Post-Quantum Security of Messaging Protocols: Analysis of Double Ratcheting Algorithm

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

***Abstract*—Development in the area of quantum technologies led to the appearance of first quantum computers. The threat of using a quantum computer for cryptanalysis requires wide implementing post-quantum security in computing algorithms and communication protocols. We evaluate the computational power of some existing quantum computers to illustrate the relevance of research in post-quantum security. One of the best ways to test post-quantum protocols is to embed them into some non-critical but widely-used sphere. Secure messaging is an excellent example of such an application. In the paper, we analyze the post-quantum security of well-known messaging specification Signal, which is considered to have high-security properties. The core of Signal specification is the Double Ratchet protocol. We notice and explain why it is not a post-quantum secure scheme. After that, we suggest some possible ways to improve the security features of Signal specification.**

Keywords—post-quantum security, messaging protocols, isogeny

I. Introduction

Many science groups from all over the world have been working on a developing of quantum technologies. The most exciting investigation area is the creation of a quantum computer. A quantum computer with several thousand qubits will be able to break all asymmetric cryptographic systems and protocols. The performance of a quantum computer is much higher than the performance of a classic machine. Also, on a quantum computer, it is possible to implement Shor's algorithm, which was created to solve complex mathematical problems that are currently used as the basis of asymmetric cryptosystems. The first quantum computers already exist; you can see most of the recent realizations in Table I.

The difference between quantum computers consists of a large number of parameters like materials and quantity of qubits. Table I presents quantum computers of gate type only. Also, the chart shows certain technologies in quantum computing: superconducting, ion trap, and neutral atoms. However, there are some other technologies like Rydberg atoms and spin for research.

Also, it is essential to understand that the quality of qubits is not in direct relation with computational ability. Described quantum computers work on the different technologies, use different materials, and has different characteristics, so Table I is not intended for comparison of quantum computers, it just shows existing devices.

We need to make some post-quantum schemes against modern widely-used asymmetric ones, considering the development of quantum technologies. NIST concerned about the posed problem and announced a competition for choosing a new asymmetric cryptographic standard resistant for attacks of a quantum computer. In February 2016, NIST published the Post-Quantum Cryptography Report, which emphasizes the need to deploy post-quantum cryptography for information security systems. In August 2018, the first round of the NIST PQC competition was held to select a new, post-quantum cryptographic algorithm for further standardization and application. The second round of PQC competition was held in August of this year, and by the 2021 year, NIST plans to choose a new standard. There are several post-quantum protocols, which take part in this competition. They are divided into groups: code-based, isogeny-based, lattice-based, and hash-based cryptography.

TABLE I. Existing Quantum Computers

Company and name	Technology	Qubits	Issue date
Intel, Tangle Lake [1]	Superconducting	49	8 January 2018
Google, Bristlecone [2]	Superconducting	72	5 March 2018
Google, Sycamore [3]	Superconducting	54	23 October 2019
IBM, Rochester [4]	Superconducting	53	18 September 2019
Rigretti, Aspen-4 [5]	Superconducting	16	14 March 2019
IonQ [6]	Ion Trap	11	19 March 2019
IQOQI, Univ.Ulm, Univ. Innsbruck [7]	Ion Trap	20	11 April 2018
Univ. of Wiscomsin [8]	Neutral Atoms	49	24 June 2016

II. The formulation of the problem

It is necessary to act proactively, so by the time of wide-spreading full-scale quantum computers, quantum-resistant protocols will be used everywhere. It is important to remember that some time for implementation and testing new protocols is required. One of the reasons for the small quantity of post-quantum protocols using is concerns of security specialists

978-1-7281-5761-0/20/$31.00 ©2020 IEEE

about the classical security of such protocols. We don't know what problem we can encounter during the employment of new protocols precisely.

The solving of this problem can be an implementation post-quantum protocol in non-critical and widely-used environments. An excellent example of such an environment is a messaging system. There are two arguments for the proposal:

- Messaging systems are using everywhere, and it is a great test site;
- Messaging systems don't operate with critical information, and in case of error detection, it would not be a severe problem.

Thus, we can use post-quantum protocols in messaging systems, test it, and correct mistakes without negative consequences. Our final goal is to create a post-quantum secure messaging. At first, we need to describe and analyze the applied schemes of messaging systems. We take Signal protocol as an example of messaging systems. The kernel of Signal protocol is Double Ratchet protocol, so we need to recall this scheme.

III. Double Ratchet scheme

The main idea of Double Ratchet is changing keys for each message. It can realize such security properties as:

- Resilience – adversary can't distinguish keys from random;
- Forward security – if an adversary knows key in a particular moment, he can't find the previous keys;
- Break-in recovery – if an adversary knows key in the specific moment, he can't predict future keys.

Thus, the protocol, which provides the listed above security properties, protects user's information. If an adversary intercepts one user's message, he will be able to reveal the content of this message only.

Figure 1 shows a simplified scheme of Double Ratchet protocol [9]. Before starting Double Ratchet protocol participants perform the following actions:

1) Each participant generates a pair of private and public keys;
2) They release their public keys in the open-access directory;
3) They derive a shared secret using the Diffie-Hellman-like protocol.

Double Ratchet protocol consists of the following steps.

1) Alice wants to send a message to Bob. She launches Diffie-Hellman algorithms, using her private key *PrivateA* and Bob's public key, to receive an output.
2) The output Alice uses as an input for the KDF function of the root chain.
3) Alice gets a new key for the root chain and a key for the sending chain.
4) The same actions let Alice receive a new key for sending chain and a message key $A1$.
5) Alice encrypts message $MA1$, using key $A1$. Then she sends encrypted data on an open channel.
6) Bob receives the message and wants to decrypt it. Then he launched the Diffie-Hellman algorithm, using his private key *PrivateB* and Alice's public key *PublicA*. He gets the same output that used Alice.
7) He uses the output as an input date for KDF function and makes a new key for the root chain and a key for a receiving chain.
8) He makes a new key for the receiving chain and a message key $A1$ from the KDF function.
9) Bob decrypts message $MA1$, using the key $A1$.

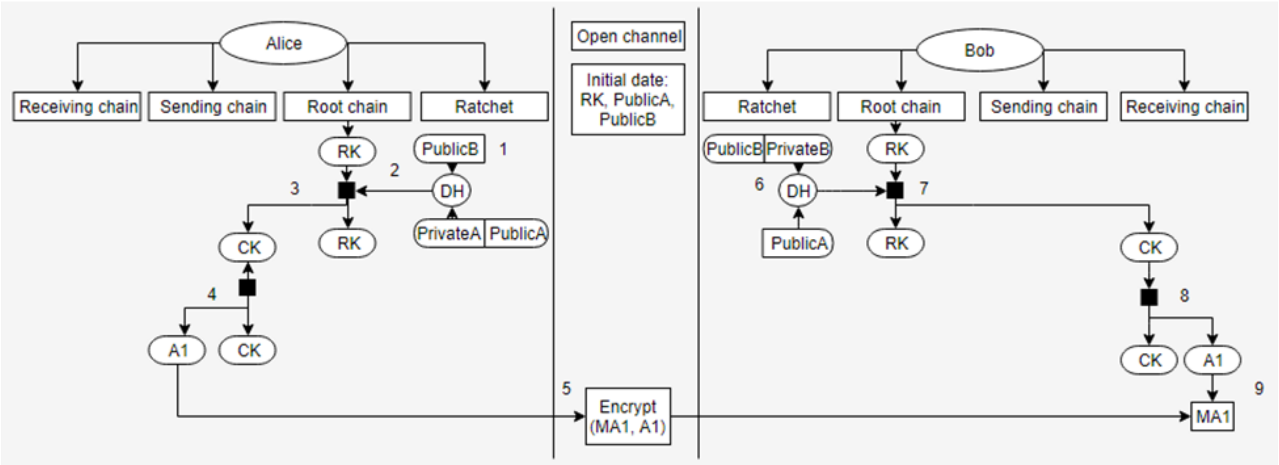

Fig. 1. Double Ratchet scheme (notations: PublicA – Alice's public key; PublicB – Bob's public key; PrivateA – Alice's private key; PrivateB – Bob's private key; DH – Diffie-Hellman algorithm; RK – key for the root chain; CK – key for sending or receiving chain; MA1 – the first Alice's message; A1 – message key).

Participants save three key chains: root chain, receiving chain, and sending chain. Alice's receiving key chain and Bob's sending key chain are identical and vice versa.

When a participant wants to send a message, he generates a new key pair and encrypts the message, using the current public key of another party and his new secret key. He transfers his public key with the massage on the open channel. On the other side, when the participant receives a message, he uses his current private key and a new key to another party. We illustrate it on the example of Alice's side in Figure 2.

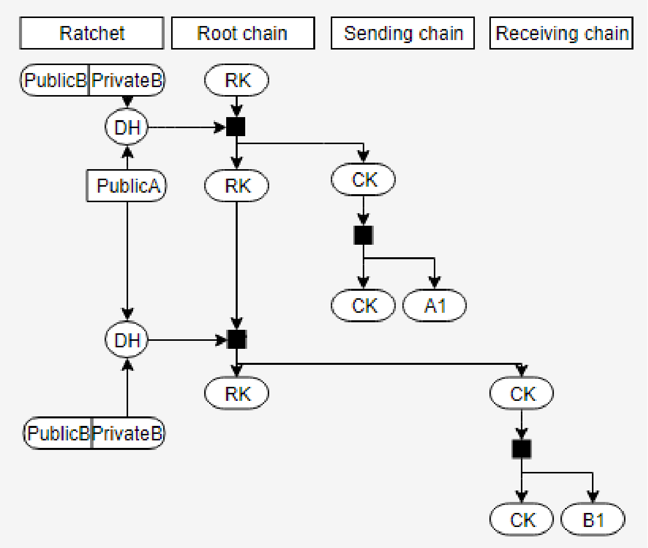

Figure 2. Key changing (notations: PublicA – Alice's public key; PublicB – Bob's public key; PrivateB – Bob's private key; DH – Diffie-Hellman algorithm; RK – key for the root chain; CK – key for sending or receiving chain; A1 – Alice's message key; B1 – Bob's message key).

IV. Analysis of Post-Quantum Security

The kernel of the Double Ratchet algorithm is the Diffie-Hellman algorithm, based on the computational infeasibility of the discrete logarithm problem. The Shor's algorithms can decide such problems in a reasonable time, so the Diffie-Hellman can't provide post-quantum security for the Double Ratchet algorithm. The Double Ratchet algorithm uses the Triple Diffie-Hellman algorithm several times. We describe all such using and explain some possible ways for improving post-quantum security properties.

First of all, we use the Diffie-Hellman algorithm for initialization. If Alice wants to send a message to Bob, she needs a primary key for root key chain RK and a public Bob's key Public B. In the original algorithm Alice and Bob get shared secret RK by launching X3DH algorithm [10]. It is a modification of the Diffie-Hellman. In quantum-resistant Double Ratchet, we can replace it with its post-quantum analogue.

The most suitable approach is the application of isogeny-based protocols because they provide low power consumption for key transfer, which is the most critical requirement for low-level microsystems, and the speed of their operations is sufficient for mobile applications. Table II presents the results of the implementation of one of the latest optimizations.

We can use the SIKE protocol [11] for our goal. SIKE can produce a shared secret for two participants, but we can face some problems during its implementation.

The X3DH protocol generates a bunch of keys for each participant. There are some keys for authentication functions, unlike the SIKE protocol isn't affect such tasks at all. We can use additional authentication mechanisms, but some keys are used in the Double Ratchet too.

TABLE II. The Performance of the Algorithms On Isogenies [12]

Protocol	Platform	Frequency, MHz	Latency, sec	
			Alice	Bob
SIDH p503	32-bit ARMv7 Cortex-A15	2,000	0.042	0.046
	64-bit ARMv8 Cortex-A53	1,512	0.050	0.041
	64-bit ARMv8 Cortex-A72	1,992	0.025	0.021
SIDH p751	32-bit ARMv7 Cortex-A15	2,000	0.135	0.157

For example, the Double Ratchet needs AD parameters for a message's encryption. AD is an abbreviation of "Associated Data". It contains some identity information for both participants. In the X3DH protocol, AD is a concatenation of participants' identification key:

$$AD = Encode\ (IK_A)\ \|\ Encode\ (IK_B).$$

We describe an encryption function to illustrate this.

```
def RatchetEncryptHE(state, plaintext, AD):
    state.CKs, mk = KDF_CK(state.CKs)
    header = HEADER(state.DHRs, state.PN, state.Ns)
    enc_header = HENCRYPT(state.HKs, header)
    state.Ns += 1
    return enc_header, ENCRYPT(mk, plaintext, CONCAT(AD, enc_header))
```

So we need to modify the SIKE or the Double Ratchet protocols for its integration.

Also, the Diffie-Hellman algorithm is used as a cryptographic module in the algorithm itself. Alice and Bob generate a shared secret from their public and private keys. Such implementation of the Diffie-Hellman doesn't require any additional modification so that we can use the SIKE protocol for this goal.

Other components of the Double Ratchet algorithm are quantum-secure. It uses the AES symmetric encryption algorithm for message encryption. A quantum computer can weaken this algorithm due to higher computation possibility, but not so much as the Shor's algorithm will weak asymmetric schemes. We can increase the key sizes of AES to increase quantum cryptographic strength.

Another component in the Double Ratchet scheme is a key derivation function (KDF). This function affords to get two keys from some input data, for example, a new sending chain key and a message key. KDF is usually based on cryptographic hash functions. It is crucial to use quantum secure hash-functions, which is not related to the factorization problem. The Double Ratchet specification recommends implementing HKDF on the HMAC with SHA-256 or SHA-512. This approach allows getting 32-bytes encryption key, 32-bytes authentication key, and 16-bytes IV. We can consider this scheme as a post-quantum secure until there are not any other schemes with proven post-quantum security.

CONCLUSION

To sum up, the Double Ratchet algorithm is an elegant and straightforward protocol for providing classical security in messaging systems, although we can't use it in post-quantum systems. It is required some modifications to accept it as post-quantum secure. One of the possible ways is an implementation of the SIKE algorithms instead of the Diffie-Hellman and the X3DH ones. It can lead to some related work associated with additional keys, especially authentication keys, and changing data types for protocols integration. There are some suggestions about authenticated key exchange protocols based on isogenies [13], which we can try to embed into the Double Ratchet protocol.

The ultimate goal of creating a quantum-resistant Double Ratchet algorithm is developing post-quantum cryptography in total. Using post-quantum cryptography in practical applications can increase their efficiency and improve the security of application both. The quantum-secure protocols must replace all asymmetric protocols, used in modern systems and applications. The messaging systems are only the one area for future work, with which we propose to begin post-quantum protocols implementation.

REFERENCES

[1] Intel Corporation. (2018) 2018 CES: Intel Advances Quantum and Neuromorphic Computing Research. Available from: https://newsroom.intel.com/news/intel-advances-quantum-neuromorphic-computing-research/#gs.g2qmfo (accessed 14 November 2019)

[2] Kelly, J. (2018) A Preview of Bristlecone, Google's New Quantum Processor. Available from: https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html (accessed 14 November 2019)

[3] Martinis, J. Quantum Supremacy Using a Programmable Superconducting Processor. Available from: https://ai.googleblog.com/2019/10/quantum-supremacy-using-programmable.html (accessed 14 November 2019)

[4] Armonk, N. IBM Opens Quantum Computation Center in New York; Brings World's Largest Fleet of Quantum Computing Systems Online, Unveils New 53-Qubit Quantum System for Broad Use. Available from: https://newsroom.ibm.com/2019-09-18-IBM-Opens-Quantum-Computation-Center-in-New-York-Brings-Worlds-Largest-Fleet-of-Quantum-Computing-Systems-Online-Unveils-New-53-Qubit-Quantum-System-for-Broad-Use (accessed 14th November 2019)

[5] Rigretti. Rigetti 16Q Aspen-4. Available from: https://www.rigetti.com/qpu (accessed 14 November 2019)

[6] Wright, K., Beck, K. M., Debnath, S., Amini, J. M., Nam, Y., Grzesiak, N., Kim, J. (2019). Benchmarking an 11-qubit quantum computer. 1–8. Retrieved from http://arxiv.org/abs/1903.08181

[7] Friis, N., Marty, O., Maier, C., Hempel, C., Holzäpfel, M., Jurcevic, P., … Lanyon, B. (2018). Observation of Entangled States of a Fully Controlled 20-Qubit System. Physical Review X, 8(2). https://doi.org/10.1103/PhysRevX.8.021012

[8] Saffman, M. (2016). Quantum computing with atomic qubits and Rydberg interactions: progress and challenges. Journal of Physics B: Atomic, Molecular and Optical Physics, 49(20). https://doi.org/10.1088/0953-4075/49/20/202001

[9] Perrin, T., Marlinspike, M. The Double Ratchet Algorithm. Available from: https://signal.org/docs/specifications/ doubleratchet/#ref-rfc7748 (accessed 14 November 2019)

[10] Perrin, T., Marlinspike, M. The X3DH Key Agreement Protocol. Available from: https://signal.org/ docs/ specifications/x3dh/ (accessed 14th November 2019)

[11] Azarderakhsh, R., Campagna, M., Costello, C., Feo, L. De, Hess, B., Global, I., … Urbanik, D. (2019). Supersingular Isogeny Key Encapsulation April 17, 2019.

[12] H. Seo, A. Jalali, and R. Azarderakhsh, "SIKE Round 2 Speed Record on ARM Cortex-M4," pp. 39–60, 2019.

[13] Longa, P. (2018). A Note on Post-Quantum Authenticated Key Exchange from Supersingular Isogenies. IACR Cryptology EPrint Archive. Retrieved from https://eprint.iacr.org/2018/267.pdf