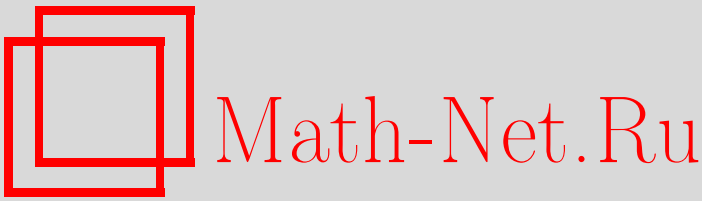


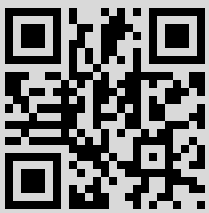
All Russian mathematical portal

D. S. Bogdanov, V. O. Mironkin, Data recovery for a neural network-based biometric authentication scheme, *Mat. Vopr. Kriptogr.*, 2019, Volume 10, Issue 2, 61–74

DOI: https://doi.org/10.4213/mvk284

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use
http://www.mathnet.ru/eng/agreement

Download details:
IP: 109.252.90.35
July 28, 2019, 22:14:25



МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ
2019 Т. 10 № 2 С. 61–74

УДК 519.719.2 DOI https://doi.org/10.4213/mvk284

Data recovery for a neural network-based biometric authentication scheme

D. S. Bogdanov [1], **V. O. Mironkin** [2]

[1] *TVP Laboratories, Moscow*
[2] *National Research University Higher School of Economics, Moscow*

Получено 06.II.2018

Abstract. The project of the standard of neural network biometric containers protection using cryptographic algorithms is analysed. The inconsistency of the suggested combination of password and neural network biometric information protection systems is shown.

Key words: neural network, password, key, biometric authentication system, protection of neural network container

Восстановление данных в схеме биометрической аутентификации, основанной на нейронной сети

Д. С. Богданов [1], **В. О. Миронкин** [2]

[1] *Лаборатории ТВП, Москва*
[2] *Национальный исследовательский университет «Высшая школа экономики», Москва*

Аннотация. Исследован проект стандарта защиты нейросетевых биометрических контейнеров, использующего криптографические алгоритмы. Показана несостоятельность рассмотренного совмещения парольной и нейросетевой биометрической систем защиты информации. Предложен алгоритм, позволяющий восстанавливать ключевую информацию, а также служебную информацию, определяющую процесс функционирования нейронной сети. Получен ряд численных характеристик алгоритма.

Ключевые слова: нейронная сеть, пароль, ключ, биометрическая система аутентификации, защита нейросетевого контейнера

Citation: *Mathematical Aspects of Cryptography*, 2019, v. 10, № 2, pp. 61–74 (Russian)
© Академия криптографии Российской Федерации, 2019 г.

1. Introduction

Biometry is a very attractive tool for user authentication in information and cipher systems [3, 9]. The classic way to implement biometric authentication schemes is to compare the vector of biometric parameters with the existing template and to decide for the user rights according to some criterion. However, this method has a significant disadvantage because it requires a secret keeping of a biometric templates, which makes it difficult to implement such schemes in portable devices [8].

In [2] a concept of fuzzy extractors was introduced. Fuzzy extractors don't require a storage of a confidential data on devices. At the registration phase fuzzy extractors generate a secret vector and a public vector (a so called helper) from person's biometric parameters. The helper together with uploaded biometric parameters should be transformed into a secret vector used for user authentication.

In [2] for some formal model it was proved that the secret information doesn't leak through the helper. Nevertheless, in [1] it was shown that such information could leak in some particular scenarios. So, a construction of secure fuzzy extractors is an open problem. In [7] a good overview of the discussed theme may be found. In [10] a neural network-based approach to the construction of similar schemes is described.

However, later it was shown that some information on a neural network biometric container may be used for effective recovery of confidential data [8]. In order to eliminate this weakness the scheme of a neural network container protection [3] using cryptographic algorithms was proposed in 2018 [12]. However, the results of the present paper show that the proposed scheme doesn't provide the required protection.

2. A neural network-based biometric scheme

Consider a neural network that converts biometric data (fingerprint, retina, handwritten signature, etc.) into some binary sequence which is used for data access or information encryption.

Definition. A neuron is a weighted summator of input parameters

$$x_0, \ldots, x_{N-1}, \quad \text{where} \quad N \in \mathbb{N}$$

and an output of the neuron is based on the Heaviside step function.

Let's introduce the following notation:

N	a number of input biometric parameters, $N \in \mathbb{N}$
$\overline{x}$	a vector of input biometric parameters, $\overline{x} = (x_0, \ldots, x_{N-1})$
m	a number of neurons in a neural network, $m \in \mathbb{N}$
n	a number of inputs of each neuron, $n \in \mathbb{N}$
V_l	a set of all binary vectors of length $l \in \mathbb{N} \cup \{0\}$
V^*	a set of all binary vectors of finite length, $V^* = \bigcup_{l \geqslant 0} V_l$
$h : V^* \to V_l$	a hash function converting vectors of arbitrary finite length into vectors of a length l
$z_{i,j}$	a subvector $(z_i, \ldots, z_j)$ of binary vector $z \in V_l$, $0 \leqslant i < j < l$
$\overline{c}$	a key sequence corresponding to a legitimate user, $\overline{c} = (c_0, \ldots, c_{m-1}), c_i \in \{0, 1\}$
$\overline{u}$	a correspondence table consisting of the neuron inputs defined by external inputs of the network
u_i	the i-th row of the table $\overline{u}$, $i \in \{0, \ldots, \}$
$u_i^{(j)}$	the j-th element of the i-th row of the table $\overline{u}$, $i \in \{0, \ldots, m-1\}$, $j \in \{0, \ldots, n-1\}$
$\overline{w}$	a weight table in which the i-th row contains weights of the biometric parameters x_{u_i} used by i-th neuron
w_i	the i-th row of the table $\overline{w}$, $i \in \{0, \ldots, m-1\}$
$w_i^{(j)}$	the j-th element of the i-th row of the table $\overline{w}$, $i \in \{0, \ldots, m-1\}$, $j \in \{0, \ldots, n-1\}$
s	a fixed parameter "salt", $s \in V_t$, $t \in \mathbb{N}$
p	a password, $p \in V_r$, $r \in \mathbb{N}$
$]v[$	$]v[= \min\{n \in \mathbb{Z} : n \geqslant v\}$ for $v \in \mathbb{R}$
$a \| b$	a concatenation of vectors $a, b \in V^*$

Let's consider the analysed neural network [12] which consists of m neurons, each having n inputs. Suppose that the biometric data is encoded by parameters $x_0, \ldots, x_{N-1}$.

The transformation of the correspondence table and the weight table after the stage of a neural network training [4] is organized in such a way that for a legitimate biometric sample the i-th bit of the key sequence is calculated by the formula

$$c_i = Z\left(\sum_{j=0}^{n-1} w_i^{(j)} x_{u_i^{(j)}}\right),$$

where Z is the Heaviside step function, and for any other biometric samples the corresponding bits are supposed to be equiprobable and independent.

The general scheme of the key sequence generation based on the input parameters $\overline{x}$ is shown in Fig. 1.

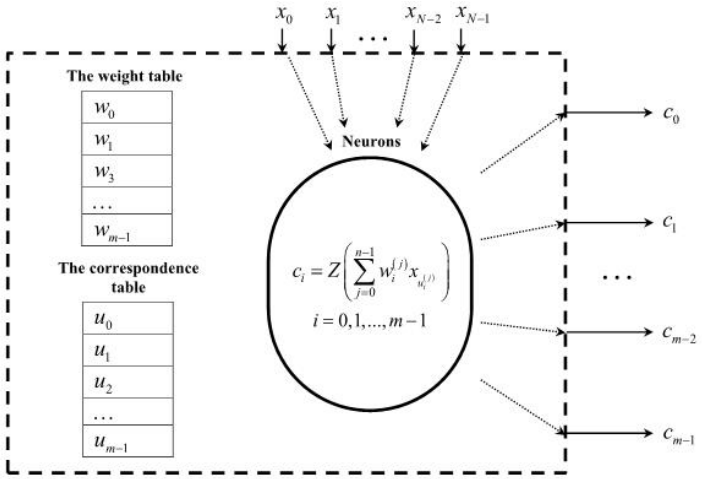


Fig. 1. The generation of the key sequence $\overline{c}$

Let's describe a purpose of some functional elements of the neural network [12]:

1. The binary vector of length $b =]\log_2 N[$ is used to encode rows of $\overline{u}$. If $\log_2 N \notin \mathbb{N}$, a calculation of used biometric parameter number is determined by the formula $\alpha' = \alpha \,(\bmod N)$, where α is a row number of $\overline{u}$ and α' is a biometric parameter number.
2. The set of integer values in the range $[-2^{d-1}, 2^{d-1} - 1]$ is used to encode rows of $\overline{w}$, where $d \geqslant 8$.
3. The binary sequence $\{\gamma_i\}_{i=0}^{m-1}$ is used to encrypt rows of the correspondence table and the weight table. Components of this sequence are formed for each neuron separately by the hash function h defined in [5]:

$$\gamma_i = \left(h_i^{(1)} \| h\left(h_i^{(1)}\right) || \ldots \| h^{k-1}\left(h_i^{(1)}\right)\right)_{0,n(b+d)-1}, \tag{1}$$

where $k = \left]\frac{n(b+d)}{l}\right[$ and the components γ_i are defined as follows:

$$h_0^{(1)} = h\,(s \| p \| 0)\,, \tag{2}$$

$$h_i^{(1)} = h\,(s \| p \| i \| c_0, \ldots, c_{i-1})\,, \quad i > 0. \tag{3}$$

The network encryption is based on an addition modulo 2 of γ_i with a concatenation of the corresponding rows of tables $\overline{u}, \overline{w}$: $E_i = (u_i \| w_i) \oplus \gamma_i$.

Remark 1. According to the scheme [12] the values $E_i,\ i = 0, \ldots, m-1,$ are stored in a neural network biometric container instead of tables $\overline{u}$ and $\overline{w}$.

The rows of the correspondence table and the weight table are successively decrypted according to the following rule:

1. For $i = 0$ we calculate the rows

$$\begin{aligned} u_0 &= (E_0 \oplus \gamma_0)_{0,bn-1}, \\ w_0 &= (E_0 \oplus \gamma_0)_{bn,n(b+d)-1}, \end{aligned} \tag{4}$$

   and form the bit $c_0,$ where γ_0 is determined by relations (1) and (2).

2. For $i \in \{1, \ldots, m-1\}$ we calculate the next rows using the bit sequence $(c_0, \ldots, c_{i-1})$ formed on the previous steps:

$$\begin{aligned} u_i &= (E_i \oplus \gamma_i)_{0,bn-1}, \\ w_i &= (E_i \oplus \gamma_i)_{bn,n(b+d)-1}, \end{aligned} \tag{5}$$

   where γ_i is determined by relations (1) and (3).

3. A recovery of bits of the key

In this section we propose a method of recovery of the password, the key sequence $\overline{c}$ and the encrypted tables $\overline{u}$ and $\overline{w}$ using only the analysed neural network data [12] without any additional input biometric information.

According to the standard [4] a neural network training satisfies the following conditions:

1. Any four successive rows of the table $\overline{u}$ consist of $4n$ different elements.
2. $\forall i, j \in \{0, \ldots, N-1\}$ a number of usages of x_i differs from a number of usages of x_j by no more than 2 in the table $\overline{u}$.

Remark 2. In further reasoning we will assume that the hash function $h : V^* \to V_l$ is an equiprobable random mapping.

For rows of the table $\overline{u}$ define the following event

$$A_{i,j}(\overline{u}) = \left\{ \left| \left\{ u_i^{(0)}, \ldots, u_i^{(n-1)}, \ldots u_j^{(0)}, \ldots, u_j^{(n-1)} \right\} \right| = (j-i+1)\, n \right\},$$

where $0 \leqslant i \leqslant j \leqslant m-1$. Then the following result is true.

Lemma 1. *Let* $\overline{c} \in V_m$, $m \geqslant 2$, *be the key sequence formed by the neural network* [12] *with* $4n < N$. *Let* $\overline{c}' \in V_j$, $j \in \{1, \ldots, m-1\}$, *be a sequence such that* $c'_i = c_i$, $i = \{0, \ldots, j-2\}$, $j > 1$, *and the last bit* c'_{j-1} *is chosen randomly and equiprobably. Then for the table* $\overline{u'}$ *such that* $u'_0 = u_0, \ldots, u'_{j-1} = u_{j-1}$ *and the row* u'_j *defined by the rule* (5)

$$\mathbf{P}\left\{A_{\max(0,j-3),j}\left(\overline{u'}\right) \mid c'_{j-1}=c_{j-1}\right\} = 1,$$
$$\mathbf{P}\left\{A_{\max(0,j-3),j}\left(\overline{u'}\right) \mid c'_{j-1}\neq c_{j-1}\right\} = \frac{(N-n\cdot\min(3,j))_n}{2^{nb}},$$

where $(n)_k = n(n-1)\ldots(n-k+1)$ *is the falling factorial.*

Proof. In the case $c'_{j-1} = c_{j-1}$ we have the equalities

$$h\big(s\|p\|j+1\|c_0,\ldots,c_{j-2},c'_{j-1}\big) = h\big(s\|p\|j+1\|c_0,\ldots,c_{j-2},c_{j-1}\big) \quad \text{for} \quad j>1$$

and

$$h\big(s\|p\|1\|c'_0\big) = h\big(s\|p\|1\|c_0\big) \quad \text{for} \quad j=1.$$

So, using expressions (4), (5), we get $u'_j = u_j$. According to the standard [4] the first statement of lemma is obvious.

Let $c'_{j-1} = c_{j-1} \oplus 1$. According to the remark 2 for the false vector $(c_0, \ldots, c_{j-2}, c'_{j-1})$ the hash value $h(s\|p\|j+1\|c_0,\ldots,c_{j-2},c'_{j-1})$ for $j > 1$ and the hash value $h(s\|p\|1\|c'_0)$ for $j = 1$ have equiprobable distribution on the set of images. So, from (1) it follows that the corresponding value γ'_j has equiprobable distribution on $V_{n(b+d)}$ and analogously the row $u'_j = (E_j \oplus \gamma'_j)_{0,bn-1}$ has equiprobable distribution on V_{nb}.

Under the conditions of the lemma it follows that $\mathbf{P}\{A_{\max(0,j-3),j-1}(\overline{u'})\}=1$. Then for $4n < N$ we obtain

$$\mathbf{P}\{A_{0,1}\left(\overline{u'}\right) \mid c'_0 \neq c_0\} = \frac{(N-n)_n}{2^{nb}},$$

$$\mathbf{P}\{A_{0,2}\left(\overline{u'}\right) \mid c'_1 \neq c_1\} = \frac{(N-2n)_n}{2^{nb}},$$

$$\mathbf{P}\{A_{j-3,j}\left(\overline{u'}\right) \mid c'_{j-1} \neq c_{j-1}\} = \frac{(N-3n)_n}{2^{nb}}, \qquad j \geqslant 3. \qquad \square$$

Example 1. Under the conditions of the lemma 1 for the considered neural network authentication model with parameters $N = 416,\ n = 16$ (see for example [6]) we have

$$\mathbf{P}\{A_{0,1}\left(\overline{u'}\right) \mid c'_0 \neq c_0\} \approx 1.4 \cdot 10^{-2},$$

$$\mathbf{P}\{A_{0,2}\left(\overline{u'}\right) \mid c'_1 \neq c_1\} \approx 7.3 \cdot 10^{-3},$$

$$\mathbf{P}\{A_{(j-3),j}\left(\overline{u'}\right) \mid c'_{j-1} \neq c_{j-1}\} \approx 3.6 \cdot 10^{-3},\ j \geqslant 3.$$

For the same scheme [12] with parameters $N = 416,\ n = 32$

$$\mathbf{P}\{A_{0,1}\left(\overline{u'}\right) \mid c'_0 \neq c_0\} \approx 2.7 \cdot 10^{-5},$$

$$\mathbf{P}\{A_{0,2}\left(\overline{u'}\right) \mid c'_1 \neq c_1\} \approx 1.5 \cdot 10^{-6},$$

$$\mathbf{P}\{A_{(j-3),j}\left(\overline{u'}\right) \mid c'_{j-1} \neq c_{j-1}\} \approx 5.9 \cdot 10^{-8},\ j \geqslant 3.$$

3.1. A recovery algorithm

Here is the probabilistic algorithm of password and key sequence recovery based on structural features [4] and lemma 1.

Algorithm A

1. *Set* $C_0 = \cdots = C_{m-2} = M = \varnothing$.
2. *For* $p \in V_r \backslash M$ *calculate the row* u_0 *according to the rule* (4).
3. *For each value* $c_0 \in \{0,1\}$ *calculate the rows* u_1 *according to the rule* (5). *If event* $A_{0,1}$ *occurs, supplement* C_0 *by the corresponding bit* c_0. *If* $C_0 = \varnothing$, *supplement* M *by* p *and go to step 2.*
4. *For each values* $c_1 \in \{0,1\}$ *calculate the row* u_2. *If event* $A_{0,2}$ *occurs, supplement* C_1 *by the corresponding bit* c_1. *If* $C_1 = \varnothing$, *supplement* M *by* p *and go to step* 2.
5. *Set* $i = 3$.
6. *If* $i \geqslant m - 1$, *the algorithm finishes its work, otherwise for each value* $c_{i-1} \in \{0,1\}$ *calculate the row* u_i. *If event* $A_{(i-3),i}$ *occurs, supplement* C_{i-1} *by the corresponding bit* c_{i-1}.

   If $C_{i-1} = \varnothing$, *supplement* M *by* p *and go to step* 2, *otherwise set* $i = i+1$ *and go to step* 6.

The algorithm A forms the set of possible key sequences:

$$C = \{(c_0, c_1, \ldots, c_{m-2}) \,|c_i \in C_i,\ i = \{0, \ldots, m-2\}\}\,.$$

Remark 3. The result of [8] allows to define the last bit c_{m-1} of the true key sequence $c_0, \ldots, c_{m-1}$ and the vector of input biometric parameters $x_0, \ldots, x_{N-1}$.

3.2. Analysis of the algorithm A

Denote by W the success of the algorithm A consisting in forming only the true sequence $c_0, c_1, \ldots, c_{m-2}$ and password p. So, to calculate $\mathbf{P}\{W\}$, we prove several auxiliary statements.

Let $q_i = \mathbf{P}\left\{A_{\max(0,i-3),i}\left(\overline{u'}\right) \mid c'_{i-1} \neq c_{i-1}\right\}, \quad i \in \{1, \ldots, m-1\}$. Consider the binary tree of height i, all vertices of which appear independently with probability q_3 each (Fig. 2).

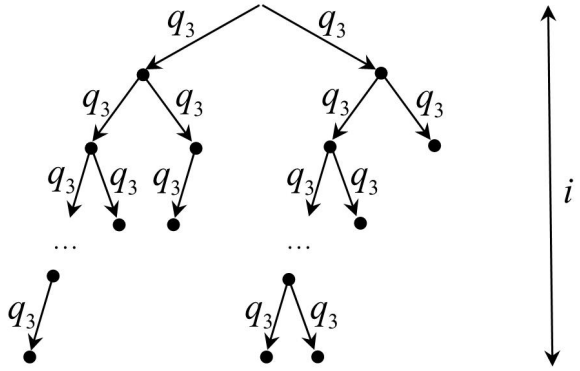


Fig. 2. Example of the binary tree of height i with probability q_3

Let P_i be the probability of existence of this tree. It isn't difficult to show that

$$\begin{cases} P_0 = 1, \\ P_i = 2q_3 P_{i-1} - q_3^2 P_{i-1}^2, \quad i \geqslant 1. \end{cases} \tag{6}$$

Further consider the tree with vertices corresponding to the values of the testing bits $c_0, c_1, \ldots, c_{m-2}$ in the algorithm A. The root of the tree has mark u_0 formed as a result of password search and determines the further formation of the elements $c_0, c_1, \ldots, c_{m-2}$. Herewith, all vertices of i-th layer are marked by the corresponding values of the testing bits $c_{i-1},\ i = 1, \ldots, m-1$. So, the set C contains only sequences corresponding to branches of length $m-1$.

In particular, consider subtree corresponding to the true password p (Fig. 3). In this case the vertices of the i-th layer are formed with probabilities 1 or q_i respectively. Herewith, $q_i = q_3$ for $i \geqslant 3$.

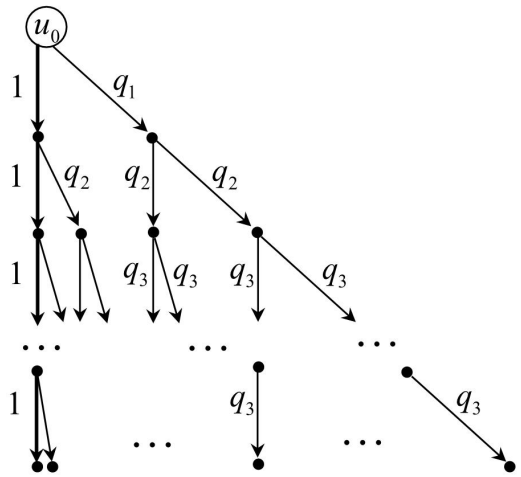


Fig. 3. The scheme of the algorithm A for a true p

Let B_i, $i \in \{1, \ldots, m-1\}$, be an event that for the true password p the i-th layer of the subtree has at least one vertex in addition to the true vertex c_{i-1}.

Lemma 2. *For* B_i, $i \in \{1, \ldots, m-1\}$, *for the neural network model* [12] *the following relations are true*:

$$\begin{cases} \mathbf{P}\{B_1\} &= q_1, \\ \mathbf{P}\{B_2\} &= q_2 + 2q_1q_2 - 3q_1q_2^2 + q_1q_2^3, \\ \mathbf{P}\{B_{i+2}\} &= \alpha_i + (1-\alpha_i)\left(q_2P_i + q_1\left(1 - q_2P_i\right)\left(2q_2P_i - q_2^2P_i^2\right)\right), \\ & \qquad\qquad i \in \{1, \ldots, m-3\}, \end{cases} \tag{7}$$

where values α_i *are defined as follows*:

$$\begin{cases} \alpha_1 = q_3, \\ \alpha_i = \alpha_{i-1} + (1-\alpha_{i-1})\, q_3P_{i-1}, \quad i \in \{2, \ldots, m-1\}. \end{cases}$$

Now consider subtree with vertices corresponding to values of testing bits $c_0, c_1, \ldots, c_{m-2}$ in the algorithm A in the case of a false password p (Fig. 4).

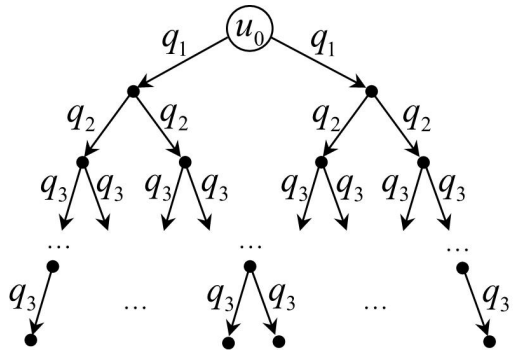


Fig. 4. The scheme of the algorithm A for a true p

Let D_i, $i \in \{1, \ldots, m-1\}$, be an event that for a false p the i-th layer of the corresponding subtree has at least one vertex.

Lemma 3. *For* D_i, $i \in \overline{1, m-1}$, *for the neural network model* [12] *the following relations are true*:

$$\begin{cases} \mathbf{P}\{D_1\} = 2q_1 - q_1^2, \\ \mathbf{P}\{D_{i+2}\} = 2q_1\left(2q_2P_i - q_2^2P_i^2\right) - q_1^2\left(2q_2P_i - q_2^2P_i^2\right)^2, \\ i \in \overline{0, m-3}. \end{cases} \tag{8}$$

Using the results formulated above we proceed to calculate the probability $\mathbf{P}\{W\}$.

Proposition 1. *For the algorithm A of key and password recovery in the neural network model* [12] *for any* $m \in \{2, 3, \ldots\}$ *the following formula is true*:

$$\mathbf{P}\{W\} = \frac{1 - (1 - \mathbf{P}\{D_{m-1}\})^{2^r}}{2^r \mathbf{P}\{D_{m-1}\}}\left(1 - \mathbf{P}\{B_{m-1}\}\right), \tag{9}$$

where values $\mathbf{P}\{B_{m-1}\}$ *and* $\mathbf{P}\{D_{m-1}\}$ *are defined by relations* (7) *and* (8) *respectively.*

Proof. Let the event W_k consists in the recovery the true sequence and password for the first time at the k-th password value test, $k \in \{1, \ldots, 2^r\}$. Then the following equation holds:

$$\mathbf{P}\{W\} = \sum_{k=1}^{2^r} \mathbf{P}\{W_k\}. \tag{10}$$

For the true password value the algorithm A completes its work at the corresponding step with probability $1 - \mathbf{P}\{B_{m-1}\}$ and for a false password value the algorithm A rejects the corresponding value with probability $1 - \mathbf{P}\{D_{m-1}\}$. So, for a fixed value $k \in \{1, \ldots, 2^r\}$ we have

$$\begin{aligned} \mathbf{P}\{W_k\} &= \big(1 - \mathbf{P}\{D_{m-1}\}\big)^{k-1} \left(\prod_{i=0}^{k-2} \left(1 - \frac{1}{2^r - i}\right)\right) \frac{1 - \mathbf{P}\{B_{m-1}\}}{2^r - k + 1} = \\ &= \frac{1}{2^r} \big(1 - \mathbf{P}\{D_{m-1}\}\big)^{k-1} \big(1 - \mathbf{P}\{B_{m-1}\}\big). \end{aligned} \tag{11}$$

So, substituting (11) into (10) we obtain

$$\begin{aligned} \mathbf{P}\{W\} &= \frac{1}{2^r} \left(1 - \mathbf{P}\{B_{m-1}\}\right) \sum_{k=1}^{2^r} \left(1 - \mathbf{P}\{D_{m-1}\}\right)^{k-1} = \\ &= \frac{1 - \left(1 - \mathbf{P}\{D_{m-1}\}\right)^{2^r}}{2^r \mathbf{P}\{D_{m-1}\}} \left(1 - \mathbf{P}\{B_{m-1}\}\right). \end{aligned}$$

□

Corollary. *For the algorithm A of key and password of the neural network model* [12] *recovery for any* $m \in \{3, 4, \ldots\}$ *the following lower bound for* $\mathbf{P}\{W\}$ *is true*:

$$\mathbf{P}\{W\} > \left(1 - 2^{m+r-2} q_1 q_2 q_3^{m-3}\right) \left(1 - \mathbf{P}\{B_{m-1}\}\right). \tag{12}$$

Proof. In order to estimate the expression (9) from below we consider the two-sided inequality [11] valid for arbitrary $x \in [-1, 1]$ and $k \in \mathbb{N}$:

$$1 - kx \leqslant (1 - x)^k \leqslant 1 - kx + C_k^2 x^2,$$

Since $0 < \mathbf{P}\{D_{m-1}\} < 1$ then using inequalities $\mathbf{P}\{D_{m-1}\} > 4q_1q_2P_{m-3}$ and $P_{m-3} \geqslant (2q_3)^{m-3}$ for $m \geqslant 3$ we obtain

$$\begin{aligned}
\mathbf{P}\{W\} &= \frac{1-(1-\mathbf{P}\{D_{m-1}\})^{2^r}}{2^r\mathbf{P}\{D_{m-1}\}}\,(1-\mathbf{P}\{B_{m-1}\}) \geqslant \\
&\geqslant \frac{2^r\mathbf{P}\{D_{m-1}\} - C_{2^r}^2\mathbf{P}\{D_{m-1}\}^2}{2^r\mathbf{P}\{D_{m-1}\}}\,(1-\mathbf{P}\{B_{m-1}\}) > \\
&> \left(1-2^{r-1}\mathbf{P}\{D_{m-1}\}\right)(1-\mathbf{P}\{B_{m-1}\}) > \\
&> \left(1-2^{r+1}q_1q_2P_{m-3}\right)(1-\mathbf{P}\{B_{m-1}\}) \geqslant \\
&\geqslant \left(1-2^{m+r-2}q_1q_2q_3^{m-3}\right)(1-\mathbf{P}\{B_{m-1}\}).
\end{aligned}$$

□

Example 2. For the scheme [6] with $m = 256$, $n = 16$ and $r = 64$ the corollary gives the following lower bound for probability of success of algorithm A

$$\mathbf{P}\{W\} > 0.996.$$

For the scheme [12] with $m = 256$, $n = 32$ and $r = 64$ we have

$$\mathbf{P}\{W\} > 1 - 5.9 \cdot 10^{-8}.$$

Let T be a random variable equal to the complexity of the algorithm A at random choice of the password measured by the number of hash function h calculations.

Proposition 2. *For the algorithm A of key and password of the neural network model* [12] *recovery for any* $m \in \{5, 6, \ldots\}$ *and* $k \in \{2, \ldots, m\}$ *the following inequality is true*:

$$\mathbf{P}\{T \leqslant 2^{r+k} + 2m\} > \delta\left(1-\mathbf{P}\{D_{k-1}\}\right)^{2^r-1},$$

where

$$\delta = (1-q_3)^{m-2}\left((1-q_1)\left(1+(m-5)\,q_3\,(1-q_2)\right) + q_1\,(1-q_2)^3\right).$$

Proof. In the case of a true password p we have

$$\begin{aligned}
&\mathbf{P}\{T \leqslant 2m \,|\, p \text{ is true}\} = \\
&= (1-q_1)(1-q_2)(1-q_3)^{m-5} + (1-q_1)\,q_2\,(1-q_3)^{m-2} + \\
&\quad + (m-5)(1-q_1)(1-q_2)\,q_3\,(1-q_3)^{m-3} + q_1\,(1-q_2)^3\,(1-q_3)^{m-4} > \\
&> (1-q_3)^{m-2}\left((1-q_1)(1+(m-5)\,q_3\,(1-q_2)) + q_1\,(1-q_2)^3\right) = \delta.
\end{aligned}$$

Taking into account (6), it is easy to show that for a false password p the following inequalities hold for $k \in \{2, \ldots, m\}$:

$$\begin{aligned}
\mathbf{P}\left\{T \leqslant 2^k \,\middle|\, p \text{ is false}\right\} &> \mathbf{P}\left\{T \leqslant 2^k - 2 \,\middle|\, p \text{ is false}\right\} > \\
&> 1 - \mathbf{P}\left\{D_{k-1}\right\}.
\end{aligned}$$

In the worst case a number of password p checkings by the algorithm A is 2^r. Thus, using the independence of checks of bits $c_0, c_1, \ldots, c_{m-2}$ for different values of p we obtain

$$\mathbf{P}\left\{T \leqslant 2m + 2^k\,(2^r - 1)\right\} > \delta\left(1 - \mathbf{P}\left\{D_{k-1}\right\}\right)^{2^r-1}.$$

This inequality implies the stated relation. □

Example 3. For the scheme [6] with $m = 256,\ n = 16,\ r = 64$ the proposition 2 for $k = 10$ allows to obtain the following estimate

$$\mathbf{P}\left\{T \leqslant 1.9 \cdot 10^{22}\right\} > 0.753.$$

So, for the scheme [12] with $m = 256,\ n = 32,\ r = 64$ and $k = 3$ we have

$$\mathbf{P}\left\{T \leqslant 1.5 \cdot 10^{20}\right\} > 1 - 1.8 \cdot 10^{-7}.$$

Remark 4. The algorithm A allows to reduce the problem of password and key sequences recovery to the problem of testing password values with a maximum complexity $O\,(2^r)$.

4. Conclusions

The proposed method of the key information recovery doesn't require any knowledge of the biometric data and allows to recover all parameters determining the neural network, including the biometric template. The obtained results showed that the key sequence recovery is equivalent to the password recovery.

It should be noted that the considered protection scheme of a neural network container doesn't provide any protection against the proposed method regardless of the used encryption rules (5). Therefore, protection schemes like [12] are unsafe.

In conclusion the authors thank A.M. Zubkov for his interest to the paper and helpful comments.

References

[1] Boyen X., "Reusable fuzzy extractors", CCS'04 Proc. 11th ACM Conf. Computer and communic. security, 82–91

[2] Dodis Y., Reyzin L., Smith A., "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data", *Lect. Notes Comput. Sci.*, **3072** (2004), 523–540

[3] *GOST R 52633.0-2006. Information protection. Information protection technology. Requirements to the means of high-reliability biometric authentication*, M.: Standardinform, 2007, 24 pp. (in Russian)

[4] *GOST R 52633.5-2011. Information protection. Information protection technology. The neural net biometry-code convertor automatic training*, M.: Standardinform, 2012, 20 pp. (in Russian)

[5] *GOST R 34.11-2012. Information technology. Cryptographic data security. Hash function*, M.: Standardinform, 2013, 24 pp. (in Russian)

[6] Efimov O. V., Funtikov V. A., Jazov Y. K., "The neuronet converter "biometry-code" structure and interconnections choice strategy", *Neurocomputers: development, application*, № 6 (2009), 14–16 (in Russian)

[7] Kevenaar T., Skoric B., Tuyls P., *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, Heidelberg etc.: Springer Science and Business Media, 2007, 340 pp

[8] Marshalko G. B., "On the security of a neural network-based biometric authentication scheme", *Matematicheskie voprosi kriptografii*, **5**:2 (2014), 87–98

[9] Marshalko G. B., Mironkin V. O., "Geometric analysis of a neural symmetric cipher", *Information Security Problems. Computer Systems*, № 1 (2017), 43–49 (in Russian)

[10] Nazarov I. G., Efimov O. V., Yazov Y. K., "The package of national standards, ensuring biometric and neural network protection of mass circulated personal data privacy", *Neurocomputers: development, application*, № 3 (2012), 9–17 (in Russian)

[11] Sachkov V. N., *Probabilistic methods in combinatorial analysis*, M.: Nauka, 1978, 288 pp. (in Russian)

[12] *Technical specification* (*project*). *Neural network container protection using cryptographic algorithms*, 2018, 11 pp. (in Russian), https://tc26.ru/discussions/tekhnicheskaya-spetsifikatsiya-zashchita-neyrosetevykh-biometricheskikh-konteynerov-s-ispolzovaniem-.html.