, 0, 0, -1)\}.$$

The output $\left(\tau^{\mathbf{A/B}}\right)_O$ of the TPOM networks is the result of using a formula that is an extension of (1) and (2) and is based on the TPQM (QVTPM) model [21, p. 3]:

$$\left(\tau^{\mathbf{A/B}}\right)_O = \left(\prod_{u=1}^{K} \boldsymbol{R}\left(\sigma\left(\alpha_u^{\mathbf{A/B}}\right)\right) + i\prod_{u=1}^{K} \boldsymbol{I}\left(\sigma\left(\alpha_u^{\mathbf{A/B}}\right)\right) + j\prod_{u=1}^{K} \boldsymbol{J}\left(\sigma\left(\alpha_u^{\mathbf{A/B}}\right)\right) + \ldots + p\prod_{u=1}^{K} \boldsymbol{P}\left(\sigma\left(\alpha_u^{\mathbf{A/B}}\right)\right)\right)_O. \quad (9)$$

The limitation of the elements of the weights $\left(\boldsymbol{W}^{\mathbf{A/B}}\right)_O$ occurs separately for each part of the octonion. For example,

$$\boldsymbol{R}\left(w_{uv}^{\mathbf{A/B}}\right)_O = \begin{cases} \operatorname{sign}\left(\boldsymbol{R}\left(\left(w_{uv}^{\mathbf{A/B}}\right)_O\right)\right)L, \text{ if } \left|\boldsymbol{R}\left(\left(w_{uv}^{\mathbf{A/B}}\right)_O\right)\right| > L, \\ \boldsymbol{R}\left(w_{uv}^{\mathbf{A/B}}\right)_O, \text{ otherwise,} \end{cases} \quad (10)$$

$$\boldsymbol{I}\left(w_{uv}^{\mathbf{A/B}}\right)_O = \begin{cases} \operatorname{sign}\left(\boldsymbol{I}\left(\left(w_{uv}^{\mathbf{A/B}}\right)_O\right)\right)L, \text{ if } \left|\boldsymbol{I}\left(\left(w_{uv}^{\mathbf{A/B}}\right)_O\right)\right| > L, \\ \boldsymbol{I}\left(w_{uv}^{\mathbf{A/B}}\right)_O, \text{ otherwise.} \end{cases} \quad (11)$$

If, for example, the Hebbian rule is used for mutual learning of TPOM networks [4; 5; 11; 20], then the modification of the weights at the step $(t+1)$, taking into account (4)–(11), will occur according to the following expressions separately for the real and imaginary parts. In particular, to change the real part and the first imaginary part of the weight, one should seize the expressions

$$\boldsymbol{R}\left(w_{uv}^{(t+1)}\right)_O = g\left[\boldsymbol{R}\left(w_{uv}^{(t)}\right)_O + \boldsymbol{R}\left(x_{uv}\tau^{(t)}\right)_O \Theta\left(\boldsymbol{R}\left(\tau^{(t)}\right)_O \boldsymbol{R}\left(\sigma_u^{\mathbf{A}(t)/\mathbf{B}(t)}\right)_O\right)\Theta\left(\boldsymbol{R}\left(\tau^{\mathbf{A}(t)}\right)_O \boldsymbol{R}\left(\tau^{\mathbf{B}(t)}\right)_O\right)\right], \quad (12)$$

$$\boldsymbol{I}\left(w_{uv}^{(t+1)}\right)_O = g\left[\boldsymbol{I}\left(w_{uv}^{(t)}\right)_O + \boldsymbol{I}\left(x_{uv}\tau^{(t)}\right)_O \Theta\left(\boldsymbol{I}\left(\tau^{(t)}\right)_O \boldsymbol{I}\left(\sigma_u^{\mathbf{A}(t)/\mathbf{B}(t)}\right)_O\right)\Theta\left(\boldsymbol{I}\left(\tau^{\mathbf{A}(t)}\right)_O \boldsymbol{I}\left(\tau^{\mathbf{B}(t)}\right)_O\right)\right]. \quad (13)$$

In (12) and (13), $\Theta(x)$ is the Heaviside function, which takes the value 0 for $x < 0$ and the value 1 in other cases, $g(x)$ is defined in the standard way (see (10) and (11)):

$$g(x) = \begin{cases} -L, \text{ if } x < -L, \\ L, \text{ if } x > L, \\ x, \text{ if } |x| \le L. \end{cases}$$

It is clear that with $(a_4)_{w,uv} = (a_5)_{w,uv} = (a_6)_{w,uv} = (a_7)_{w,uv} = 0$ (in (6)) and $(a_4)_{x,uv} = (a_5)_{x,uv} = (a_6)_{x,uv} = (a_7)_{x,uv} = 0$ (in (5)) the model and architecture TPOM turns into the TPQM, if additionally $(a_2)_{w,uv} = (a_3)_{w,uv} = 0$ and $(a_2)_{x,uv} = (a_3)_{x,uv} = 0$, then we get the TPCM model.

Results and discussion

Simulation of the NNs process synchronisation. To implement the considered mathematical models of TPMs, operating on the basis of algebras of real and hypercomplex numbers, as well as to analyse and compare the dynamic characteristics of the weights synchronising process of the corresponding NNs, a software tool (*RCQOv2*) has been developed. The tool is based on the classes *Simple*, *Complex*, *Quaternion and Octonion*, each of which implements the template class *NumberSystemBase*<T> and allows to perform mathematical operations on the corresponding numbers. In turn, the basic structural component of each model is the *Perceptron*. It corresponds to the template class *PerceptronBase*<T>, which inherits the classes *SimpleArithmeticPerceptron*, *ComplexPerceptron*, *QuaternionPerceptron*, *OctonionPerceptron*, the distinctive feature of which is its own implementation of the weights normalisation method.

Unlike well-known TPM interaction simulators, the version we developed is focused on the exchange of output data $\left(\tau^{\mathbf{A}} - \tau^{\mathbf{B}} \text{ or } \tau^{\mathbf{B}} - \tau^{\mathbf{A}}\right)$, as well as initialisation of the input message $\boldsymbol{X}^t$ (at the synchronisation

step t) via the TCP/IP protocol. This makes it possible to bring the simulated process closer to reality: two subscribers agree on a common secret cryptographic key while being far away from each other. If the used NN training rule at a certain synchronisation step t allows us to conclude that the weights of both TPMs have reached a state of equality ($\boldsymbol{W}^{\mathbf{A}} = \boldsymbol{W}^{\mathbf{B}}$), then hash sums are formed from the string representation of $\boldsymbol{W}^{\mathbf{A}}$ and $\boldsymbol{W}^{\mathbf{B}}$ using the SHA-512 algorithm, the exchange and comparison of which on each side provides for decision making an unambiguous decision on the equality (inequality) of the generated weights.

Results of computer modelling and their analysis. The *RCQOv2* application along with performing single simulations of the TPM synchronisation procedure allows us to accumulate and process data obtained from a series of experiments. Figure 2 shows an example (application window) with the results of 1000 experiments – distribution of the number of successful synchronisations (represented in histograms by columns) according to numbers of the required steps t (vertical axis in the fig. 2), for the TPCM architecture with parameters: $K = 5$, $N = 5$, $L = \pm 5$ (left side of the window); for comparison the normal distribution for networks with the same architecture in relation to the same scale of the steps number until the moment comes when $\boldsymbol{W}^{\mathbf{A}} = \boldsymbol{W}^{\mathbf{B}}$, is presented (right side of the window).

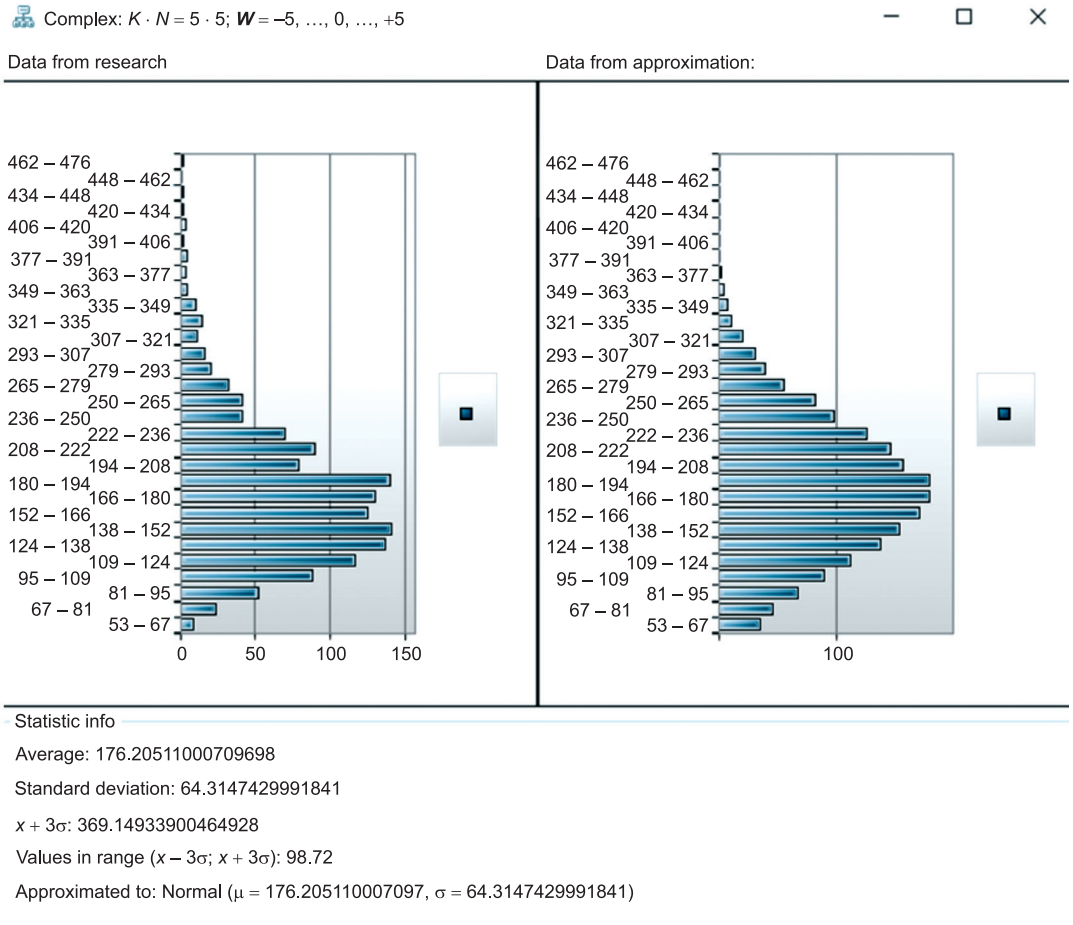

Fig. 2. Experimentally obtained (left) and theoretical (normal, right) distributions of the number of TPCMs to the number of steps until the moment comes when $\boldsymbol{W}^{\mathbf{A}} = \boldsymbol{W}^{\mathbf{B}}$ for 1000 simulations

In the lower part of the window at fig. 2 the calculated statistical parameters of the experimentally obtained distribution are given. These statistical data as well as a simple visual comparison of the distributions at fig. 2, indicate a fairly high degree of their similarity. It should be noted that 98.72 % of the observed data lies within three standard deviations from the mean (three-sigma rule). The resulting distribution was approximated using the method of moments by the Pearson criterion, provided that the experimental and theoretical values of the mean and variance (indicated at the bottom of the screen on fig. 2) are the same.

To further evaluate the degree of similarity of the resulting distribution to the normal one, an estimate in the form of distance or Bray – Curtis dissimilarity [25] was used. This characteristic is a statistic used to quantify the dissimilarity between two different samples, and ranges from 0 to 1 (0 means complete agreement (similarity), 1 means complete dissimilarity).

Tables 1 and 2 contain some statistical results obtained by simulating the synchronisation process TPCMs and TPQMs, respectively.

Table 1

Statistical results of simulations of the TPCMs synchronisation process

N	K	$\pm L$	Simulations number	Number of successful synchronisations	t (average)	Standard deviation	Bray – Curtis similarity index
5	5	5	1668	1409	176.2	64.3	0.204
5	6	5	1000	944	186.1	65.2	0.185
6	5	5	1000	925	149.2	55.5	0.200
5	5	6	1546	1285	230.4	80.7	0.182
6	6	6	1079	1061	212.1	73.4	0.185
6	7	6	1000	993	227.4	76.2	0.180
7	6	6	1000	929	211.1	68.6	0.175
6	6	7	1000	972	318.6	162.0	0.254
7	7	7	1011	970	280.8	91.6	0.174

Table 2

Statistical results of simulations of the TPQMs synchronisation process

N	K	$\pm L$	Simulations number	Number of successful synchronisations	t (average)	Standard deviation	Bray – Curtis similarity index
5	5	5	1357	1354	445.2	212.3	0.25
6	6	6	1070	1068	417.1	147.7	0.18
7	6	6	1000	1000	498.2	209.8	0.22
7	7	6	1000	1000	529.5	209.6	0.20
6	6	7	1000	999	656.5	272.1	0.21
6	7	7	1000	1000	708.9	273.1	0.19
7	6	7	1000	1000	858.5	385.8	0.23
7	7	7	1000	997	735.7	284.9	0.20
8	8	8	1000	999	780.7	273.6	0.19
9	9	9	1000	1000	1550.6	605.6	0.20

Similar results in the form of a histogram (fig. 3) and table form (table 3) are presented for TPM based on octonion algebra.

Simulation of the TPOMs ($N = 6$, $K = 6$, $L = \pm 6$) synchronisation procedure revealed the following: the minimum number of steps until the moment of equality of weights vectors comes was approximately less than 1000 (in several dozen experiments), and the maximum number of steps was more than 15 000 (also in several dozen experiments). For better visibility each scale number (t') on the horizontal axis in fig. 3 corresponds to a range of width 450. Moreover, the starting point for the number of steps is 1122 ($t = 1122$). Therefore, $t' = 1$ corresponds to the range $t = [1122, 1571]$, $t' = 2 - t = [1572, 2021]$, …, $t' = 30 - t = [14\,192, 14\,641]$.

Table 3

Statistical results of simulations of the TPOMs synchronisation process

N	K	$\pm L$	Simulations number	Number of successful synchronisations	t (average)	Standard deviation	Bray – Curtis similarity index
5	5	5	1233	1231	1258.2	607.7	0.25
5	6	5	1000	1000	1760.5	827.1	0.22

Ending of the table 3

N	K	$\pm L$	Simulations number	Number of successful synchronisations	t (average)	Standard deviation	Bray – Curtis similarity index
6	5	5	1000	1000	674.6	291.4	0.23
6	6	5	1000	1000	733.4	305.5	0.22
7	5	5	1000	1000	568.2	254.9	0.23
6	6	6	1168	1167	1604.6	711.7	0.23
5	5	7	1000	1000	12 698.3	6471.3	0.25
5	7	7	1000	997	12 923.6	5753.7	0.24
7	7	7	1000	996	5825.1	2779.1	0.23

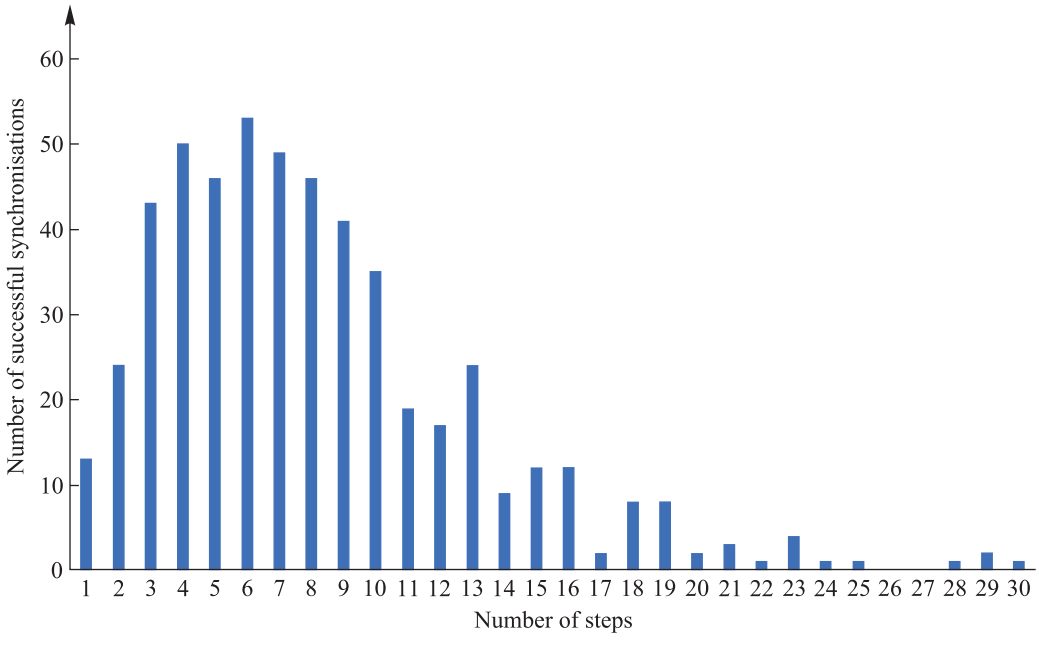

Fig. 3. Main part of the distribution of the number of TPOM ($N = 6$, $K = 6$, $L = \pm 6$) in accordance with the number of steps (t') until the moment comes when $\boldsymbol{W}^{\mathbf{A}} = \boldsymbol{W}^{\mathbf{B}}$ for 1168 simulations

We have received practical confirmation of an intuitive and expected pattern: with approximately comparable parameters of the TPM networks (N, K, L), the transition from real numbers to quaternions is characterised by an increase in the average process time until the equality of weights is established: $\boldsymbol{W}^{\mathbf{A}} = \boldsymbol{W}^{\mathbf{B}}$.

Probably the most important characteristic of the TPMs synchronisation process based on various algebras is the efficiency or success of third network (**E**) attacks, the peculiarity of which we noted above. Table 4 shows the parameter $\Delta = t_{\text{synch}}/t_{\text{learn}}$, equal to the ratio of the average synchronisation time of networks **A** and **B** (t_{synch}) to the average synchronisation time of **E** and **A** (t_{learn}) with the same parameters N, K, L for each network architecture when implementing a geometric attack, the features of which are described, for example, in [3; 11; 13; 21].

Table 4

Parameter Δ for TPMs using different algebras

Architecture	Parameter Δ
TPM	0.335 53
TPCM	0.107 52
TPQM	0.011 93
TPOM	0.000 22

As can be seen from the above data, the greatest influence on the duration of the synchronisation process has synaptic depth L. However, we probably cannot conclude that with approximately the same parameters N, K, L, the time characteristics of the interaction of NNs are in the same numerical range.

The latest results (see table 4) also mean that, the complexity of the used algebra provides a higher level of system security. This is due to the fact that one of the main factors influencing the security of TPMs is the dimension and number of possible states of network outputs (τ). For example, in the classical TPM architecture with parameter $K = 3$, for each output value there are four internal possible output values of the perceptrons (for example, $(1, 1, 1)$, $(-1, -1, 1)$, $(1, -1, -1)$, $(-1, 1, -1)$ for $\tau = 1$). In the TPCM architecture, the number of possible internal perceptron values per output value increases quadratically. For example, for $K = 3$ there are already 16 possible options (for example, $(1, 1, 1)$, $(-1, -1, 1)$, $(1, -1, -1)$, $(-1, 1, -1)$, $(i, i, -1)$, $(i, -1, i)$, $(-1, i, i)$, $(-i, i, 1)$, $(-i, 1, i)$, $(i, 1, -i)$, $(i, -i, 1)$, $(1, i, -i)$, $(1, -i, i)$, $(-i, -i, -1)$, $(-i, -1, -i)$, $(-1, -i, -i)$ for $\tau = 1$ (for simplicity, we used the canonical notation of complex numbers)).

Conclusions

The article discusses the most important features of mathematical modelling and computer simulation of a secret information exchange system based on two interacting NNs, the architecture of which is named as the tree parity machine. The mathematical basis for the description and analysis of the processes in such a system are hypercomplex numbers. On the basis of generalisation of TPM models for real and complex numbers, as well as quaternions, a TPM model based on octonions has been developed (TPOM).

A feature of the presented research is the implementation of the simulator on different machines interacting based on the TCP/IP protocol. The second distinctive feature is associated with determining the moment and confirming the fact: the state of synchronisation of the two networks has arrived. After this, in the last steps of mutual training of networks (based on the corresponding rule), each side calculates the hash of the current vector of weights and transmits this hash to the other side. Next, each party, by comparing the two hashes, makes a final decision about the end and result of synchronisation. Hash equality means that the parties have achieved an unconditional state of weights equality, which can be used as a joint secret cryptographic key.

Библиографические ссылки

1. Diffie W, Hellman M. New directions in cryptography. *IEEE Transactions on Information Theory.* 1976;22(6):644–654. DOI: 10.1109/TIT.1976.1055638.

2. Kinzel W, Kanter I. Neural cryptography. In: Wang L, Rajapakse JC, Fukushima K, Lee SY, Yao X, editors. *Proceedings of the 9th International conference on neural information processing, 2002. ICONIP'02; 2002 November 18–22; Singapore.* Singapore: Orchid Country Club; 2002. p. 1351–1354. DOI: 10.1109/ICONIP.2002.1202841.

3. Klimov A, Mityagin A, Shamir A. Analysis of neural cryptography. In: Zheng Y, editor. *Advances in Cryptology – ASIACRYPT 2002.* Berlin: Springer; 2002. p. 288–298 (Lecture notes in computer science; volume 2501). DOI: 10.1007/3-540-36178-2_18.

4. Rosen-Zvi M, Kanter I, Kinzel W. Cryptography based on neural networks analytical results. *Journal of Physics A: Mathematical and General.* 2002;35(47):707–713. DOI: 10.1088/0305-4470/35/47/104.

5. Плонковски М, Урбанович ПП. Криптографическое преобразование информации на основе нейросетевых технологий. *Труды Белорусского государственного технологического университета. Серия 6, Физико-математические науки и информатика.* 2005;13:161–164. EDN: YSCTHN.

6. Płonkowski M, Urbanowicz P. Liczby podwójne i ich modyfikacje w neurokryptografii. *Przegląd Elektrotechniczny.* 2002; 88(11b):340–341.

7. Choi Y, Sim J, Kim L-S. CREMON: cryptography embedded on the convolutional neural network accelerator. *IEEE Transactions on Circuits and Systems II: Express Briefs.* 2020;67(12):3337–3341. DOI: 10.1109/TCSII.2020.2971580.

8. Jeong S, Park C, Hong D, Seo C, Jho N. Neural cryptography based on generalized tree parity machine for real-life systems. *Security and Communication Networks.* 2021;11:1–12. DOI: 10.1155/2021/6680782.

9. Sarkar A. Neural cryptography using optimal structure of neural networks. *Applied Intelligence.* 2021;51:8057–8066. DOI: 10.1007/s10489-021-02334-1.

10. Dourlens S. *Neuro-cryptographie appliquée et neuro-cryptanalyse du DES.* Paris: University of Paris; 1995. 218 p. DOI: 10.13140/RG.2.2.35476.24960.

11. Ruttor A. *Neural synchronization and cryptography* [dissertation]. Würzburg: Julius-Maximilians-Universität Würzburg; 2006. 120 p. DOI: 10.48550/arXiv.0711.2411.

12. Плонковски М, Урбанович ПП. Синхронизация криптографических ключей на основе нейронных сетей и в системах криптопреобразования на основе XML. *Труды Белорусского государственного технологического университета. Серия 6, Физико-математические науки и информатика.* 2006;14:152–155. EDN: HSLOUF.

13. Ruttor A, Kinzel W, Kanter I. Dynamics of neural cryptography. *Physical Review E.* 2007;75(5):056104. DOI: 10.1103/PhysRevE.75.056104.

14. Dolecki M, Kozera R. Distribution of the tree parity machine synchronization time. *Advances in Science and Technology.* 2013; 7(18):20–27. DOI: 10.5604/20804075.1049490.

15. Урбанович ПП, Чуриков КВ. Сравнительный анализ методов взаимообучения нейронных сетей в задачах обмена конфиденциальной информацией. *Труды БГТУ. № 6. Физико-математические науки и информатика.* 2010;6:163–166. EDN: TGUDVZ.

16. Seoane LF, Ruttor A. Successful attack on permutation-parity-machine-based neural cryptography. *Physical Review E.* 2012; 85(2):025101. DOI: 10.1103/PhysRevE.85.025101.

17. Shacham LN, Klein E, Mislovaty R, Kanter I, Kinzel W. Cooperating attackers in neural cryptography. *Physical Review E.* 2004; 69(6):066137. DOI: 10.1103/PhysRevE.69.066137.

18. Кантор ИЛ, Солодовников АС. *Гиперкомплексные числа.* Москва: Наука; 1973. 144 с.

19. Płonkowski M, Urbanowicz P, Lisica E. Wykorzystanie kwaternionów w protokole uzgadniania klucza kryptograficznego, opartym na architekturach sieci neuronowych TPQM. *Przegląd Elektrotechniczny.* 2010;86(7):90–91.

20. Dong T, Huang T. Neural cryptography based on complex-valued neural network. *IEEE Transactions on Neural Networks and Learning Systems.* 2020;31(11):4999–5004. DOI: 10.1109/TNNLS.2019.2955165.

21. Zhang Y, Wang W, Zhang H. Neural cryptography based on quaternion-valued neural network. *International Journal of Innovative Computing, Information and Control.* 2022;6(22):1871–1883.

22. Wu J, Xu L, Wu F, Kong Y, Senhadji L, Shu H. Deep octonion networks. *Neurocomputing.* 2019;397:179–191. DOI: 10.1016/j.neucom.2020.02.053.

23. Takahashi K, Fujita M, Hashimoto M. Remarks on octonion-valued neural networks with application to robot manipulator control. In: *2021 IEEE International Conference on Mechatronics (ICM); 2021 March 7–9; Kashiwa, Japan.* [S. l.]: IEEE; 2021. p. 1–6. DOI: 10.1109/ICM46511.2021.9385617.

24. Cariow A, Cariowa G. Fast algorithms for deep octonion networks. *IEEE Transactions on Neural Networks and Learning Systems.* 2023;34(1):543–548. DOI: 10.1109/TNNLS.2021.3124131.

25. Ricotta C, Podani J. On some properties of the Bray – Curtis dissimilarity and their ecological meaning. *Ecological Complexity.* 2017;31:201–205. DOI: 10.1016/j.ecocom.2017.07.003.

References

1. Diffie W, Hellman M. New directions in cryptography. *IEEE Transactions on Information Theory.* 1976;22(6):644–654. DOI: 10.1109/TIT.1976.1055638.

2. Kinzel W, Kanter I. Neural cryptography. In: Wang L, Rajapakse JC, Fukushima K, Lee SY, Yao X, editors. *Proceedings of the 9th International conference on neural information processing, 2002. ICONIP'02; 2002 November 18–22; Singapore.* Singapore: Orchid Country Club; 2002. p. 1351–1354. DOI: 10.1109/ICONIP.2002.1202841.

3. Klimov A, Mityagin A, Shamir A. Analysis of neural cryptography. In: Zheng Y, editor. *Advances in Cryptology – ASIACRYPT 2002.* Berlin: Springer; 2002. p. 288–298 (Lecture notes in computer science; volume 2501). DOI: 10.1007/3-540-36178-2_18.

4. Rosen-Zvi M, Kanter I, Kinzel W. Cryptography based on neural networks analytical results. *Journal of Physics A: Mathematical and General.* 2002;35(47):707–713. DOI: 10.1088/0305-4470/35/47/104.

5. Płonkowski M, Urbanovich PP. Cryptographic transformation of information based on neural network technologies. *Trudy Belorusskogo gosudarstvennogo tekhnologicheskogo universiteta. Seriya 6, Fiziko-matematicheskie nauki i informatika*. 2005;13:161–164. Russian. EDN: YSCTHN.

6. Płonkowski M, Urbanowicz P. Liczby podwójne i ich modyfikacje w neurokryptografii. *Przegląd Elektrotechniczny.* 2002; 88(11b):340–341.

7. Choi Y, Sim J, Kim L-S. CREMON: cryptography embedded on the convolutional neural network accelerator. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2020;67(12):3337–3341. DOI: 10.1109/TCSII.2020.2971580.

8. Jeong S, Park C, Hong D, Seo C, Jho N. Neural cryptography based on generalized tree parity machine for real-life systems. *Security and Communication Networks.* 2021;11:1–12. DOI: 10.1155/2021/6680782.

9. Sarkar A. Neural cryptography using optimal structure of neural networks. *Applied Intelligence.* 2021;51:8057–8066. DOI: 10.1007/s10489-021-02334-1.

10. Dourlens S. *Neuro-cryptographie appliquée et neuro-cryptanalyse du DES.* Paris: University of Paris; 1995. 218 p. DOI: 10.13140/RG.2.2.35476.24960.

11. Ruttor A. *Neural synchronization and cryptography* [dissertation]. Würzburg: Julius-Maximilians-Universität Würzburg; 2006. 120 p. DOI: 10.48550/arXiv.0711.2411.

12. Płonkowski M, Urbanovich PP. Neural network-based cryptographic key synchronization in XML-based cryptographic transformation systems. *Trudy Belorusskogo gosudarstvennogo tekhnologicheskogo universiteta. Seriya 6, Fiziko-matematicheskie nauki i informatika*. 2006;14:152–155. Russian. EDN: HSLOUF.

13. Ruttor A, Kinzel W, Kanter I. Dynamics of neural cryptography. *Physical Review E.* 2007;75(5):056104. DOI: 10.1103/PhysRevE.75.056104.

14. Dolecki M, Kozera R. Distribution of the tree parity machine synchronization time. *Advances in Science and Technology.* 2013; 7(18):20–27. DOI: 10.5604/20804075.1049490.

15. Urbanovich PP, Churikov KV. Comparative analysis of methods for mutual learning of neural networks when solving problems of confidential information exchange. *Trudy BGTU. No. 6. Fiziko-matematicheskie nauki i informatika*. 2010;6:163–166. Russian. EDN: TGUDVZ.

16. Seoane LF, Ruttor A. Successful attack on permutation-parity-machine-based neural cryptography. *Physical Review E.* 2012; 85(2):025101. DOI: 10.1103/PhysRevE.85.025101.

17. Shacham LN, Klein E, Mislovaty R, Kanter I, Kinzel W. Cooperating attackers in neural cryptography. *Physical Review E.* 2004; 69(6):066137. DOI: 10.1103/PhysRevE.69.066137.

18. Kantor IL, Solodovnikov AS. *Giperkompleksnye chisla* [Hypercomplex numbers]. Moscow: Nauka; 1973. 144 p. Russian.

19. Płonkowski M, Urbanowicz P, Lisica E. Wykorzystanie kwaternionów w protokole uzgadniania klucza kryptograficznego, opartym na architekturach sieci neuronowych TPQM. *Przegląd Elektrotechniczny.* 2010;86(7):90–91.

20. Dong T, Huang T. Neural cryptography based on complex-valued neural network. *IEEE Transactions on Neural Networks and Learning Systems.* 2020;31(11):4999–5004. DOI: 10.1109/TNNLS.2019.2955165.

21. Zhang Y, Wang W, Zhang H. Neural cryptography based on quaternion-valued neural network. *International Journal of Innovative Computing, Information and Control.* 2022;6(22):1871–1883.

22. Wu J, Xu L, Wu F, Kong Y, Senhadji L, Shu H. Deep octonion networks. *Neurocomputing.* 2019;397:179–191. DOI: 10.1016/j.neucom.2020.02.053.

23. Takahashi K, Fujita M, Hashimoto M. Remarks on octonion-valued neural networks with application to robot manipulator control. In: *2021 IEEE International Conference on Mechatronics (ICM); 2021 March 7–9; Kashiwa, Japan.* [S. l.]: IEEE; 2021. p. 1–6. DOI: 10.1109/ICM46511.2021.9385617.

24. Cariow A, Cariowa G. Fast algorithms for deep octonion networks. *IEEE Transactions on Neural Networks and Learning Systems.* 2023;34(1):543–548. DOI: 10.1109/TNNLS.2021.3124131.

25. Ricotta C, Podani J. On some properties of the Bray – Curtis dissimilarity and their ecological meaning. *Ecological Complexity.* 2017;31:201–205. DOI: 10.1016/j.ecocom.2017.07.003.

Received 03.11.2023 / revised 27.06.2024 / accepted 27.06.2024.