[1]* Begimbayeva Y.
[2] Ospanov Zh.
[3] Gorlov L.
[4] Ibrayev R.
[5] Ussatova O.

Approaches to Developing Key Distribution Protocols Based on Quantum Key Distribution

Abstract: - Quantum protocols for generating shared secret keys represent an effective method of ensuring secure information transmission using quantum communication channels. However, they are limited in scalability due to the physical constraints of quantum channels, which necessitates the use of classical communication channels for exchanging information between participants. This leads to the necessity of developing security threat models and network protection, especially in the case of hybrid networks, where quantum channels are used alongside classical ones.

The peculiarity of constructing quantum key distribution protocols is associated with the costliness of implementation and implies the inevitable use of classical communication channels for key distribution.

This paper addresses the task of ensuring the security of communication between subscribers not directly connected by quantum channels, through intermediary nodes, and analyzes threats from internal and external adversaries. Special attention is paid to situations where both types of adversaries have information about the processes and protocols in the network.

The paper examines widely used key distribution protocols employed in practice. It describes the main security requirements applied to key distribution protocols, and presents the results of an analysis of the compliance of the discussed protocols with these requirements

Keywords: Cryptology, cryptographic protocols, information security, quantum key distribution.

I. INTRODUCTION

Research into quantum computing is currently being actively pursued on a global scale. The development of computers based on quantum computation models is anticipated to negatively impact various cryptographic systems and key distribution protocols. While these quantum computers offer considerable benefits to information technology, particularly when paired with artificial intelligence, they also pose the risk of being utilized as powerful surveillance tools [1]. This has spurred a competition between quantum computers and the development of quantum-resistant cryptographic techniques.

Quantum key generation protocol operates by utilizing a physical channel that allows the transmission of single photons between two protocol participants. However, the process of key generation itself requires the presence of a classical communication channel [2],[3], through which participants exchange public key information and subsequently conduct further information exchange. Thus, participants of the quantum protocol are connected both by a quantum communication network and a classical one.

Despite the reliability of such quantum communication, the quantum communication channel possesses a number of operational limitations, preventing its application as an alternative to existing secure communication networks. Specifically, connecting a large number of participants pairwise via quantum communication channels would be physically impractical. Therefore, in practice, architectures resembling "stars" or various types of trees are likely to be used.

Hence arises the challenge of ensuring secure communication among subscribers not directly connected by quantum channels, through other subscribers who do have such channels. Moreover, the compromise of intermediate nodes should not lead to the compromise of information transmitted between subscribers connected in the quantum network through that node.

This challenge is pertinent even in cases where there are only two protocol participants, but they are located at significant distances from each other, as in this scenario, due to the limitation on the length of the quantum channel between the endpoints, intermediate nodes would need to be established [4]. Within the framework of the considered model, such nodes can be equated to full-fledged protocol participants.

[1-5] Satbayev University, Almaty, Kazakhstan, Almaty, Kazakhstan
[2, 3] Al-Farabi Kazakh National University, Almaty, Kazakhstan
[1, 5] Energo University, Almaty, Kazakhstan
* Corresponding Author Email: enlikb89@gmail.com

To assess the security of such hybrid communication networks, it is necessary to construct a threat model defining the capabilities of potential information security violators. Let's define such a model for an insider threat and an external adversary. A significant difference between them lies in the fact that an insider threat has access to a quantum communication channel with at least one legitimate user, and in the worst case scenario, if they are the center of a network built according to the "star" topology, they have access to all quantum communication channels of other users. Meanwhile, an external adversary may possess significant resources, both technical and financial, as well as high scientific potential.

Naturally, the most dangerous scenario for the communication network is one in which both external and internal malefactors have exhaustive information about all processes and protocols in the secure communication network, besides the secret keys of other users, and act in concert.

II. METHODOLOGY

2.1. Enemy model

The adversary does not possess the technical capability to compromise the quantum properties-based key distribution protocol. The adversary can use any available combination of standard operations to construct new messages from those they know. It is assumed that the adversary always knows the structure of messages observed in the communication channel. The only unknowns to the adversary may be specific values of individual message fragments. In particular, they can extract parts from intercepted messages, combine them to form new ones, decrypt and encrypt messages using known keys, compute hash values, and so forth. Additionally, keys, random tags, subscriber identifiers, etc., are always considered by the adversary as indivisible fragments, meaning they can only extract them from protocol participants' messages as whole entities.

If encrypted information is present in a certain message, the adversary can only use it in its encrypted form unless they possess the corresponding decryption key. More precisely, it is assumed that all cryptographic algorithms have perfect properties - all one-way functions are indeed one-way, the publicly accessible catalog of public keys is public and protected from substitution, and only those possessing the secret key can decrypt the message.

It is assumed that the adversary has full control over all communication channels. This implies that the adversary can eavesdrop on all messages, delete messages from communication channels, redirect them to other recipients, and generate and send messages to any participants.

The adversary has financial resources comparable to the value of the protected information. For specificity, we make the assumption, as in [5], that they have access to the world's most powerful computational systems, which they can exploit for one year to attack individual cryptographic primitives and protocols as a whole.

The adversary can impose a certain (limited) amount of open data for transmission in a secure (encrypted) form. This allows them to conduct attacks on encryption algorithms with knowledge of pairs of plaintexts and ciphertexts. The number of such imposed messages is limited by the communication system's operating speed and the amount of memory available to the adversary for storing the results of their actions. To estimate the amount of memory available to humanity, it should be noted that the volume of the entire Internet by the end of 2022 was about 90 Zettabytes (approximately 2^{76} bytes), and by 2025, the volume of data created only by connected IoT devices will reach 79.4 Zettabytes [6]. On the other hand, if there is a constant communication channel operating at a speed of 10 Gigabits per second between participants of the secure communication network, then approximately 2^{56} bytes of data can be transmitted over it in a year. Therefore, let's assume that the adversary can utilize a data storage capacity of about 2^{56} bytes.

2.2. Insider Intruder Model

The intruder does not possess the technical capability to compromise the quantum properties-based key distribution protocol.

The intruder can view all messages passing through their portion of the network, including those addressed to other users, and can decrypt those for which they possess the corresponding keys.

The intruder has complete control over their segment of the communication network, including the ability to delete messages from communication channels, redirect them to other recipients, and generate and send messages to any participants.

2.3. Description of attacks

User impersonation [7] is an attempt to substitute one user for another. The intruder, acting on behalf of one party and fully mimicking its actions, receives responses of a specific format necessary to forge individual protocol steps. In protocols involving a third party, attacks based on substituting a trusted server are possible. For instance, one party, having trusted relations with the server, acts on its behalf, substitutes its traffic exchange with other parties, and as a result gains the ability to disclose values generated by the key center.

Message replay [8] involves reusing previously transmitted messages or parts thereof in the current or previous protocol session. For example, retransmitting information from a previously conducted identification protocol could lead to a repeated successful identification of the same or different user. In key exchange protocols, this attack is often used to replay previously used session keys—a freshness-based attack.

Reflection attack [9] is a method targeting the "challenge-response" authentication system, using the same protocol in both directions. Each side uses the same "challenge-response" protocol to authenticate the other side. The essence of the attack is to deceive the target into providing a response to its own challenge.

Message delay [10] involves intercepting a message by the adversary and imposing its delivery at a later time, also a form of message replay attack.

Parallel session attack [11] entails the adversary intentionally opening multiple parallel sessions simultaneously to use messages from one session in another.

Attack using chosen plaintexts [12] targets "request-response" type protocols, where the adversary selectively chooses requests to obtain information about the long-term key of the prover. This attack may include specially crafted plaintexts if the prover must sign or encrypt a request, or encrypted texts if the prover must decrypt a request. The adversary exploiting their means as part of the telecommunication structure ("man-in-the-middle") [13].

Known session key attack aims to obtain information about the long-term key or any other key information allowing the reconstruction of session keys for other protocol sessions.

Unknown common key attack [14] results in subject A believing they share a common key with B, while in reality, B mistakenly assumes the key is used with subject $E \neq A$.

Binding-based attack [15] involves substituting one participant's public key with another public key possessing a secret part known to the adversary. This enables the adversary to learn the contents of encrypted messages sent to that participant.

2.4. Protocol Security Properties

Subject authentication [16] ensures one party's confidence by providing evidence and/or credentials of the identity of the second party participating in the protocol, verifying that the second party indeed participated during the current protocol run. Message authentication [17], [18] requires the protocol to provide means to ensure that the received message or data segment was created by a specific party at some (usually unspecified) point in the past and that this data has not been tampered with or forged. Replay protection [19] guarantees to one party that an authenticated message is not stale. Depending on the context, this may mean: that the message was created during this session; the message was created within a known recent time window; the message has not been received before. Implicit (hidden) recipient authentication [20] ensures that the sent message is readable only by entities authorized by the sender. Thus, only legitimate authorized participants will have access to the current information, multicast message, or group communication. Source authentication [21] allows legitimate group members to authenticate the source and content of information or group communication, even in cases where group members do not trust each other. Authorization (by a trusted third party) [22] enables each protocol participant to verify that another source is authorized by a trusted third party. Key authentication [23] is a property where one party is assured that no other party, except for a specifically designated second party (and possibly additional identified trusted parties), can access a particular secret key. Key confirmation [24] allows one party to ascertain that the second (possibly unidentified) party indeed possesses a specific secret key (or all keying material necessary for its computation). Backward secrecy [25] ensures that compromising long-term keys does not compromise keys from past sessions. Key agreement [26] employs dynamic key management to derive new session keys. Secure negotiation of security parameters ensures that declared capabilities and agreed parameters have not been tampered with by an attacker. Confidentiality [27] is the property that specific data or information (typically sent or received as part of "protected" message content or otherwise derived from exchanged data) remains inaccessible or

undisclosed to unauthorized individuals, organizations, or processes and remains unknown to an attacker. (Limited) Denial of Service (DoS) resilience [28] protocols may be vulnerable to: memory distribution DoS; computational power DoS; third-party spam (a situation where one or more hosts send a victim a large number of packets). Accountability [29] ensures that the actions of a system entity can be uniquely traced back to that entity, which may be held accountable for its actions.

III. RESULTS

3.1 Existing protocols

Currently, there are protocols that satisfy most of the requirements. In the article, two of them will be examined: the Needham-Schroeder protocol [30] and Kerberos [31]. When describing the protocols, the following conventions will be used:

- Alice, Bob - network subscribers;
- Trent - the trusted center;
- A, B - subscriber identifiers;
- K_A - subscriber A's secret key;
- K - session key;
- $\{M\}_K$ - information M encrypted with key K;
- R_B - randomly generated non-repeating data by user B;
- t - timestamp;
- L - key lifetime.

3.2 Needham-Schroeder Protocol

Formal representation of the protocol [30]:

Step Action

$$Alice \rightarrow A, B, R_A \rightarrow Trent$$
$$Trent \rightarrow \left\{R_A, A, B, \{K\}_{K_A}\right\}_{K_A}, \{t, A, B, \{K\}_{K_B}\}_{K_B} \rightarrow Alice$$
$$Alice \rightarrow \{t, A, B, \{K\}_{K_B}\}_{K_B} \rightarrow Bob$$
$$Bob \rightarrow \{R_B\}_K \rightarrow Alice$$
$$Alice \rightarrow \{R_B - 1\}_K \rightarrow Bob$$

Protocol idea: The secret key of each user is pre-known to a trusted third party (usually a server). A user wishing to initiate secure communication with another user sends a corresponding request to the server, containing user identifiers along with some unique (pseudo) random value. The server generates a session key for communication between the subscribers by encrypting it with the keys of both users, so that each can obtain the value of this key and also confirm the source of the message (the authenticity of the second user). The protocol is based on symmetric encryption algorithm. It is protected from replay attacks by introducing a timestamp in some messages.

3.3 Kerberos Protocol

Formal representation of the Kerberos Protocol [31]:

Step Action

$$Alice \rightarrow A, B \rightarrow Trent$$
$$Trent \rightarrow \{t, L, K, B\}_{K_A}, \{t, L, K, A\}_{K_B} \rightarrow Alice$$
$$Alice \rightarrow \{t, A\}_K, \{t, L, K, A\}_{K_B} \rightarrow Bob$$
$$Bob \rightarrow \{t + 1\}_K \rightarrow Alice$$

Protocol Overview: Similar to the Needham-Schroeder protocol, the trusted third party generates a session key with a limited lifetime. Users authenticate each other by comparing values encrypted with the session key. In this case, instead of a random unique value, a timestamp is used.

3.4 Security Properties of Protocols

1. Subject Authentication:

1.1 In the Needham-Schroeder protocol, the center cannot authenticate the subject's authenticity at step 1. However, by sending a response encrypted with the subject's secret key, it prevents the use of this data by a party without the subject's secret key. At step 2, the subject ensures that the message came from the center because the response contains encrypted random data generated by the subject at step 1. Subsequent steps authenticate subjects by confirming possession of secret keys. At the last step, the initiating subject authenticates itself by confirming a random sequence generated by the responding subject on the session key received from the center in encrypted form.
1.2 In the Kerberos protocol, similarly, the trusted center encrypts data with the secret keys of the subjects. Therefore, no one except the subjects can use the data obtained from the center. Authenticity of the subject is verified in the last step using a timestamp change.
2. Message Authentication: The authenticity of data in both protocols is confirmed by encryption with the users' secret keys.
3. Replay Protection:
3.1 In the Needham-Schroeder protocol, the center does not keep track of previously sent messages and responds to any incoming requests from subscribers. However, the subscriber ensures that the response is received for its message by the uniqueness of random data resent at step 1 and returned at step 2. The second subscriber also verifies that the initiating party is authentic at the time of exchange by comparing the values of unique random data at steps 4 and 5.
3.2 The Kerberos protocol provides this property using timestamps, which requires participants to synchronize the current time.
4. Implicit (Hidden) Receiver Authentication: If the subscribers' secret keys have not been compromised before the exchange, then no one except them can use the messages encrypted by the center for the subscribers and authenticate themselves.
5. Source Authentication: In both cases, subscribers transmit information encrypted with the secret key they are in contact with. Since this key is known only to the responding subscriber and the center, the transmission of the correct message confirms that the initiating subscriber can interact correctly with the center, thus the provided identifier is authentic.
6. Authorization (by a Trusted Third Party): Both protocols use a trusted third party to prove the authenticity of subscribers.
7. Key Authentication: In both protocols, the session key is transmitted only when encrypted with the secret key of one of the subscribers and cannot be compromised without compromising the secret key of one of the subscribers.
8. Key Confirmation: Both protocols provide confirmation of the correctness of the session key by encrypting random data or a timestamp.
9. Backward Secrecy: In both protocols, compromising a user's secret key compromises all past, current, and future sessions.
10. Session Key Generation: The purpose of the protocols is to obtain new session keys. Session keys are generated by the center, making it the sole node responsible for changing the session key.
11. Protected Negotiation of Security Parameters: The extension capability introduced in Kerberos version 5 allows subscribers to negotiate the primitives and other session parameters used.
12. Confidentiality: Compromising protected data in both protocols requires compromising the secret key of one of the users. An internal intruder can also disclose the session key to unauthorized parties, but this will compromise the information already known to them.
13. (Limited) Denial-of-Service (DoS) Resilience: Both protocols are not protected from such attacks because any unauthorized participant can initiate a session on behalf of any other subscriber. Protection against attacks can be implemented in the center by limiting the number of sessions opened for each user or other methods.
14. Sender Invariance – The party is confident that the message source remains the same as the one that started the message, although the actual identity of the source is not important to the recipient:
14.1. In the Needham-Schroeder protocol, the initiating subscriber confirms its authenticity by knowing the session key issued by the center.
14.2. In the Kerberos protocol, subscribers exchange messages encrypted with the session key received from the center at the final stage.
15. Proof of Message Source:

15.1. At the third step of the Needham-Schroeder protocol, the message may be a replay of a previously transmitted message. This will be detected due to the expired timestamp. Further messages to the intruder also make no sense to spoof, as the session key value at the time of their transmission is already determined.
15.2. In the Kerberos protocol, the authenticity of the subscriber is guaranteed by sending their identifier in encrypted form (on the session key), and in the final stage, by confirming possession of the session key when encrypting the modified timestamp.
16. Proof of Message Receipt – In both cases, parties confirm the correctness of the generated session key.
The comparison results of the protocols are presented in the table below. In it, the symbol "+" denotes security properties provided by the protocol, "-" denotes those not provided, and "?" denotes properties whose provision depends on the protocol's implementation.

Table I: Ensuring the Required Security Properties of the Nidhugg-Schroeder and Kerberos Protocols.

№	Security Properties	Needham-Schroeder Protocol	Kerberos Protocol
1.	Subject Authentication	+	+
2.	Message Authentication	+	+
3.	Replay Protection	+	+
4.	Implicit Recipient Authentication	+	+
5.	Source Authentication	+	+
6.	Authorization (dedicated trusted third party)	+	+
7.	Key Authentication	+	+
8.	Key Correctness Confirmation	+	+
9.	Backward Secrecy	-	-
10.	New Key Establishment	+	+
11.	Secure Parameter Negotiation Capability	-	+
12.	Confidentiality	+	+
13.	Denial of Service (DoS) Protection	?	?
14.	Sender Invariance	+	+
15.	Message Source Proof	+	+
16.	Message Receipt Proof	+	+

The protocols described above do not fully meet the requirements outlined in this section, specifically in terms of properties 9, 11, and 13 (see Table I). Let's take a closer look at these properties.

3.5 Backward Secrecy

The protection against replay attacks can be implemented by using a mechanism called "rolling code," where encryption keys are changed upon request by either party when there is suspicion of key compromise, after a certain period of time, or under other conditions (in extreme cases, a new key is generated for each communication session). Additionally, the session key can also be regenerated during the session to localize the consequences in case of compromise (in extreme cases, each message between users is encrypted with a separate key).
One of the most promising ways to ensure the secure alteration of the master secret key is through multi-factor authentication mechanisms, significantly complicating the task of an attacker to interfere with this process.

3.6 Secure Parameter Negotiation Capability

The Needham-Schroeder protocol, in its original form, does not provide users with the ability to define a set of cryptographic primitives for future use. However, this can easily be addressed by introducing additional steps in the protocol or by adding necessary data to the messages exchanged in existing steps. For example, at step 2, the initiating user can append information about their cipher suite to the message, and the second user can select cryptographic primitives and return them in the subsequent step.

3.7 Denial of Service (DoS) Protection

The ability to send an unlimited number of messages to one user on behalf of others can be limited by the server's capabilities. Old messages will be discarded by any user due to timestamp inconsistencies. Therefore, an attacker, who loses the ability to generate new requests because the server stops responding, will not be able to cause denial of service to a legitimate network subscriber.

IV. Conclusion

with these requirements. By evaluating the strengths and weaknesses of protocols like Needham-Schroeder and Kerberos, we've gained valuable insights into their applicability and effectiveness in real-world scenarios.
Furthermore, it's worth highlighting that this research contributes to the broader field of network security and cryptography, shedding light on the practical implications of key distribution protocols. By addressing fundamental security requirements and assessing protocol performance, we pave the way for future advancements in secure communication protocols. moving forward, the next phase will involve the development of a quantum key distribution protocol.

V. Acknowledgment

Research work was carried out within the framework of the project AP19675961 “Development and research of keys distribution protocols based on quantum properties”, which is being implemented at the Non-profit joint-stock company "Kazakh National Research Technical University named after K.I. Satbayev".

References

[1] Algazy, K.; Sakan, K.; Khompysh, A.; Dyusenbayev, D. Development of a New Post-Quantum Digital Signature Algorithm: Syrga-1. Computers 2024, 13, 26. https://doi.org/10.3390/computers13010026.

[2] Bennett C. H., Brassard G., «Quantum Cryptography: Public Key Distribution and Coin Tossing», Proceedings of International Conference on Computers, Systems & Signal Processing, p. 175, 1984.

[3] Begimbayeva, Y., Zhaxalykov, T., Ussatova, O. Investigation of Strength of E91 Quantum Key Distribution Protocol (2023) Proceedings - 2023 19th International Asian School-Seminar on Optimization Problems of Complex Systems, OPCS 2023, pp. 10-13. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=10275322 ISBN: 979-835033113-4 doi: 10.1109/OPCS59592. 2023.10275771.

[4] infotecs.ru, «ViPNet QTS Lite». Available: https://infotecs.ru/products/vipnet-qts-lite/.

[5] A. Abdrakhmanov, “Models of cryptographic protection violators and the ST RK 1073-2007 standard,” Proceedings of the National Academy of Sciences of the Republic of Kazakhstan, vol. 6, pp. 62-71, 2017.

[6] D. Reinsel, «How You Contribute to Today’s Growing DataSphere and Its Enterprise Impact,» idc.com, 2019. Available: https://blogs.idc.com/2019/11/04/how-you-contribute-to-todays-growing-datasphere-and-its-enterprise-impact/.

[7] ElShafee, A., El-Shafai, W., «Design and analysis of data link impersonation attack for wired LAN application layer services,» Journal of Ambient Intelligence and Humanized Computing, № 14, p. 13465–13488, 2023.

[8] Menezes A, van Oorschot P, Vanstone S, Handbook of applied cryptography, CRC Press, 1997.

[9] A. S. Tanenbaum, Computer Networks (Updated), Pearson Education Limited, 2013.

[10] Alghamdi, W., Schukat, M., «Precision time protocol attack strategies and their resistance to existing security extensions,» Cybersecurity, т. 4:12, 2021.

[11] Pasca, Vladimir & Jurcut, Anca & Dojen, Reiner & Coffey, Tom, «Determining a parallel session attack on a key distribution protocol using a model checker,» The 6th International Conference on Advances in Mobile Computing and Multimedia, pp. 150-155, 2008.

[12] B. Schneier, Applied Cryptography, Triumph, 2003.

[13] Diffie, Whitfield; Hellman, Martin E., «Exhaustive Cryptanalysis of the NBS Data Encryption Standard,» Computer, т. 6, № 10, pp. 74-84, 1977.

[14] Blake-Wilson, S.; Menezes, A., «Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol,» Lecture Notes in Computer Science, т. 1560, p. 154–170, 1999.

[15] J. Clark, «Attacking Authentication Protocols,» 1996.

[16] I. Cervesato, C. Meadows, and D. Pavlovic, «Deriving Key Distribution Protocols and their Security Properties,» CMU-CS-06-172 School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, 2006.

[17] R. M. Needham and M. D. Schroeder, «Authentication revisited,» ACM Operating Systems Review, т. 21(1), 1987.

[18] B. C. Neuman and T. Ts'o, «Kerberos: An Authentication Service for Computer Networks,» IEEE Communications, т. 32(9), pp. 33-38, 1994.

[19] Sreekanth Malladi, Jim Alves-Foss, Robert B. Heckendorn, «On Preventing Replay Attacks on Security Protocols,» Center for Secure and Dependable Systems Department of Computer Science University of Idaho.

[20] Jianfeng Ma (Xidian University, China) and Xinghua Li (Xidian University, China), «The Provably Secure Formal Methods for Authentication and Key Agreement Protocols,» 2008.

[21] Cyprien Delpech de Saint Guilhem , Marc Fischlin , and Bogdan Warinschi, «Authentication in Key-Exchange: Definitions, Relations and Composition,» 2019.

[22] Suganya Ranganathan , Nagarajan Ramasamy , Senthil Karthick Kumar Arumugam , Balaji Dhanasekaran , Prabhu Ramalingam, Venkateswaran Radhakrishnan, and Ramesh Karpuppiah, «A Three Party Authentication for Key Distributed Protocol Using Classical and Quantum Cryptography,» IJCSI International Journal of Computer Science Issues, т. 7, 2011.

[23] Anne Canteaut Prof. (auth.), Henk C. A. van Tilborg, Sushil Jajodia (eds.), Encyclopedia of Cryptography and Security, Springer US, 2011.

[24] Dojen, Reiner; Lasc, Ioana; Coffey, Tom; Gyorodi, Robert, «Verifying a Key Distribution and Authentication Protocol Using a Logic-Based Proving Engine,» Journal of Computer Science and Control Systems, Oradea, 2008.

[25] Alexander Bienstock, Jaiden Fairoze, Sanjam Garg, Pratyay Mukherjee, Srinivasan Raghuraman, «A More Complete Analysis of the Signal Double Ratchet Algorithm,» International Association for Cryptologic Research 2022 Y. Dodis and T. Shrimpton (Eds.) CRYPTO 2022, p. 782–811, 2022.

[26] Alwen, J., Coretti, S., Dodis, Y., Tselekounis, Y., «Security analysis and improvements for the IETF MLS standard for group messaging,» Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I, p. 248–277, 2020.

[27] NIST Special Publication 800-57, «Part 1 Revision 4,» 2016.

[28] S. Hirose and K. Matsuura, «Key agreement protocols resistant to a denial-of- service attack,» IEICE Trans. Inf. & Syst., т. 4, pp. 477-484, 2001.

[29] Matthew Green , Gabriel Kaptchuk and Gijs Van Laer, «Abuse Resistant Law Enforcement Access Systems,» EUROCRYPT 2021, 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, pp. 553-583, 2021.

[30] G. Lowe, «An attack on the Needham-Schroeder public key authentication protocol,» Information Processing Letters, т. 56(3), pp. 131-136, 1995.

[31] «kerberos.org,». Available: https://kerberos.org/ dist/index.html.