2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) | 978-1-6654-0476-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/ElConRus51938.2021.9396513

Post-quantum Secure Group Messaging

Julia Bobrysheva
Institute of Cyber Intelligence Systems, National Research Nuclear University (Moscow Engineering Physics Institute),
Moscow, Russia
julia@epage.ru

Sergey Zapechnikov
Institute of Cyber Intelligence Systems, National Research Nuclear University (Moscow Engineering Physics Institute)
All-Russian Institute for Scientific and Technical Information of Russian Academy of Sciences (VINITI RAS)
Moscow, Russia
svzapechnikov@mephi.ru

Abstract—**Due to development in quantum computing, we need to create and implement new cryptographic protocols, which are resistant to attacks using a quantum computer, in all practical cases. Last years humans became using messengers to transfer far more important information than earlier, so it is necessary to create new post-quantum secure messaging protocols for peer-to-peer and group communications. In this article, we describe needed security properties, existing ways for the creation of group chats, and our suggestions. We described ways and needed primitives for the creation of a group key establishment scheme based on isogenies of elliptic curves. We describe such protocol as an extended Double Ratchet protocol.**

Keywords—messaging system; group protocol; group communication; end-to-end encryption; post-quantum security; post-compromise security

I. Introduction

Nowadays technology is used everywhere. Online communications such as e-mail, video streaming, and instant messaging become more and more important. For example, if earlier instant messengers were used mainly for personal communication, now they are used for business purposes, the information circulating in them may be limited for transmission to third parties. Thus, it became necessary to make secure instant-messaging tools. Messengers transfer data over the open channel, so it is a widely-known problem to protect such messages. There are many variants of how to protect data transfer via an open channel.

Special attention should be paid to the development of group protocols for instant messaging, as they are the least researched and secured. For example, Signal's Double Ratchet protocol [1] allows to reach post-compromise security in the case of pairwise channels, but it is not used in group communications. We want to modify this protocol for using in groups. In this case, pairwise channels will become a case of n-members group communications when the number of group members is $n = 2$.

The progress of quantum technologies and advances in the creation of quantum computers force to develop protocols that are resistant to attacks of the quantum adversary. There are several ways to construct post-quantum protocols, that is, protocols, which remain secure even to attacks using a quantum computer: code-based cryptography, isogeny-based cryptography, lattice-based cryptography, hash-based cryptography, multivariate cryptography. We opted for isogeny-based protocols due to their compatibility with currently used elliptic curve protocols and small key sizes.

II. Security Properties

At first, we need to define the properties of secure group key exchange protocols in more detail.

Asynchrony suggests that parties can be offline and send or receive messages asynchronously [2]. This property is not a prerequisite to developing a secure messaging protocol, however, current conditions and world's trends suggest that this property is necessary for an effective and user-friendly messaging protocol.

We need to define the necessary security properties for the group gr with the set of members $G_{gr} = \{U_1, \cdots, U_n\}$. The security goals of group exchange can be divided into three groups [3]:

Confidentiality of the conversation:

- End-to-end encryption. The adversary cannot disclose the message m sent by the participant $U \in G_{gr}$ to the target group *gr*. This concept applies only to uncompromised participants in a group exchange.
- Forward secrecy. After leaking session state of the user U, the confidentiality of U's past messages in *gr* is preserved.
- Post-compromise security. If the session state of the user U is leaked after λ multicast messages, where λ is a constant, the confidentiality of future U's messages in *gr* is restored. The property is achieved by continually updating the states of user sessions and invalidating the old states.

The integrity of the conversation:

- Message Authentication. If message m is delivered by user $U \in G_{gr}$ to group *gr*, then it was actually sent by user U.
- No creation. If the message m is delivered to a member $U \in G_{gr}$ by the sender V, then the condition $V \in G_{gr}$ is satisfied. Thus, users who are not members of the group should not have the opportunity to post messages to the group.

978-1-6654-0476-1/21/$31.00 ©2021 IEEE

- No duplication. Each member $V_i \in G_{gr}$ of the group gr invokes the delivery algorithm only once for each action initiated by the user U for the group gr.

- Traceable Delivery. If the execution of the action of the user U is confirmed for the user $V \in G_{gr}$, then the corresponding algorithm was called by all members $V_i \in G_{gr} \backslash \{U\}$. If a group member has received a delivery notification, then the corresponding delivery of his action has been initiated by all members of the group.

- Message ordering. If user U takes action *Act₁* before action *Act₂*, then the group member $V \in G_{gr}$ will not process action *Act₁* after he has processed action *Act₂*.

Unlike all other security and reliability goals, ordering limits instant message delivery: a later message will be delivered only if all previous messages have been delivered. For this reason, instant delivery is possible with the constraint that order among delivered messages is maintained.

Group management:

- Changing the composition of the group. Only members of the group or the group administrator can add or remove users from a group.

- Except for the last point on group management security, all of the defined goals apply to both group messaging and direct messaging. Some of them are usually excluded in the case of two-party messaging because they are trivial to achieve.

Thus, a group instant messaging protocol is considered secure if all of the above properties are done.

III. Threat model

We focus on the following types of adversaries in our work:

- Network attacker. A network attacker can intercept unprotected packets via the communication network, destroy and modify them.

- Malicious server. In this case, we consider attacks on transport layer security between users and a central server, as well as attacks in which an attacker impersonates a central server.

We do not consider a malicious user in this work for the following reasons. If messaging protocols are considered as open systems intended to be used by a wide audience of people, any attacker can become a user of a messaging application and a malicious user. To avoid trivial attacks on messaging protocols, it is assumed that participants of the target group do not break the rules of the protocol, strictly following the algorithm, and are not internal attackers. If it is considered a messaging system intended for use into an internal corporate network, this type of attacker also can be excluded since only company employees have access to this system. In this case, other methods of information security like the intrusion of security policy can be used for excluding the malicious user.

To analyze the strength of protocols in case of compromising user's keys we consider the adversary can break the previously described properties of "forward secrecy" and "post-compromise security". An adversary has such abilities as:

- Compromise of a long-term secret key. An attacker can compromise a user's secret long-term key during or after the protocol execution.

- Compromise of session state. An attacker can compromise a user's session key at some intermediate stage in the protocol execution.

IV. Existing methods of group communication

Currently, messengers use pair channels for group communications, when communication or key sending only are carried out through a special channel for two participants [4]. Participants change keys only when the composition of the group changes.

We can divide currently used methods of group communication on two main strategies [1].

A. Group communication as a set of dialogues between pairs of participants

A member of the group $x_k \in gr$ with a set of participants $G_{gr} = \{x_1, x_2, \cdots, x_n\}$ obtains a pair of keys $\{sk_i, ck_i\}$ for each participant $x_i \in G_{gr} \backslash \{x_k\}$, where sk_i is the shared secret between participants x_k and x_i, obtained in Diffie-Hellman protocol, ck_i is the private key of the participant x_k, k, i - group member identifiers, $k \leq n, i \leq n$. Thus, each group member stores $(n-1)$ key pairs, considering only long-term keys, where n is the quality of group members.

Let the participant A want to send a message m to the group gr with the set of group members $G_{gr} = \{A, B, C\}$. In this case, he sends messages $Enc(m, sk_B)$ and $Enc(m, sk_C)$, where $Enc(m, sk_i)$ is a message m encrypted with the key sk_i, to the server (Fig. 1). The server forwards messages to the appropriate group members.

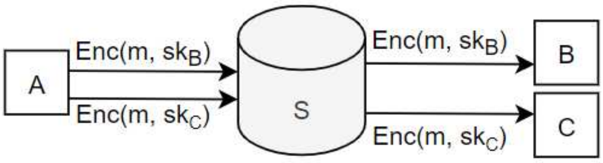

Fig. 1. Schematic representation of traffic in case of organizing group communication as a set of dialogues between pairs of participants. The participant *A* sends the message *m* to group members *B* and *C* in group *gr* with $G_{gr} = \{A, B, C\}$.

This approach is implemented in Signal group messaging [1]. The Signal protocol treats the multicast message in groups as a simple direct message, but it appends the group identifier ID to the ciphertext. In this case, the server processes both group messages and direct messages in the same way, without distinguishing between them. The message is encrypted for each member of the group with the time stamp tm. All ciphertexts are sent to the server with the corresponding recipient ID and timestamp using TLS. The server sends the encrypted messages to the appropriate group members, replacing the recipient ID with the sender ID.

This approach is also implemented in the Threema application [1], which handles group messages as multiple direct messages. Unlike Signal, in Threema protocol participants encrypts and decrypts group and direct messages using the same key. There is no special key for group communication.

B. Group communication as many interactions with the server

A member of the group $x_k \in gr$ with a set of members $G_{gr} = \{x_1, x_2, \cdots, x_n\}$ obtains a key pair $\{sk_k, ck_k\}$ for communication with the server, where sk_k is the shared secret between participants x_k and server S, obtained in Diffie-Hellman protocol, ck_k is the private key of the participant x_k, k - group member identifiers, $k \leq n$. Thus, each group member stores one key pair, and the server stores n key pairs, considering only long-term keys, where n is a quality of group members.

Let the participant A want to send a message m to the group gr with the set of group members $G_{gr} = \{A, B, C\}$. In this case, he sends a message $Enc(m, sk_A)$ to the server, the server sends the messages $Enc(m, sk_B)$, $Enc(m, sk_C)$ to the appropriate recipients (Fig. 2).

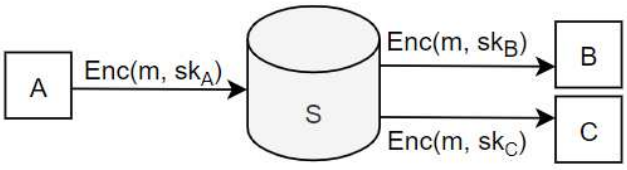

Fig. 2. Schematic representation of traffic in case of organizing group interaction as a set of interactions with the server. The participant A sends the message m to group members B and C in group gr with $G_{gr} = \{A, B, C\}$.

This approach is implemented in the WhatsApp system. Each of the participants generates a key, which he will use for encryption his messages in the group. The key is symmetric and constitutes a chain key. The key is then transmitted once to each other member of the group using the Double Ratchet algorithm for direct messaging. All messages between users and the server are encrypted at the transport level. Only the actual content is encrypted end-to-end. As a result, the sender computes one ciphertext for the entire group.

These methods do not provide all properties needed to create a secure group protocol even against attacks using classical computers. In particular, since the key is updated only when the composition of the group changes, such protocols cannot provide post-compromise security, as, for example, the Double Ratchet protocol in Signal messenger for two participants [1]. Also, participants have to store a large number of keys in the case of using a separate key pair for each member of the group. If a protocol uses a server to send key information, security can be compromised if the server is not trusted.

V. Extended Double Ratchet Protocol

It is necessary to obtain a secure group messaging protocol which provides described above security properties and resistant to quantum computer attacks. A similar protocol for two participants and attacks using a classic computer will be the Double Ratchet protocol of the Signal application [1].

The main idea is that the key changes with every new message. Also, the session key must be dependent on the state in the previous step for post-compromise security property respected. In Double Ratchet, participants generate key pairs and post their public keys in the public domain. Participant A, who wants to send a message, initiates key establish protocol, which in this case is the X3DH protocol [5], to obtain a shared key. Then he uses the output of this protocol as an input to the key derivation function (KDF) for root chain, and then obtains a new root key and a key that is the input to the KDF for sending chain. The output of the KDF for sending chain is a new sending key and a message key. The participant uses the message key to encrypt the message for sending over the open channel. Participant B receives the message and does the same thing, but ends up with a key for receiving chain and a message key. Thus, each of the participant stores three key chains: the root chain, the sending chain, and the receiving chain. The receiving chain of participant A and the sending chain of participant B are identical, and vice versa.

We would like to extend the Double Ratchet protocol for the number of participants $n > 3$. Also, it is necessary to use post-quantum primitives for the resistance of the protocol to attacks using a quantum computer.

The protocol will consist of the following steps (Fig. 3):

1. Each of participants generates a key pair, consisting of a public key Pub_i and a private key Pr_i, and then publish their public key on the server, where i – the index number of the participant;

2. Participant A wants to send a message. He initiates a group key establishment protocol. Then he uses the output of the protocol as an input to the KDF for root chain and obtains a new root key RK and an initial send key CK. The participant then uses the KDF to obtain a new send chain key CK and message key A_1. He encrypts the message with the message key A_1 and transmutes it over an open channel.

3. Other participants receive an encrypted message. They initialize a group key establishment protocol and use the result as an input to the KDF for root chain. In this step, they receive a new root key RK and an initial receive key CK, which they use as an input to the KDF for receiving chain. They then obtain a new receive key CK and the message key A_1. The participants decrypt the received message with the message key A_1.

4. Participant A wants to send the second message. In this case, the group shared key will not be generated again. The

participant initializes the sending key chain and the KDF to get a new send key *CK* and a key for the second message A_2.

5. Participant *B* wants to send a message. He updates its key pair and then computes the shared secret using the group key establishment protocol. Then he uses the result in the KDF for the root chain, starts the KDF for sending chain and receives the message key B_1, sends an encrypted message with this key along with his new public key.

6. Other participants, receiving the message from participant *B*, calculate a new shared secret and repeat step 3 to obtain the message decryption key.

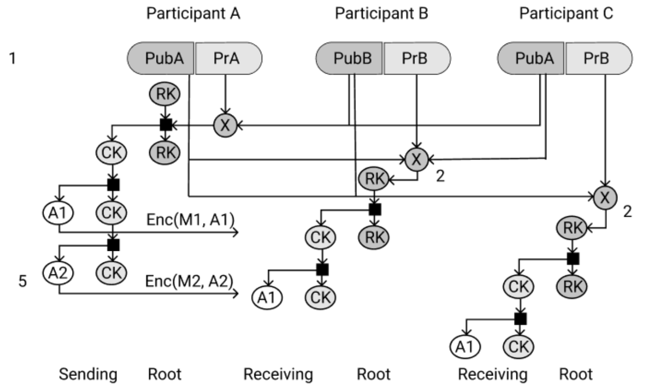

Fig. 3. Extended Double Ratchet protocol for the group with $n = 3$ participants. Stages 1, 2 and 5 are shown.

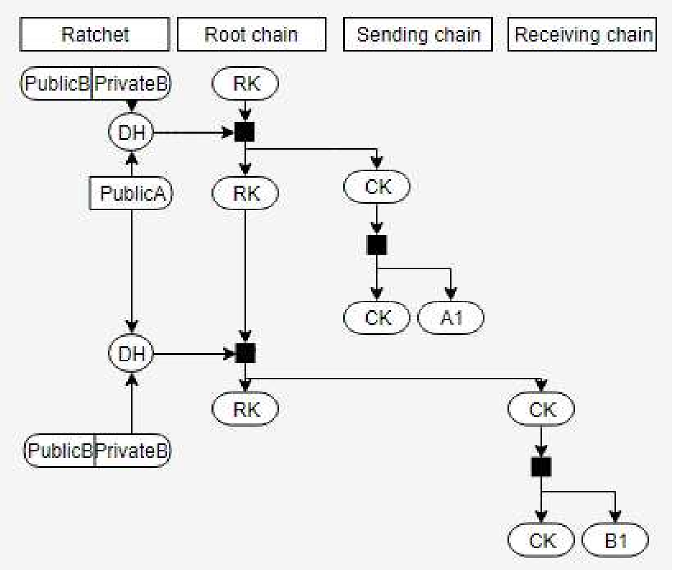

Fig. 4. The Double Ratchet protocol on the side of one participant.

Note that the sending and receiving chains will be reinitialized each time, while the root chain will be the same for all participants and each new key for root chain will depend on the previous steps (Fig. 4).

To build such a protocol, it is necessary to select or create a group key establishment scheme. For example, in [4, 6] group key establishment schemes based on isogenies of elliptic curves are described. However, this scheme does not use authentication mechanisms, therefore, it is not explicitly applicable in our case. It is necessary to create a group authenticated key exchange scheme that is resistant to quantum attacks. This scheme should also provide the ability to update keys asynchronously.

Conclusion

Thus, we established the necessity to develop a quantum-resistant group instant messaging protocol for use in messengers. We have defined the required security properties and a threat model for such a protocol. The currently used communication methods for a group of participants were reviewed and analyzed. The possibility of extension the Double Ratchet protocol to a group of participants was determined and we described how to execute the extended protocol. It was established that to achieve our goal, we need a group post-quantum key establishment scheme. Further work will be to study post-quantum group schemes, AKE on isogeny of elliptic curves and construct the necessary scheme.

Acknowledgement

This work was partly supported by the Ministry of Science and Higher Education of the Russian Federation (state assignment project No. 0723-2020-0036).

References

[1] M. Marlinspike and T. Perrin, “Double Ratchet Algorithm,” *Signal*, 35 pp., 2016, [Online]. Available: https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf. (accessed 24 December 2020)

[2] N. Unger et al., "SoK: Secure Messaging," 2015 IEEE Symposium on Security and Privacy, San Jose, CA, 2015, pp. 232–249, doi: 10.1109/SP.2015.22.

[3] P. Rösler, C. Mainka and J. Schwenk, "More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema," 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, 2018, pp. 415–429, doi: 10.1109/EuroSP.2018.00036.

[4] K. Cohn-Gordon, C. Cremers, L. Garratt, J. Millican, and K. Milner, “On ends-to-ends encryption asynchronous group messaging with strong security guarantees,” Proc. ACM Conf. Comput. Commun. Secur., pp. 1802–1819, 2018, doi: 10.1145/3243734.3243747.

[5] M. Marlinspike and T. Perrin, “The X3DH Key Agreement Protocol,” Signal, p. 11, 2016, [Online]. Available: https://www.whispersystems.org/docs/specifications/x3dh/ (accessed 24 December 2020)

[6] Bobrysheva J., Zapechnikov S. (2021) Post-quantum Group Key Agreement Scheme. In: Samsonovich A.V., Gudwin R.R., Simões A..S. (eds) Brain-Inspired Cognitive Architectures for Artificial Intelligence: BICA*AI 2020. BICA 2020. Advances in Intelligent Systems and Computing, vol 1310. Springer, Cham, doi: 10.1007/978-3-030-65596-9_7