Authentication and key distribution protocol based on Diffie-Hellman algorithm and physically unclonable functions

Victor A. Yakovlev[1], Dina Zh. Satybaldina[2], Eldor Egamberdiyev[2], Yerzhan Seitkulov[2]
[1]Department of Secured Communications Systems, Faculty of Infocommunication Networks and Systems, Bonch-Bruevich Saint Petersburg State University of Telecommunications, Saint Petersburg, Russian Federation
[2]Department of Information Security, Faculty of Information Technologies, L.N. Gumilyov Eurasian National University, Astana, Republic of Kazakhstan

Article Info

Article history:

Received Sep 8, 2023
Revised Dec 22, 2023
Accepted Jan 5, 2024

Keywords:

Authentication
Cellular communication networks
Diffie-Hellman
Key distribution
Physically unclonable function

ABSTRACT

Based on the modified Diffie-Hellman (DH) protocol, a key distribution scheme between two correspondents over open communication channels is considered. The correspondents communicate through a trusted entity. An attacker can control the communication channels between the correspondents and the channels between the correspondents and the trusted authority (TA) and perform active attacks there, including a man-in-the-middle attack. DH authentication protocol using physically unclonable functions (PUF) is proposed. A formalized PUF model based on the class of universal hash functions is presented. Namely, it is proposed to use the class of strictly universal hash functions developed by Wegman and Carter. A polynomial dependence of the possible number of PPUs on the number of answers has been proven. Requirements for PPUs suitable for authentication systems are formulated. The protocol has been analyzed, and its security has been proved.

Corresponding Author:

Yerzhan Seitkulov
Department of Information Security, Faculty of Information Technologies
L.N. Gumilyov Eurasian National University
2 Satpayev str., Astana, 010000, Republic of Kazakhstan
Email: yerzhan.seitkulov@gmail.com

1. INTRODUCTION

The 6G mobile network will consist of heterogeneous nodes, from macro-level devices (satellites) to autonomous vehicles and intelligent infrastructure sensors [1]. This network heterogeneity and a significant increase in coverage can reduce the security and privacy of users of 6G networks compared to previous generations of mobile communications. Potential losses from security incidents can be critical concerning personal information, finances, health, and even the life of network subscribers if, for example, attacks on unmanned transport systems are implemented, leading to mass traffic accidents, including fatalities [2]. 6G security mechanisms will be based on symmetric and asymmetric cryptography in the context of quantum computing development [3]. Providing security against quantum computing can reduce the effectiveness of these mechanisms.

One of the promising technologies for increasing quantum stability in the public key cryptography model is the use of quantum key distribution [4]. However, due to the high cost, it is still challenging to implement a quantum network around the world. Another new method uses quantum-safe hybrid key exchange mechanisms based on the theory that a cryptosystem will remain secure if one of its key exchange methods remains secure [5]. As an example of such an approach, it is proposed to combine a classical key exchange method, such as the Diffie-Hellman (DH) scheme, and a quantum-safe key encapsulation mechanism [6].

The DH method allows a common encryption key (DH key) to be formed over the hackable communication channel to establish a secure connection between the two correspondents [7]. This method is commonly used in network protocols secure sockets layer/transport layer security (SSL/TLS), IPsecurity (IPSec), pretty good privacy privacy (PGP), and other applications. The DH algorithm security cannot be compromised because some network protocols and services depend upon DH key exchange for reliable communication. Therefore, many researchers propose various ways to modify the DH scheme to make this algorithm more resistant to attacks and more effective for new applications, for the internet of things (IoT), cloud systems, and new generation cellular communications. The user registration phase, integrated with DH key exchange and random key generation, is the core of the proposed authenticated key management scheme (AKMS) [8]. The AKMS scheme guarantees confidentiality in transferring keys between users using two keys and encryption. First, the server generates a random key to encrypt the file before transmission. The user then encrypts a random key using a key generated by the DH key exchange. The authors of the work [9] also apply integration with the classical protocol and propose a secure and efficient routing protocol (RPL) for IoT networks. To secure this powerful new RPL protocol and guarantee authentication and data integrity, nodes must have a shared secret key calculated using the new advanced DH algorithm. We considered various ways to store data in the cloud for a given time using several cryptographic solutions, including the DH key distribution protocol [10].

A DH key digital signature is one of the directions for solving the key authentication task [11]. The digital signature is verified using the open key distributed in the network utilizing a certificate. This approach is used in SIGMA protocol [12], the base for the internet key exchange (IKE) protocol v.2, and requires public key infrastructure (PKI). Another approach to solving the authentication task for the key distributed using the DH method is to use binary sequences previously distributed among users. The users develop these sequences while pairing their mobile devices in a face-to-face meeting [13]. The eavesdropper is removed from the users and does not have access to the sequences that are exchanged between the users. The users cannot directly use the generated sequences as encryption keys because these sequences contain a certain percentage of errors (misalignments). This approach is studied in detail in the paper [14]. The peculiarity of the approach is that users need to connect their mobile devices to obtain almost identical sequences and use them efficiently to select a key using the DH method. This method is preferably oriented to mobile devices such as smartphones.

A physically unclonable function (PUF) is a property of a physical (digital) system that cannot be cloned (reproduced, copied) in other physical systems [15]. PUFs owe their unclonability to the fact that they consist of several random components in the production process and cannot be controlled. Due to random parameters, each digital system can be treated as unique and physically unconnected. The PUFs is based on extracting unique parameters from digital systems. PUFs have gained great popularity in the last 10-15 years in solving various cybersecurity problems and, primarily, in solving authentication tasks [16].

In this paper, we consider a method for authenticating a key generated by the DH protocol, with the participation of a trusted center (TC) and using PUF. We have proposed two options for key distribution protocols among network users with hardware implementation of a PUF in their devices. The trusted authority (TA) has a database of request-response value pairs for each user’s PUF. The main contributions of this work are highlighted:

a. We have formulated requirements for PUFs suitable for authentication systems.
b. We develop two variants of DH key authentication protocols using arithmetic operations and PUFs.
c. We analyze and evaluate the security of the proposed protocols.

The rest of the paper is organized as follows. The next part briefly describes the standart scheme of DH algorithm and analyses the PUF construction principles, their features and the models used to formalize their parameters. The third section explains the approach to building DH key authentication systems based on PUF and TA presents developed authentication protocol variants using arithmetic operations and hash functions. Analyze and evaluate results for the security of the proposed protocols are present in the fourth part. The conclusion summarizes the work and points out promising directions for further research.

2. THE COMPREHENSIVE THEORETICAL BASIS

2.1. The existing scheme of DH algorithm and its vulnerabilities

Let us assume that Alice and Bob exchange information over a network using the standard DH key exchange process [7]. Network users Alice (A) and Bob (B) agree on the parameters p and g, where p is a prime number and g is an element of a finite field GF(p), which generates a group having a high group, and the following protocol is executed.

a. Alice generates an element of the field $x \in (1, p-1)$, she computes $X = g^x(mod\, p)$, and sends it to Bob.
b. Bob generates an element $y \in (1, p-1)$, he computes it $Y = g^y(mod\, p)$ and sends it to Alice.
c. Alice computes a key $K_A = Y^x(mod\, p)$.
d. Bob computes a key $K_B = X^y(mod\, p)$.

Keys computed by Alice and Bob are equal $K_A = g^{yx}(mod\, p) = K_B = g^{xy}(mod\, p) = K_s$. Cryptographic protocols based on cryptographic algorithms can provide a high level of security. But cryptographic protocols can be compromised by vulnerabilities such as man-in-the-middle (MITM) attacks in areas of remote user interaction [17]. An overview of MITM attacks targeting the DH protocol was provided [18]. Eavesdropper eve can record the messages that will be sent from Alice to Bob, and she can later send a copy of the messages to Bob. Bob will assume that these messages come from Alice. Eve can then send her messages to Alice, who would believe that they came from Bob. Many researchers have proposed defenses against this type of attack. The most well-known approaches are digital signatures and message authentication codes [18]. Although they are convenient for many systems, they still have some weaknesses. The proposed paper aims to distribute keys between Alice and Bob without compromising them with Eve. Consider the MITM attack in more detail see Figure 1.

a. User A generates a random number $x \in [1, p-1]$, calculates $X_A = g^x\, mod\, p$ and sends the obtained value to the correspondent B.
b. User B generates a random number $y \in [1, p-1]$, calculates value $Y_B = g^y\, mod\, p$and sends the obtained value to the correspondent A.

- Eavesdropper E intercepts X_A, saves it in the memory, generates a random number $e \in [1, p-1]$, finds $Y_E = g^e\, mod\, p$ and sends it to the user A under the guise of the user B.
- The eavesdropper E intercepts Y_B, saves it in the memory, generates a random number $e' \in [1, p-1]$, finds $X_E = g^{e'}\, mod\, p$ and sends it to the user B under the guise of the user A.

a. User A calculates the session key value:

$$K_A = (Y_E)^x\, mod\, p = g^{ex}\, mod\, p$$

- The eavesdropper finds the key to communication with A.

$$K_E = (X_A)^e\, mod\, p = g^{xe}\, mod\, p$$

b. User B calculates the session key value:

$$K_B = (X_E)^y\, mod\, p = g^{e'y}\, mod\, p$$

- The eavesdropper finds the key to communication with B.

$$K_E' = (Y_B)^{e'}\, mod\, p = g^{ye'}\, mod\, p$$

It is obvious that $K_A = K_E$ and $K_B = K_E'$.

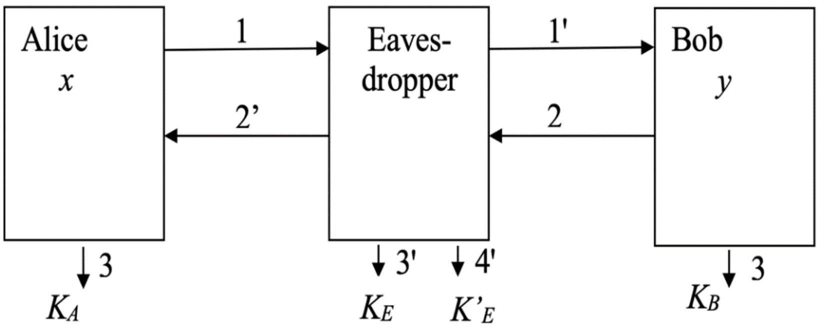


Figure 1. MITM attack for the DH algorithm

Thus, the eavesdropper E has formed a common key K_E with the User A and a common key $K_E^{'}$ with the user B. If the user A sends an encrypted message on the key K_A, then the eavesdropper will decipher this message on the key K_E and will re-encrypt on the key $K_E^{'}$. Then he will send cryptography to the user B, who will decipher it on the key K_B. At the same time, the user A believes that he directly works with the user B, and the user B thinks that he works directly with the user A. The actual exchange between the users A and B is supervised by the eavesdropper E.

Let us highlight that even if the eavesdropper uses the same numbers $e = e'$ when forming keys by the users A and B, the keys between A and B will be different since it is highly likely that $x \neq y$. It means the man-in-the-middle attack is carried out $K_E \neq K_E^{'}$ and $K_A \neq K_B$. This fact will be used when building the authentication protocols based on the PUFs.

2.2. The principles of PUF building and features

PUF can be described by pairs of input and corresponding to output parameters (signals): $R = f(C)$, where input signals $C = c_1, c_2, \dots, c_t$ are called challenges, and output signals $R = r_1, r_2, \dots, r_k$ are called answers (responses) [19]. A pair consisting of an input physical parameter (challenge) and an output parameter (response) is called challenge–response challenge-response pair (CRP). PUF must satisfy the following requirements [19]:

a. Response signal R may be extracted repeatedly and reliably by measuring for challenge C.
b. The number of possible challenges C_i must be so large that all responses corresponding to it R_i cannot be obtained by going over within the observable time.
c. Since in the physical system, there is an extremely large number of data determining the response to this challenge, and it must be computationally impossible to calculate, simulate or by any other way to find a CRP (C, R) when knowing the other pair $(C^{'},\ R^{'})$ or some number of such pairs.
d. Cloning of a given physical system by another physical system, which is described by similar multiple CRPs, or its physical reproduction must be extremely difficult.

At present, many PUF types have been suggested: optical PUF, covering PUF, PUF of arbiter type, PUF based on ring oscillators, PUF based on static operative storing device, PUF of butterfly type (latch, multivibrator oscillator), PUF based on failures, combined PUF. Production of all PUFs is characterized by technological variations that affect the output parameters of the system. Due to this, these parameters will vary from device to device while preserving the identity of device functionality and their internal topology. The number of technological variations, such as p-n transitions or impurities in the substrate, determine the number of possible PUFs.

Our research object is an authentication system. For such a system, such properties as robustness, unclonability, and unpredictability are important.

Let us introduce the following notation:
$\{C\}$ - a set of challenges at the PUF input;
$\{R\}$- a set of responses at the PUF output;
$\{C, R\}_s$- a set of CRPs of the sth PUF;
$\{F\}$- a set of PUF for a selected production technology with specified display pairs $R = f(C)$;
$\{F_{ij}\}$- a sub-set of PUF for a specified pair $(C_i,\ R_j)$;
$|A|$ - the potency of an arbitrary set A.

Robustness can be defined as the PUF's ability to maintain its properties, particularly the univocacy of the display $C \rightarrow R$ with changing conditions of PUF functioning (temperatures, humidity, and supply voltage). Additional measures, for example, noiseless codes, are used to increase resistance to destabilizing factors. In this case, they talk about a PUF system [15]. Unclonability. The notion of unclonability in [20] is discussed in two types:

- Existential unclonability, it is understood as an impossibility for the eavesdropper to create two PUFs with the same properties;
- Selective unclonability.

In the second case, creating a new PUF clone of the original PUF is impossible if the eavesdropper accesses the original PUF. At the same time, it is assumed that some restrictions are performed. For example, the time of access to the PUF is limited, and the eavesdropper cannot physically affect the PUF and remain undetected. The eavesdropper can use side channels. In our further study, we will not discuss the features of robustness and unclonability as an assumption that they will be executed.

The unpredictability of PUF can be determined in a narrow and broad sense. Unpredictability in its narrow sense is determined for a separate PUF as follows. With any random equanimous choice of a challenge, the probability of response R_i occurrence is close to probability $\frac{1}{|R|}$. If there is some sub-set of responses $\{\hat{R}\} \subset \{R\}$, which are used in some PUF applications, then the probability of the attacker's success in guessing any response $\{\hat{R}\}$ will amount to $\frac{|\hat{R}|}{|R|}$. If the condition $|\hat{R}| \ll |R|$ is satisfied, then the probability of guessing the response to a random challenge is negligible.

In a broad sense, unpredictability is determined as the impossibility of forming the same responses to different PUFs. As known, the number of display options of type $X^k \rightarrow Y^k$, where the number of boolean k-dimensional functions determines $X, Y \in (0,1)$ and amounts to $(2^{2^k})^k$. It follows that even with moderate k, the probability of occurrence of two identical displays is negligible. However, such idealization of PUF is not confirmed by practice. Maiti *et al.* [21] notes that the number of its states is polynomially dependent on its linear dimensions for any physical system. Therefore, the assessment of the PUF number cannot be achieved in practice. Hence, it is necessary to assume that the PUF number polynomially depends on the potency of a set of responses, $|F| = Poly(|R|)$. It means the possibility of sub-sets $\{F_{ij}\}$, which have the same CRP in some quantity.

In this regard, estimating the potency of such sub-sets and the number of CRPs coinciding with them is necessary. To model the relationship of the CRP of different PUFs, let us apply the class of strictly universal hash functions suggested by Carter and Wegman [22].

Definition. Class of *strictly universal* hash functions is such a set of displays $H: X \rightarrow Y$ that:

a. for any $x \in X, y \in Y$:$\#\{h \in H: y = h(x)\} = \frac{|H|}{|Y|}$, where $|H|$ is the total number of hash functions h, $|Y|$ is the total number of hash codes Y, $\#\{..\}$ is the number of hash functions satisfying the condition given in the curly brackets;
b. for any $x_1,\ x_2 \in X, x_1 \neq x_2$ and $y_1,\ y_2 \in Y$.

$$\#\{h \in H: h(x_1) = y_1, h(x_2) = y_2\} = \frac{|H|}{|Y|^2}.$$

Concerning PUF, let us introduce the notion of a PUF class, under which we will understand a set of PUFs made according to the same technology and having fixed parameters of the challenge and response signals. Then, from condition 1), it follows that for any CRP.

$$|F_{ij}:\ C_i \rightarrow R_j| = \frac{|F|}{|R|} \tag{1}$$

Execution of condition 2) for PUF means that the number of PUFs, for which $(C_i \xrightarrow{F_{ir}} R_r,\ C_j \xrightarrow{F_{js}} R_s),\ C_i \neq C_j$ is determined by the potency of the intersection of sub-sets $F' = F_{ir} \cap F_{js}$ and inversely proportional to the square of the potency of a response set $|R|$.

$$|F'| = \frac{|F|}{|R|^2} \tag{2}$$

In particular, in the case when the PUF response signal R is binary sequence with a length of k symbols, the number of possible answers equals the number of all sorts of binary combinations with the length of k, it means $|R| = 2^k$, from (1) and (2) it follows that:

$$|F_{ij}:\ C_i \rightarrow R_j| = \frac{|F|}{2^k}$$

$$|F'| = \frac{|F|}{(2^k)^2}.$$

From (1) and (2), it is obvious that if $|F'| = 1$, then the number of PUFs is $|F| = |R|^2$, which means it polynomially (by the second degree polynom) depends on $|R|$. The polynomial dependence of the PUF number gives grounds to assume that, on the one hand, the proposed model does not contradict the practice. On the other hand, as shown, it is sufficient to ensure the security of the authentication system using PUF. Thus, we assume that PUFs having the following characteristics are used to solve authentication tasks:

- The number of digits of the binary representation of the response -*k* (PUF dimension) linearly depends on its physical size;
- The number of pairs (CRP) exponentially depends on the PUF dimension $|R| = 2^k$;
- The number of PUFs polynomially depends on the response potency $|F| = Poly(|R|) = |R|^2$.

3. METHOD

3.1. DH authentication protocol using a TA and PUFs

Let us consider the general chart of authentication of a key generated by users A and B if there is a TA under conditions of attacks of an active eavesdropper E Figure 2. The users communicate with TA, where they are preliminary authenticated using protocols that apply certificates, for example, protocols SSL/TLS or IPSec [23]. The users have integrated PUF blocks into their devices. The user's task is to generate the key $K_{AB} = K_A = K_B$ according to the DH method. For this, users have a two-way communication channel between themselves. The key is authenticated via TA based on PUFs. The eavesdropper E has an opportunity to control both communication channels between the users A and B, and channels between the users and TA, and carry out active attacks there.

A database is created, which records sub-sets $\{(\hat{C},\ \widehat{R})_s\}$ of randomly selected CRP for each PUF. The number of such pairs for one device is $\left|(\hat{C},\hat{R})_s\right| \ll 2^k$. The meaning of this restriction is that if the eavesdropper "senses" the device implementing PUF by sending random challenges to it, then the probability of choosing a request from a subset $\{(\hat{C},\ \widehat{R})_s\}$ will be negligible. CRPs $(\hat{C},\ \widehat{R})_s$ for each PUF in some numbers are computed at the plant during the PUF production and recorded in the TA database, which is stored in an encrypted form.

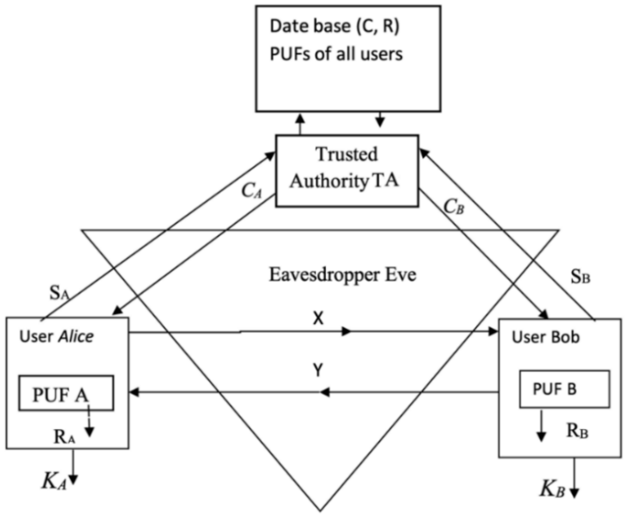


Figure 2. Keys authentication using TA and PUFs

The principle of DH-keys authentication by legal users is in the proof that the keys K_A and K_B generated by the user are the same. Remember that when the eavesdropper carries out a man-in-the-middle attack, he generates two keys: K_E and $K_E^{'}$ with large probability $K_E \neq K_E^{'}$. To confirm that the users generated the same keys, the TA sends challenges to the users, and they send responses S_A and S_B generated using PUF to the TA. If the information contained in the responses confirms that the keys are the same, then the DH key is authenticated as genuine. If not, the key is not authenticated. Therefore, the task of the eavesdropper is to generate and transfer false answers $S_A^{'}$ and $S_B^{'}$ to TA, and they must persuade TA that the keys K_E and $K_E^{'}$ coincide.

3.2. DH-keys authentication protocol using a TA and PUFs (option 1)

Let us consider the key authentication protocol based on the principle. After the users generate DH-key: $K_s = K_A = K_B$, one or both users send a challenge to the TA for execution of the authentication protocol of the key they generated. Key authentication protocol includes the following types Figure 3:

a. The TA sends challenges C_A, C_B from the list of the challenges it has to users A and B.

b. The user A computes the value of his PUF for this challenge $R_A = f(C_A)$. The user B computes a similar value of his PUF $R_B = f(\ C_B)$.

We record the response R_A in the form of concatenation of three parts $R_A = R_{A1}\|R_{A2}\|R_{A3}$, where each part may be presented as a number – Galois field element - $R_{A_i} \in GF(N)$, similarly the response R_B is recorded in the form of $R_B = R_{B1}\|R_{B2}\|R_{B3}$, where $R_{B_i} \in GF(N)$. *i*=1,2,3 (N is a prime number).

c. User A generates a response to the TA in the form of $S_A = [R_{A1} \oplus \mathrm{h}(K_s)] \times R_{A2} modN$, where $h(K_s)$ is the hash function from the generated by the user key $(h(K_s) \in GF(N))$, and sends it to T. The user B similarly generates and sends the response to the TA in the form of $S_B = [R_{B1} \oplus \mathrm{h}(K_s)] \times R_{B2} modN$, where signs “+” and “×” correspond to addiction and multiplication in the field $GF(N)$.

d. Having received S_A andS_B, the TA carries out conversions:

$$S_A \cdot {R'}_{A2}^{-1} modN = R_{A1} \oplus h(K_s),\ S_B \cdot {R'}_{B2}^{-1} modN = R_{B1} \oplus h(K_s),$$

where ${R'}_{A2}^{-1}$, ${R'}_{B2}^{-1}$ are inverse element for R'_{A2}, R'_{B2}according to $modN$, and then computes.

$$R_{A1} \oplus h(K_s) \oplus R_{B1} \oplus h(K_s) = R_{A1} \oplus R_{B1}.$$

The obtained value is compared to $R'_{A1} \oplus R'_{B1}$. Here R'_{A1}, R'_{B1}, R'_{A2}, $and\ R'_{B2}$ are reference responses of A and B devices PUF, which are stored in the database of TA. If:

$$R_{A1} \oplus R_{B1} = R'_{A1} \oplus R'_{B1}, \tag{3}$$

The TA verifies that the keys of A and B coincide, which means there was no man-in-the-middle attack.

e. The TA notifies the users A and B that the keys coincide and authentication is done. For this, he sends messages R'_{A3} and $R'_{\mathrm{B}3}$ to the users A and B, respectively.

f. The user A, having received R'_{A3}, verifies the equality $R'_{A3} = R_{A3}$.

The user B, having received $R'_{\mathrm{B}3}$, verifies the equality $R'_{\mathrm{B}3} = R_{\mathrm{B}3}$. If equalities are true, the users are sure they have generated the same keys. When equality (3) is true, the centre informs the users by inverse values: $\bar{R}'_{A3}$ and $\bar{R}'_{B3}$. After the authentication procedure, the TA deletes the used pair $(C_i,\ R_i)$ from its database.

In this protocol, the most difficult operation of the users is multiplication in the final field by masking multiplier $R_{A2}R_{B2}$. In the TA, it is N modulo addressing of the element. We also assume that used by the users A, B hash function $h(K_s)$ satisfies the cryptographic requirements of collision strength and one-wayness [24]. The PUF of $f_A(c)$ and $f_B(C)$ are computed automatically by integrated devices. In this protocol, unlike the popular Needham-Schroeder authenticated key distribution protocol [25], the TA is used only for authenticating keys generated by the users, and it does not participate in their generation; hence, it cannot access them.

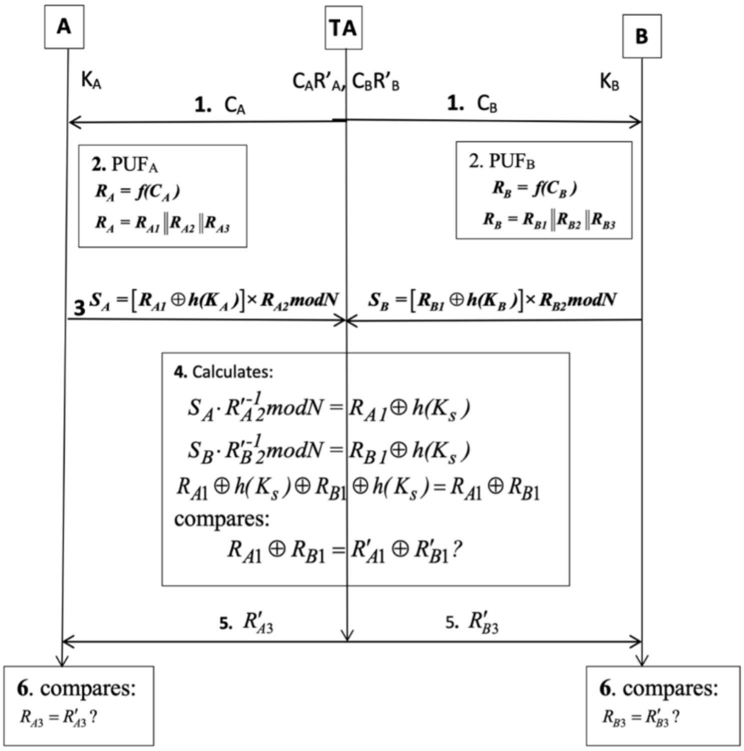

A
TA
B
K_A
$C_AR'_A, C_BR'_B$
K_B
1. C_A
1. C_B
2. PUF$_A$
$R_A = f(C_A)$
$R_A = R_{A1} \| R_{A2} \| R_{A3}$
2. PUF$_B$
$R_B = f(C_B)$
$R_B = R_{B1} \| R_{B2} \| R_{B3}$
3 $S_A = [R_{A1} \oplus h(K_A)] \times R_{A2} modN$
$S_B = [R_{B1} \oplus h(K_B)] \times R_{B2} modN$
4. Calculates:
$S_A \cdot R'^{-1}_{A2} modN = R_{A1} \oplus h(K_s)$
$S_B \cdot R'^{-1}_{B2} modN = R_{B1} \oplus h(K_s)$
$R_{A1} \oplus h(K_s) \oplus R_{B1} \oplus h(K_s) = R_{A1} \oplus R_{B1}$
compares:
$R_{A1} \oplus R_{B1} = R'_{A1} \oplus R'_{B1}?$
5. R'_{A3}
5. R'_{B3}
6. compares:
$R_{A3} = R'_{A3}?$
6. compares:
$R_{B3} = R'_{B3}?$

Figure 3. Authentication protocol chart with the TA based on PUF (option 1)

3.3. Key authentication protocol using the TA and PUFs without the use of arithmetic operations (option 2)

This protocol option does not require any operation of multiplication of large numbers by users and the operation of searching for the inverse element in the module in the TA. Let us consider this protocol, starting with the DH key authentication procedure.

a. The TA sends challenges C_A and C_B to users *A* and *B*.

b. 2. The user A finds response PUF$_A$ $R_A = f(C_A)$, the user B computes similarly and finds response PUF$_B$ $R_B = f(\,C_B)$. Let us present responses R_A and R_B in the form of concatenation of two parts $R_A = R_{A1} \| R_{A2}$, $R_B = R_{B1} \| R_{B2}$ respectively.

c. The users A and B generate responses:

$$S_A = h(K_A) \| h(R_{A1} \parallel h(K_A)),\ S_B = h(K_B) \| h(R_{B1} \parallel h(K_B)),$$

where $h(.)$is hash function.

d. Having received S_A and S_B, the TA verifies the equality of the first parts of the responses $h(K_A) = h(K_B)$. If the equality is true, then the hash function value is computed $(R'_{A1} \parallel h(K_A))$ (parameter R'_{A1} is taken by the TA from its database), which is compared to the value $h(R_{A1} \parallel h(K_A))$ received from the user *A* in the second part of the response. Similarly, having received S_B, the TA finds $h(R'_{\mathrm{B}1} \parallel h(K_B))$ and compares it to $h(R_{\mathrm{B}1} \parallel h(K_B))$ received from the user *B*. If comparisons are true, then the users *A* and *B* have generated similar keys; hence, the authentication is successful.

e. The TA notifies the users A and B that the keys are authenticated. For this, it sends messages R'_{A2} and $R'_{\mathrm{B}2}$ to the users *A* and *B*, respectively. If comparisons are not fulfilled, the TA may notify the users by sending inverse values $\bar{R}'_{A2}$ and $\bar{R}'_{B2}$.

f. The user A, having received R'_{A2}, verifies the equality $R'_{A2} = R_{A2}$. The user B, having received $R'_{\mathrm{B}2}$, performs a similar comparison $R'_{\mathrm{B}2} = R_{\mathrm{B}2}$. If equalities are true, the users are sure they have generated the same keys.

The protocol work chart is presented in Figure 4. Let us highlight that in this protocol option, the length of the response $|S_A|$ $(|S_B|)$ of the users may be decreased if $|h(\,K_s)| < |K_s|$and $\left|h(\,R_{A(B)})\right| < \left|R_{A(B)}\right|$ are chosen.

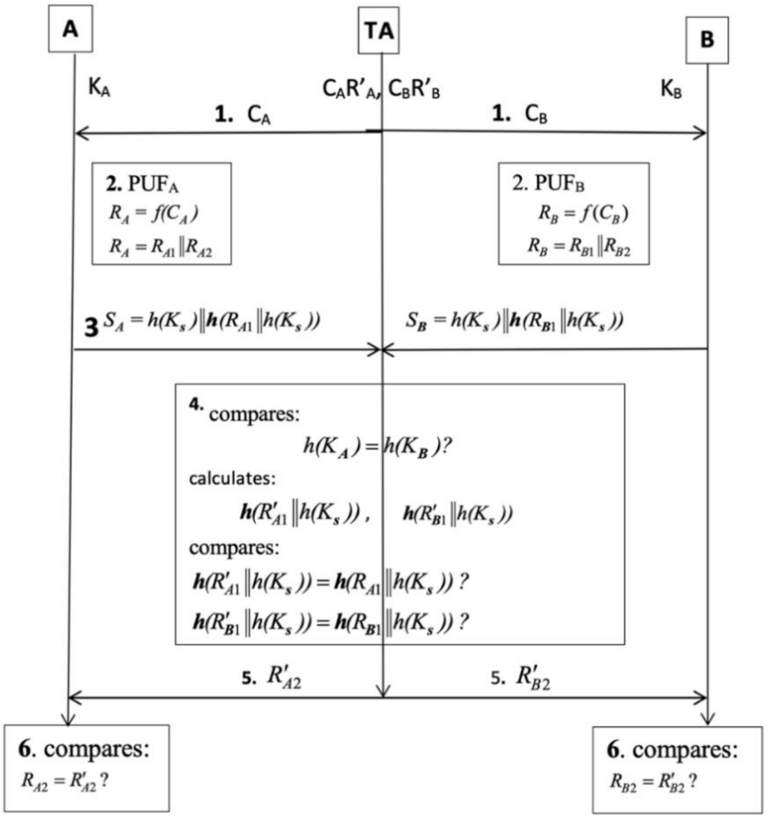

A
TA
B
K_A
$C_A R'_A, C_B R'_B$
K_B
1. C_A
1. C_B
2. PUF$_A$
$R_A = f(C_A)$
$R_A = R_{A1} \| R_{A2}$
2. PUF$_B$
$R_B = f(C_B)$
$R_B = R_{B1} \| R_{B2}$
3 $S_A = h(K_s) \| \boldsymbol{h}(R_{A1} \| h(K_s))$
$S_B = h(K_s) \| \boldsymbol{h}(R_{B1} \| h(K_s))$
4. compares:
$h(K_A) = h(K_B)$?
calculates:
$\boldsymbol{h}(R'_{A1} \| h(K_s))$,
$\boldsymbol{h}(R'_{B1} \| h(K_s))$
compares:
$\boldsymbol{h}(R'_{A1} \| h(K_s)) = \boldsymbol{h}(R_{A1} \| h(K_s))$?
$\boldsymbol{h}(R'_{B1} \| h(K_s)) = \boldsymbol{h}(R_{B1} \| h(K_s))$?
5. R'_{A2}
5. R'_{B2}
6. compares:
$R_{A2} = R'_{A2}$?
6. compares:
$R_{B2} = R'_{B2}$?

Figure 4. Chart of the authentication protocol with the TA based on PUF (option 2)

4. RESULTS AND DISCUSSION

Let us analyze the second option of the DH-key authentication protocol without arithmetic calculations. To do this, we will present and prove several lemmas.

Lemma 1. *Eavesdropping in the information exchange channels (IEC) between users and in IEC between abonents and the TA is less informative for the eavesdropper.*

Proof: The eavesdropper does not obtain any information by inspecting the communication channel between the users since the users do not exchange any data except DH values. An eavesdropper monitors the exchange of messages in channels between users and a TA. It intercepts calls $C_\mathrm{A}, C_\mathrm{B}$ from TA. Then he answers calls S_A and S_B, sending them the second parts of responses to calls R_{A2}, R_{B2} or their inversion.

Responses to the challenges $S_A = h(K_s)\|h(R_{A1} \parallel h(K_s))$ and $S_B = h(K_s)\|h(R_{B1} \parallel h(K_s))$ contain hash codes of the key and hash codes of the first part of the response. If the hash function is chosen correctly, for example, according to SHA-3 [24], restoration of the key K_s by its hash code $h(K_s)$ is computationally impossible. Based on monitoring the challenges and information in the responses, the eavesdropper can set a task of building a challenge-response table $\{C_i \rightarrow R_j\}$ for PUF of the User to carry out an active attack later. However, opportunities for such attacks are limited. In fact, challenges transferred to the eavesdropper and known and are random numbers. In response to the challenge, the eavesdropper can access the first part of the response of PUF in the form of hash code -$h(R_{A1})$, and if the hash function is built correctly, the pre-image cannot be restored. The second part of the response of PUF R_{A2} becomes known to the eavesdropper after the authentication procedure completion. Since the challenge and response are one-time and after their use, they are deleted from the database of the TA, this information becomes useless. And this proves the lemma.

Lemma 2. *The suggested protocol reliably detects the MITM attack.*

Proof: Let's assume that when implementing a protocol for distributing common key between users A and B, the eavesdropper managed to carry out a MITM attack, as a result of which he generated the key $K_E = K_A$ with the user A, and key $K_E' = K_B$ with the user B, while $K_E \neq K_\mathrm{E}'$. The further task of the eavesdropper is to convince the TA that the keys K_E and $K_E^{'}$ coincide.

Suppose also that the eavesdropper intercepted the messages $S_A = h(K_E)\|h(R_{A1} \parallel h(K_E))$ and $S_B = h(K_E')\|h(R_{B1} \parallel h(K_E'))$, which the users sent to the TA. To prove that the keys coincide, the eavesdropper may broadcast the message $S_A = h(K_E)\|h(R_{A1} \parallel h(K_E))$ to the TA, and instead of the message S_B, he must generate the message $S_B' = h(K_E)\|h(R_{B1} \parallel h(K_E))$. The first parts of the messages S_A and S_B' coincide, so the first verification in the TA is passed successfully. For the successful verification of the second part, it is necessary that the equality $h(R_{B1}' \parallel h(K_E)) = h(R_{B1} \parallel h(K_E))$ is true. Sequence R_{B1} is not known to be the eavesdropper. Considering that R_{B1} is a random sequence with a length of *k/2*, the only option for the eavesdropper is to guess such a sequence. Choosing *k* to be large enough, the likelihood of such an attack will be negligible.

Another attack of the eavesdropper may be a transfer of false messages about the completion of authentication to both users, $\tilde{R}_{A2}$ and $\tilde{R}_{B2}$ (although authentication was not completed). Sequences $\tilde{R}_{A2}$ and $\tilde{R}_{B2}$ are binary random sequences, each having a length of *k/2* bit. The probability of their guessing is also negligible. The properties of the PUF can cause another authentication protocol vulnerability to be used. Attack of both users, which creates a false message $S_B' = h(K_E)\|h(R_{B1} \parallel h(K_\mathrm{E}))$ may be successful if it occurs that responses to the challenges C_A and C_B for $\mathrm{PUF_A}$ and $\mathrm{PUF_B}$ coincide, it means $R_B = R_A$.

Let us estimate the probability of this event. Suppose that $|F_i|$ is the number of PUFs, for which $\mathrm{C}_A \rightarrow R_A$. According to feature 1 of the universal hash functions $|F_i| = \frac{|F|}{|R|}$. The eavesdropper's attack will be successful if for $\mathrm{PUF_B}$ presentation $\mathrm{C}_B \rightarrow R_A$ is true. According to feature 2) of the universal hash functions, the number of hash functions, for which $(C_A \longrightarrow R_A,\ C_B \longrightarrow R_A)$, equals $\left|F^{'}\right| = \frac{|F|}{|R|^2}$.

Then, the probability that $\mathrm{PUF_B}$ generates the same response as $\mathrm{PUF_A}$ equals:

$$P_d = \frac{\left|F_i{}^{'}\right|}{|F_i|} = \frac{1}{|R|}.$$

Considering that the PUF response is a binary sequence of length *k*, then $|R| = 2^k$. By choosing a large enough *k*, we can get a negligible probability of a successful attack. The lemma is proved.

Lemma 3. *By sending random challenges to the device and thus "probing" it, an active listener can select a challenge with negligible probability from a subset of selected CRPs stored at the DB of TA.*

Proof: Suppose the eavesdropper can "probe" the User's device, sending random challenges to it, to select a CRPs, which is included in the sub-set of CRP from DB. Then, having found such a pair, the eavesdropper may act as the TA and send a challenge to the users, receive the correct response and confirm authentication. If the potency of the sub-set stored in the DB challenge-request pairs is $\left|\hat{F}_i\right| << 2^k$ for one device, then the probability of choosing the desired request from the pairs sub-set is negligible. In practice, the probability of probing can be reduced by introducing restrictions on the generation of responses by the User after receiving a certain number of challenges, as it is done in password systems. The lemma is proved. Combining the proven lemmas, we formulate a theorem.

Theorem. *Key authentication protocol using TA and PUFs is secured.*

Proof: Based on Lemma 1, it can be stated that any communication between the protocol participants is secured, and the eavesdropper does not receive information on the key. Lemma 2 allows making sure that if the eavesdropper broadcasts the intercepted response of the user A, and the response of the user B will be generated by him using a random selection of a response of the PUF, he will not be able to perpetuate the legal user's identity. According to Lemma 3, the eavesdropper cannot choose a request from a subset of CRPs by sending random challenges to the user. As a result, it can be concluded that legal users can safely authenticate their keys using the proposed protocol.

Comment. Similarly, the security theorem for the first option of the protocol may be proved. Lemmas 1 and 3 may be used without changes and additions. In lemma 2, it is necessary to show that the success of the key substitution attack will not be achieved if the eavesdropper broadcasts the response of the user A $S_A = [R_{A1} \oplus h(K_E)] \times R_{A2} mod N$, and generates the response from the user B in the form of $S'_B = [R_{B1} \oplus h(K_E)] \times R_{B2} mod N$. It is possible that the eavesdropper guesses a part of the response from PUF$_B$ - R_{B2}. Considering that $R_{B_i} \in GF(N)$ and choosing large enough N, the probability of such an attack will be negligible. On the other hand, the attack can be successful if the responses for PUF$_A$ and PUF$_B$ coincide. As shown in the proof of lemma 2, the probability of this event equals $P_d = \frac{1}{|R|} = \frac{1}{2^k}$.

Let us consider an example of selecting PUF parameters. Suppose the DB of the TA for each User contains 100,000 CRPs ($|\hat{F}_i| = 100000$), then if the length k of the bit sequence in the challenge and response is equal to 128 bits, the amount of DB memory for one device will be $128 \cdot 2 \cdot 100000$ bit = 32 Mbit. Then, the capacity of the DB, which stores information about 1 thousand devices, will be 32 Gbyte. The probability of randomly selecting a pair of numbers stored in the DB during PUF probing is $P_{prob} = 10^5/2^{128} \approx 10^{-33}$. With polynomial dependence of PUF number from the number of responses (for second-degree polynom), it is possible to implement $(2^{128})^2 \approx 10^{77}$ of PUF. We see that the proportion of the used CRPs is negligible from their total number. The share of the used PUFs is also negligible compared to their total number, even when the polynomial approximates the PUF number. The proposed authentication protocol can be implemented.

5. CONCLUSION

The paper solves the task of authenticating keys distributed by the DH method among network users, each with a built-in block with a PUF in his device. The keys are authenticated by a TA with a database of challenge-response value pairs for each user's PUF. We briefly describe PUF features and emphasizes the need to formalize PUF features. It is proposed to use th e class of strictly universal hash functions developed by Wegman and Carter. A polynomial dependence of the possible number of PUFs on the number of answers has been proven. Requirements for PUFs suitable for authentication systems are formulated.

We proposed two options for DH key authentication protocols based on the submission of challenges to users by a trusted centre and the generation of responses by them to these challenges using the PUF. The trusted centre makes the authentication decision based on the coincidence (equality) criterion of the keys received from a pair of users. The security of this protocol is proved here. The article also contains an example of evaluating the capacity of the TA database for storing CRPs of users' PUFs, demonstrating the possibility of practical implementation of the method. We plan further research in continuing the work in the following directions: verification of the formal model of the PUF structure and its description in the framework of the extended class of ε-almost universal hash functions; optimization of the DH key authentication protocol parameters; development of PUF-based authentication protocols without a challenge-response database for each PUF in the TA.

ACKNOWLEDGEMENTS

This research is funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (PTF, No. BR18574045).

REFERENCES

[1] L. Mucchi *et al*., "Physical-layer security in 6G networks," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1901-1914, 2021, doi: 10.1109/OJCOMS.2021.3103735.

[2] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Network*, vol. 34, no. 3, pp. 134-142, 2019, doi: 10.1109/MNET.001.1900287.

[3] V. L. Nguyen *et al*., “Security and privacy for 6G: A survey on prospective technologies and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2384-2428, 2021, doi: 10.1109/COMST.2021.3108618.

[4] S. A. Abbas and N. A. Al-Shareefi, “Secure quantum key distribution system by applying decoy states protocol,” *TELKOMNIKA* (*Telecommunication Computing Electronics and Control*), vol. 20, no. 4, pp. 707-714, 2022, doi: 10.12928/ telkomnika.v20i4.22796.

[5] M. Clark and K. Psounis, “Optimizing primary user privacy in spectrum sharing systems,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 533-546, 2020, doi: 10.1109/TNET.2020.2967776.

[6] M. Azouaoui, Y. Kuzovkova, T. Schneider, and C. van Vredendaal, “Post-Quantum Authenticated Encryption against Chosen-Ciphertext Side-Channel Attacks,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 4, pp. 372-396, 2022, doi: 10.46586/tches.v2022.i4.372-396.

[7] M. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976, doi: 10.1109/TIT.1976.1055638.

[8] T. S. Algaradi and B. Rama, “An authenticated key management scheme for securing big data environment,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 3238-3248, 2022, doi: 10.11591/ijece.v12i3.pp3238-3248.

[9] S. R. Boualam, M. Ouaissa, M. Ouaissa, and A. Ezzouhairi, “Secure and efficient routing protocol for low-power and lossy networks for IoT networks,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 1, pp. 478-487, 2022, doi: 10.11591/ijeecs.v27.i1.pp478-487.

[10] B. Yergaliyeva, Y. Seitkulov, D. Satybaldina, and R. Ospanov, “On some methods of storing data in the cloud for a given time,” *TELKOMNIKA (Telecommunication Computing Electronics and Control*), vol. 20, no. 2, pp. 366-37, 2022, doi: 10.12928/telkomnika.v20i2.21887.

[11] Z. Cai, Q. Zhang, M. Li, Y. Gan, and J. Zhang, “Multi-Domain Authentication Protocol Based on Dual-Signature,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 13. no. 1, pp. 290-298, 2015, doi: 10.12928/telkomnika.v13i1.1164.

[12] H. Krawczyk, “SIGMA: The SIGn and Mac Approach to Authentication Diffie Hellman and Its Use in the IKE Protocols,” in *Annual international cryptology conference*, Berlin, 2003, pp. 400-423, doi: 10.1007/978-3-540-45146-4_24.

[13] R. Jin *et al*., “MagPairing: Pairing Smartphones in Close Proximity Using Magnetometer,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 11, pp. 1306–1320, 2015, doi: 10.1109/TIFS.2015.2505626.

[14] V. Yakovlev, V. Korzhik, and S. Adadurov, “Authentication of Diffie-Hellman Protocol for Mobile Units Executing a Secure Device Pairing Procedure in Advance,” in *Proc. 29th Conference of Open Innovations Association (FRUCT*), 2021, pp. 385-392, doi: 10.23919/FRUCT52173.2021.9435495.

[15] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, “Physical unclonable functions and applications: A tutorial,” in *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126-1141, 2014, doi: 10.1109/JPROC.2014.2320516.

[16] S. Shaik, A. K. Kurra, and A. Surendar, “High secure buffer based physical unclonable functions (PUF’s) for device authentication,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no.1, pp. 377-383, 2019, doi: 10.12928/ telkomnika.v17i1.10436.

[17] Y. Yang *et al*., “Man-in-the-middle attack detection and localization based on cross-layer location consistency,” *IEEE Access*, vol. 8, pp. 103860-103874, 2020, doi: 10.1109/ACCESS.2020.2999455.

[18] S. Mitra, S. Das, and M. Kule, “Prevention of the Man-in-the-Middle Attack on Diffie–Hellman Key Exchange Algorithm: A Review,” in: *Proceedings of International Conference on Frontiers in Computing and Systems*, India, 2020, pp. 625-635, doi: 10.1007/978-981-15-7834-2_58.

[19] F. Armknecht, R. Maes, A. R. Sadeghi, F. X. Standaert, and C. Wachsmann, “A formalization of the security features of physical functions,” in *2011 IEEE symposium on security and privacy*, 2011, pp. 397-412, doi: 10.1109/SP.2011.10.

[20] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proceedings of the 44th annual design automation conference*, 2007, pp. 9-14, doi: 10.1109/DAC.2007.375043.

[21] A. Maiti, I. Kim, and P. Schaumont, “A robust physical unclonable function with enhanced challenge-response set,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 333-345, 2011, doi: 10.1109/TIFS.2011.2165540.

[22] J. l. Carter and M. N. Wegman, “Universal classes of hash functions,” *Journal of Computer and System Sciences*, vol. 18, no. 2, pp. 143−154, 1979, doi:10.1016/0022-0000(79)90044-8.

[23] A. Zivi, G. Farahani, and K. Manochehri, “The Role of SSL/TLS, IPsec & IKEv2 Protocols in Security of E-learning System’s Communications, A Study & Simulation Approach,” *International Journal of Computer Trends and Technology*, vol. 50, no. 1, pp. 20-33, 2017, doi: 10.14445/22312803/IJCTT-V50P105.

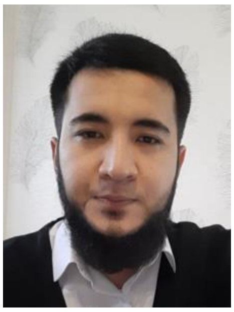
[24] S. Al Busafi and B. Kumar, “Review and Analysis of Cryptography Techniques,” in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2020, pp. 323-327, doi: 10.1109/SMART50582.2020.9336792.

[25] R. M. Needham and M. D. Schroeder, “Using encryption for authentication in large networks of computers,” *Communications of the ACM*, vol. 21, no. 12, pp. 993-999, 1978, doi: 10.1145/359657.359659.

BIOGRAPHIES OF AUTHORS

Victor A. Yakovlev received the Telecommunication Engineer degree from the Higher Military School of Communications, Kyiv, Ukraine, in 1972 and the Ph.D. (Candidate of Technical Sciences) from the Military Communications Academy in 1981. In 1994, he received the Technical Sciences Doctor degree. Since 2004, he has been a Department of Secured Communications Systems Professor at the Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Saint-Petersburg, Russian Federation. He is the author of three textbooks, more than 150 articles, and 13 inventions. His research interests include authentication, cryptography, and key-sharing methods in the networks. He can be contacted at email: viyakov4@gmail.com and viyakovlev4@gmail.com.

Dina Zh. Satybaldina received a B.S. degree in Physics and Computer Science from the Pedagogical Institute, Astana, Republic of Kazakhstan, in 1993 and a Ph.D. degree (Candidate of Physical and Mathematical Sciences) in Solid State Physics from Buketov Karaganda State University, Karaganda, Republic of Kazakhstan, in 2000. Also, she received a Ph.D in Computer Science from Al-Farabi Kazakh National University, Almaty, Republic of Kazakhstan, in 2011. Currently, she is a Head of the Institute of Information Security and Cryptology and a Professor of the Department of Information Security at the L.N. Gumilyov Eurasian National University, Astana, Republic of Kazakhstan. She is the author of three textbooks, more than 100 articles, and 11 inventions. Her research interests include cryptography, error correction coding, malware analysis, and machine learning. She can be contacted at email: dinasaty@gmail.com.



Eldor Egamberdiyev received a B.S. in Computer Engineering and Software Engineering from the L.N. Gumilyov Eurasian National University (ENU), Astana, Republic of Kazakhstan, in 2013 and a master’s degree in Computer Engineering and Software Engineering from ENU, in 2015. He is pursuing a Ph.D in Computer Engineering and Software Engineering at ENU. His research interests include cryptography, cloud computing, secure cloud storage, error correction coding, and applications. He can be contacted at email: eeldoru@gmail.com.

Yerzhan Seitkulov he is a Professor at the Information Security Department ENU. His research interests include cryptography, cloud computing, secure cloud storage, and internet of things. He has over 25 articles in peer-reviewed journals indexed in Scopus. He is also the supervisor of 8 scientific projects funded by the Ministry of Science and Higher Education and the Ministry of Digital Development, Innovation and Aerospace of the Kazakhstan Republic. His professional career consists of activities ranging from the conference chair, a technical program committee member, and a reviewer for several international journals, and conferences. He can be contacted at email: yerzhan.seitkulov@gmail.com.