A Comparative Study of Cryptographic Key Distribution Protocols

Alexandru-Ștefan Gheorghieș[1], Darius-Marian Lăzăroi[1], and Emil Simion[2]
[1] Faculty of Computer Science, Alexandru Ioan Cuza University of Iași, România: {alexandru.gheorghies, darius.lazaroi}@info.uaic.ro
[2] Politehnica University of Bucharest, România: emil.simion@upb.ro

Abstract. *Key distribution protocols deal with generating, exchanging, and storing information (especially shared keys). In this paper, we compare three different types of protocols: classical, quantum key distribution, and blockchain-based protocols, with examples from each category, presenting the particularities and challenges of each one, including solutions and the impact of these protocols.*

Keywords*: key sharing, protocols, public key infrastructure, quantum computing, blockchain, authentication.*

I. Introduction

Nowadays, more and more people use networked systems to communicate with each other. However, there are many disadvantages in using this type of communication due to the fact the online medium has become rather insecure, with attackers trying to obtain more information about the personal data of users while using such communication channels. For this reason, the necessity for security to authorize only the members of the network has increased. As a result, the key distribution protocols appeared. As the name suggests, their aim is to securely share one or more keys through the channel before the communication phase can begin [1].

Key distribution protocols play a very important role in modern day cryptography as they allow users to utilize the more efficient symmetric cryptography (algorithms such as AES), since common asymmetric algorithms (such as RSA) do not offer efficiency for large messages. Because they play a setup role, compromising them can mean compromising every single message that is sent subsequently.

Due to the ease of use and due to the small size of the symmetric keys (max 256-bits for AES), asymmetric cryptography is commonly used in the setup process. Currently, 2048-bit RSA is also the most common algorithm used for signing certificates in the PKI, which helps users in identifying legitimate websites.

The paper is organized as follows: section II briefly explains the public key infrastructure and the rest of the section is structured in three distinct parts, discussing classical key distribution protocols, quantum key distribution protocols and blockchain protocols. In section III, we outline PKI solutions proposed in the context of quantum computing and blockchain. Section IV concludes this paper.

II. State-of-the-art

II.1. Public Key Infrastructure

A PKI (**P**ublic **K**ey **I**nfrastructure) solves the problem of key sharing and authentication. There are many ways to implement a PKI, however, most, if not all of them, use the concepts of **public key certificate**, **certificate authority (CA)**, and **registration authority** as their main components [2].

The certificate authority is a trusted third party used to authenticate the entities taking part in a message exchange by issuing a public key certificate (digital certificate) for all the different entities. This certificate usually contains the public key of the said entity, additional information regarding the owner of the paired private key, a time window indicating for how long the certificate is valid, and the CA's own digital signature. Every user must establish a trust relationship with it because every valid certificate must be signed with the CA's private key. However, the authority must also issue some sort of list that keeps track of what certificates have been revoked due to being compromised or cancelled. The role of the registration authority is to keep track of new users and verify their identity for the CA [3][4]. By using the CA's certificate described above, one can ensure that the user is communicating with the correct party.

II.2. Classical Key Distribution Protocols

In every protocol we will discuss, let the general users be named Alice (A) and Bob (B).

1. Diffie–Hellman protocol is a scheme where different users send each other information over a public channel using a session key [5][6].

In Fig. 1, we present the general scheme of the Diffie-Hellman protocol:

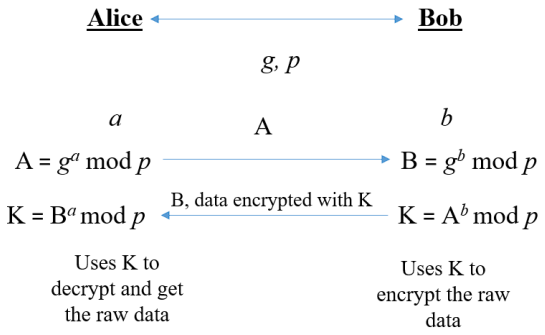

Fig. 1. Diffie-Hellman protocol [7]

The basic protocol looks as follows:

1) Setup phase:

Both agents choose two prime numbers g and p, where p is a big number and g is a primitive root modulo p. Those numbers should be kept secret from other users. Alice and Bob secretly choose big integers a and b as private keys (a for Alice and b for Bob), everyone knowing only his/her private key [5][6][8].

Note: p is at least 512 bits [6], a and $b \leq p - 2$ [9].

2) A → B:

Alice computes $A = g^a \pmod p$, then she sends it to Bob [5][10].

3) B → A:

Bob computes $B = g^b \pmod p$ and he sends it to Alice [5][10].

4) Finally, Alice computes $B^a \pmod p = (g^b)^a \pmod p = g^{ab} \pmod p$ [5][10].

5) Bob also computes $A^b \pmod p = (g^a)^b \pmod p = g^{ab} \pmod p$ [5][10].

As a result, it can be remarked that both results are equal and represent the shared key [5][10].

2. Needham–Schroeder public key protocol is generally used for mutual authentication

The protocol works in the following way:

1) Na and Nb are nonces generated by A and B, respectively, Ka is the public key of A and Kb is the public key of B [11][12].

2) A → B : {A, Na}_Kb

A starts this communication protocol by sending B a message that includes her identity and her nonce, both encrypted with B's public key [11][12].

3) B → A : {Na, Nb}_Ka

B receives the message and using his private key, he decrypts it. Thereafter, he replies the received nonce together with his nonce, encrypting this message with A's public key [11][12].

4) A → B : {Nb}_Kb

When A gets the message from B, she decrypts it and checks whether Na, the nonce sent the first time, is identical to the one received from B. If so, she sends the nonce Nb to B, encrypted with his public key. A similar procedure is made by B, too: he checks if the nonce received coincides with the one he sent. If these two tests succeed, both parties are now authenticated, and the communication channel is safe [11][12].

3. STS (Station-to-Station) protocol is an improved version of the classic Diffie–Hellman. This time, digital signatures are used to provide mutual key and entity authentication [6][13][14][15].

In Fig. 2, you can see how STS protocol works:

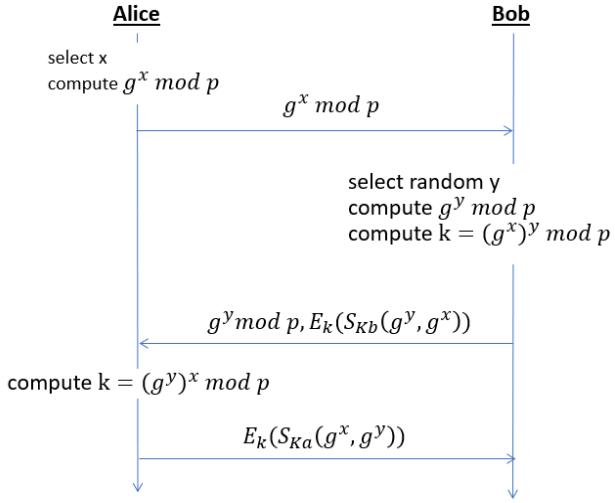

Fig. 2. STS protocol [15]

The notation S_{K_C} means that the message is signed with the private key of agent C, C being either A or B, and E_K means that the signature is encrypted with the key K [6].

4. Kerberos Protocol is a trusted third party protocol whose purpose is to authenticate the client, by issuing him a Ticket Granting Ticket (TGT) [16]

Fig. 3 illustrates the general schema and the steps during authentication of Kerberos protocol. Step 3 is missing because it only describes how TGT is encrypted: using the Ticket Granting Service (TGS) secret key [17].

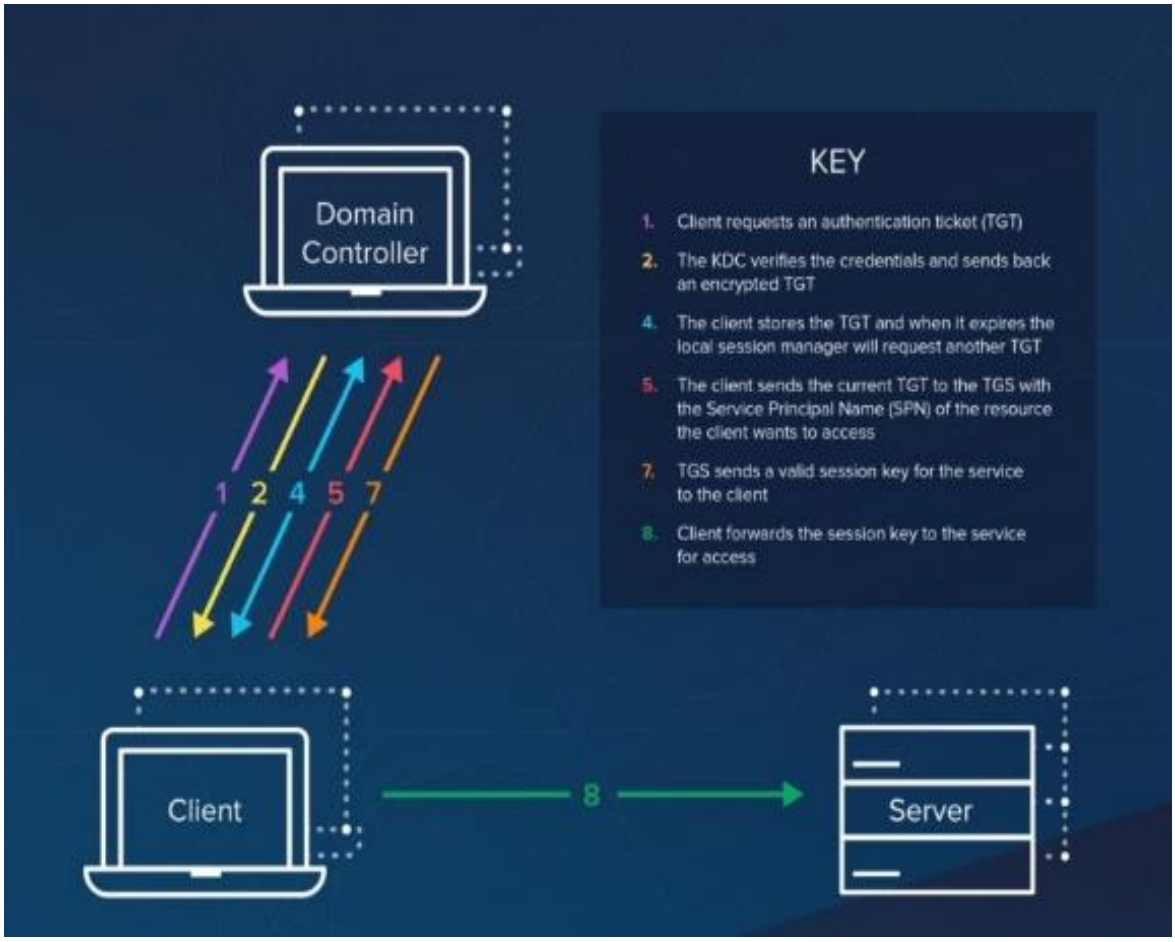

Fig. 3. Kerberos protocol [17]

II.3.1 Quantum Key Distribution

With the looming threat of quantum computers, there have been efforts to increase the security of cryptographic systems by basing them on something else other than the increased computation cost of calculation.

Paul Benioff proposed the first quantum model based on the Turing machine in 1980 [18].

In 1984, because the theory of computation was not well understood, and RSA and DES systems were not totally secure, a different foundation for cryptography was proposed, one that was based on the uncertainty principle of quantum physics. The encrypted information would be obtained using a single photon with 4 polarization directions: 0, 45, 90 and 135. The main appeal of this approach was the fact that an eavesdropper cannot even gain partial information about a transmission without altering the data. This scheme would later be known as BB84. In 2004 a protocol named SARG04 was built based on it, with a different information encoding for use with laser pulses instead of single photon sources [19].

In 1992, Bennett proposed a new Quantum Key Distribution (QKD) protocol based on the same uncertainty principle, but it only uses 2 nonorthogonal states instead of four, which is known as B92. It was demonstrated to be unconditionally secure by Tamaki in 2003. Proving that B92 remains secure even when the attacker can apply all the operations permitted [20].

In 1991, Artur Ekert developed a protocol that was based on the properties of quantum correlated particles. The protocol is based on a famous paradox presented by Einstein, Podolsky

and Rosen in 1935 which is also reflected in its name (the EPR protocol). Here if an eavesdropper is detected, the key is rejected using Bell's inequality [20][21].

II.3.2 Quantum Key Distribution Protocols

1. BB84 protocol

Fig. 4 presents the flow of BB84 protocol.

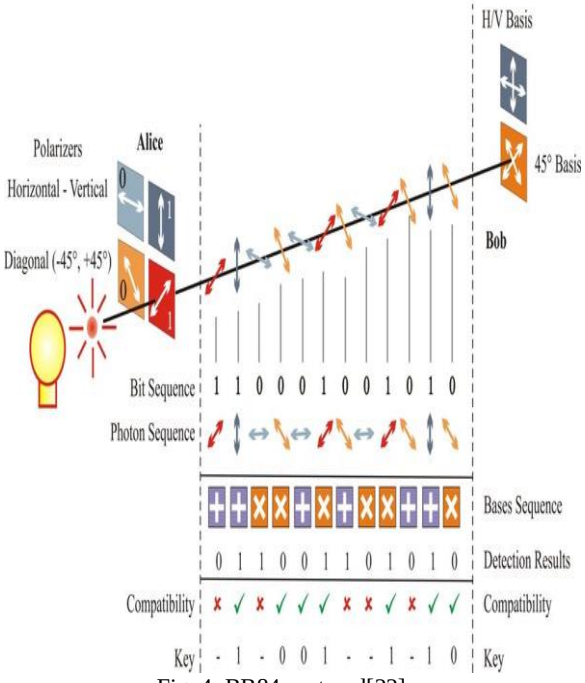

Fig. 4. BB84 protocol[22]

This protocol consists of the following steps:

1) Firstly, Alice generates an *n*-bit sequence, which will consist of the shared key.

2) Setup phase:

Alice and Bob agree on two distinct encodings of a classical bit using a qubit (for instance, 0 is encoded by a photon that is polarized horizontally (→) and at an angle of 45◦ (↗), and 1 is encoded by a photon polarized vertically (↑) and at an angle of 135◦ (↖)).

3) Afterwards, Alice generates a random string of n bits again in the basis she chose. After encoding each bit into a qubit, Alice sends the photon to Bob.

4) Bob measures the sequence in his own basis and deduces an *n*-bit string.

5) The initial string proposed by Alice and the last string deduced by Bob from the previous step are compared bit by bit. The result is stored in a vector which represents the shared key as follows: if both bits are equal, the value is stored in the vector [23].

2. B92 protocol – a modified version of BB84 introduced by Charles Bennett in 1992. Instead of using four polarization states like BB84, B92 uses only two. In the rectilinear basis, a photon polarization of 0° is used to represent 0. And in the diagonal basis 1 is encoded at 45° [24][25].

3. The EPR protocol (also known as E91)

Protocol flow:

1) First, Alice generates a random binary sequence.

2) Afterwards, she creates EPR pairs of polarized photons for every bit and she keeps one particle for herself and the other particle of each pair is sent to Bob.

3) Each polarization that Alice kept is randomly measured by her. From there, she records each measurement and polarization type.

4) Thereafter, each particle that Bob receives is randomly measured. He then records each measurement type and the polarization measured to provide a new sequence.

5) Alice and Bob communicate with each other about the measurement types used and the data kept from all particle pairs, where they both chose the same measurement type for the sequence from step 1) and also the one from step 4).

6) After an established convention the matching information is converted by them to a string of bits [26].

II.4.1 Blockchain

Nowadays the necessity of modern technology has increased because it allows communication to be more convenient and, in some ways, more effective. Therefore, concerns about security are still taken into consideration. One solution in this scope was the concept of distributed networks that can be used in various financial services such as digital assets, remittance, online payment [27], such as cryptocurrency (i.e. Bitcoin) [28]. A new and powerful distributed system is

blockchain, which was proposed by Satoshi Nakamoto [29] for the first time in 2008 and implemented in 2009 [28].

Blockchain is a technology that combines old concepts like ledger with a peer-to-peer network to access a distributed database while preserving privacy and the following security properties: integrity, distribution, and tamper-proofing [29][30].

Regarding the architecture, blockchain is a sequence of blocks, which holds a complete list of transaction records. These blocks are hashed into a binary tree structure, with each block being linked to the previous one, named parent block, which also contains its own hash. The root block, called the genesis block, is hashed and stored alongside [28][30][31]. In other articles, the blocks also include timestamps to increase the security level [32].

Fig. 5 illustrates blockchain structure:

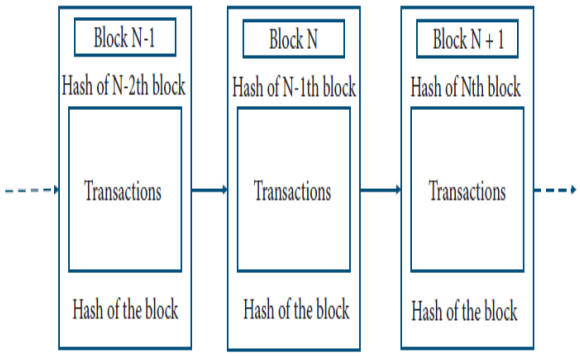

Fig. 5. Blockchain Structure [30]

Characteristics of Blockchain:

-*decentralization* – unlike the centralized transaction systems, the existence of a third party in blockchain is no longer required due to the consensus algorithms that are used to maintain data consistency in distributed networks;

-*persistency* – because of the blockchain structure, it is very difficult to delete or rollback transactions due to the fact they are stored in blocks and each one depends on the previous block; however, blocks that contain invalid transactions can be discovered swiftly;

-*anonymity* – the user can interact with the blockchain network with a randomly generated address, which does not reveal his real identity because blockchain is a decentralized system and no central authority is monitoring or recording user's private information;

-*auditability* – the transactions from a blockchain network are recorded by a digital distributed ledger and then validated by a digital timestamp; as a result, it is possible to audit and trace previous records by accessing any node in the network [27][28].

II.4.2 Blockchain Protocols

1) Shi-Cho Cha, Jyun-Fu Chen, Chunhua Su, and Kuo-Hui Yeh's Blockchain-Based Protocol

This protocol is used for sharing and access management of IoT (Internet of Things) device information. Three main entities are defined for this purpose: devices, users and Blockchain Connected Gateway (BCG). An overview of Cha et al.'s protocol can be seen in Fig. 6.

Firstly, the user searches the BCG smart contract address, which is implemented in the blockchain network, and after he finds it, he accesses the list of subset devices. Moreover, he must agree to the device's privacy policies if he wants to use any device. The agreement is stored in the blockchain network so that BCG can be used at the time of request to access the device information [29].

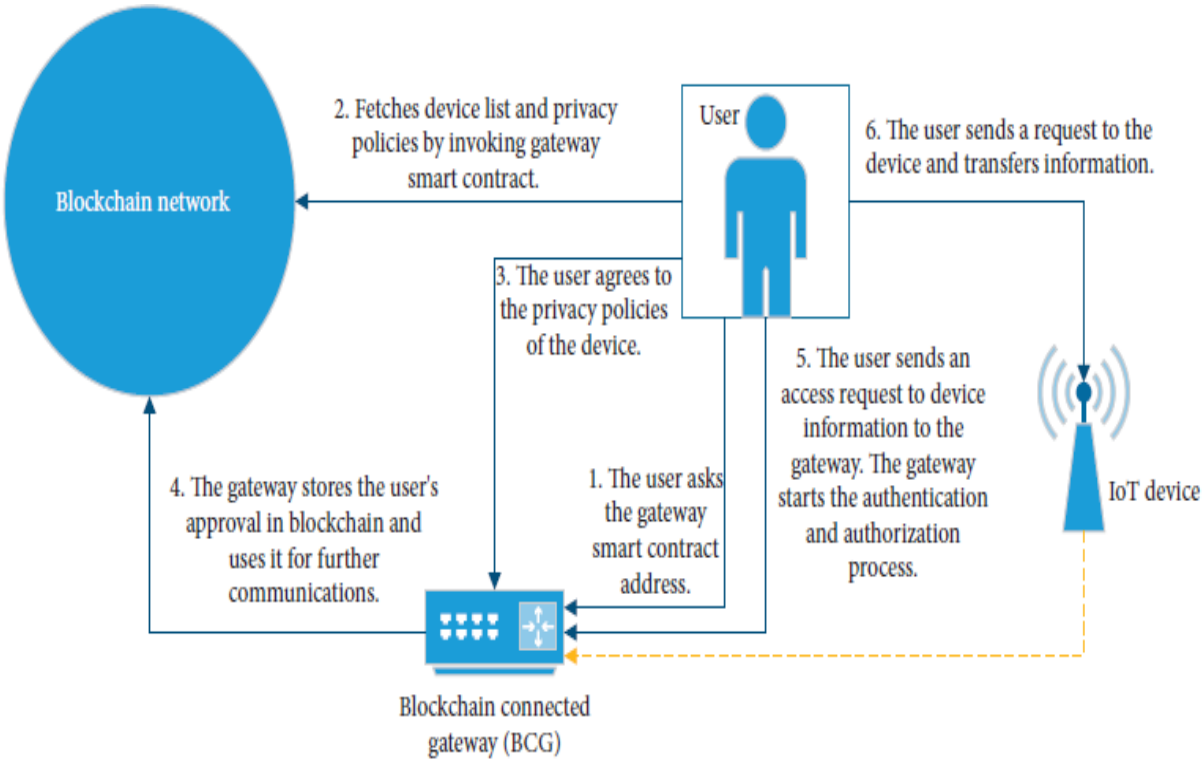

Fig. 6. Cha et al.'s Blockchain-Based Protocol [29]

2) Quantum Blockchain Protocol

Quantum blockchain is a decentralized, encrypted and distributed database. As the name implies, it is based on quantum computation and quantum information theory. For the protocol, it is based on a scheme proposed by Rajan and Visser in 2018 using entanglement in time, where microscopic particles such as photons that have never coexisted can also be entangled.

The communication security between nodes can be increased if either Quantum Secure Direct Communication (QSDC) or Quantum Key Distribution (QKD) is used. The authentication in the network is guaranteed by the properties of quantum physics. When blockchain protocols are used in combination with a quantum computer, the transaction processing speed increases. This advancement will greatly promote the development of cryptocurrency.

According to the experiments made, quantum blockchain is more secure and more efficient than a classical blockchain [32].

3) Blockchain-Based Routing Protocol for IoT Networks

In contrast to traditional routing protocols, the newly proposed Blockchain-based Contractual Routing (BCR) protocol operates in a distributed manner with no Central Authority (CA) that is required to authorize, add or remove IoT devices and has no secret key sharing mechanism. It is based on a classic routing protocol, Ad-hoc On-Demand Distance Vector (AODV), but BCR protocol comes up with something new: it utilizes smart contracts during route discovery in IoT network [33].

4) Data-Sharing Protocol under Blockchain-Based Cloud-Storage Architecture

This type of protocol is based on a meta-key mechanism, a remote data-sharing approach that enables data owners to share their encrypted data in the cloud without revealing the original key. It is used in a decentralized way. As far as key-ciphertext concerns, it is recorded into a blockchain system, so reliability and data security are ensured. The protocol is illustrated in Fig. 7 [34].

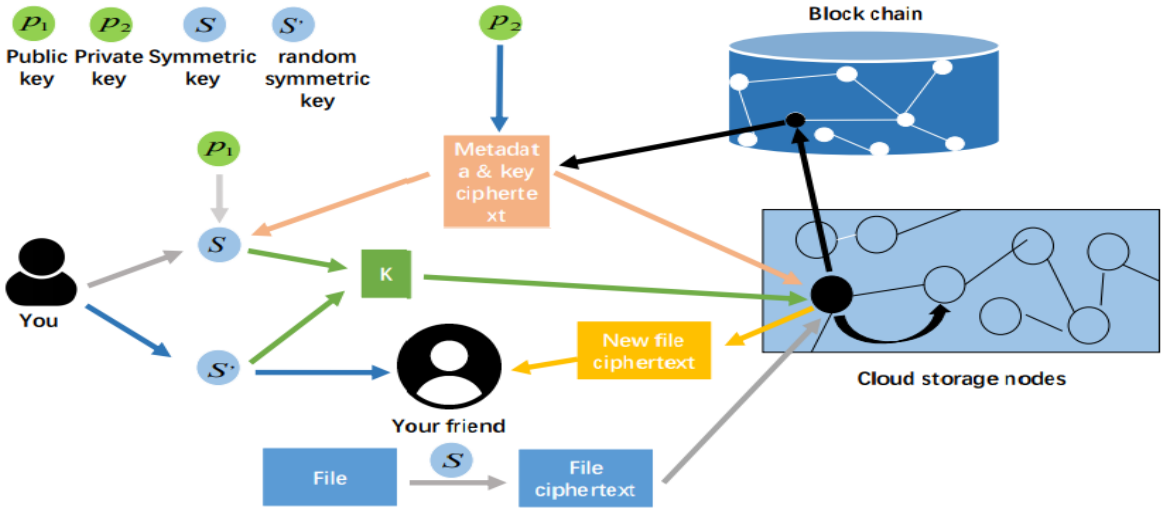

Fig. 7. Data-Sharing Protocol under Blockchain-Based Decentralized Storage Architecture [34]

III. Security

III.1 Classical key sharing algorithms in the context of quantum computing

Diffie-Hellman bases its security on the discrete logarithm, a problem which can be solved by quantum computers compromising its security.

A. Stolbunov proposed an asymmetric cryptographic scheme based on elliptic curve isogenies to remedy this problem; however, the resulting cryptosystem is inefficient as it takes 229 seconds to perform a 128-bit key exchange [35][36].

David Jao et al. attempted to resolve this issue by using isogenies over supersingular elliptic curves instead of ordinary ones to achieve the same operation in 7 seconds. This version is not vulnerable to any attacks that can be performed in less than exponential time even when quantum computers are involved [36]. However, the result is still just quantum resistant and not quantum proof [37].

Because the STS protocol is based on Diffie-Hellman, it is also vulnerable to quantum computers.

Kerberos is considered at least quantum resistant as long as public-key extensions are avoided. Because symmetric key encryption schemes remain quantum resistant, they may be the immediate answer to the threat of quantum computers. Buchanan et al. pointed out that there are multiple problems that stop this from happening immediately, one of them being the fact that Kerberos is not open source [38].

The QKD algorithms remain quantum safe, however, this does not mean all of them are secure. BB84 for example is vulnerable to a multitude of attacks [23].

The only existing possible issue is the cost of the equipment required to implement these protocols.

III.2 PKI in the context of quantum computing

One important feature of a PKI is the ability to prove the validity of certain websites of the Central Authority to generate and sign certificates. Most of the time the signing is done using the RSA algorithm, usually, using a 2048-bit key (4096 is also used). However, the biggest concern is that using Shor's algorithm on a quantum computer, one may be able to compute p, q from a public key's N faster than the legitimate RSA user can decrypt the message[38], public key which, anybody can obtain in a PKI from the CA in order to check such a certificate. After p and q have been obtained it is very easy to find the secret exponent. Now an attacker can falsify any certificate if he can obtain the respected public key. To apply Shor's algorithm over 2048-bit RSA, a quantum computer with more than 10 000 qubits would be required [40].

As of September 2020, the largest integer N computed was 35 and IBM promised 1000 qubits by 2023 [41].

NIST also estimates that a computer capable of breaking a 2000-bit RSA in a matter of hours could be built by 2030 for a budget of one billion dollars [42].

These are some solutions to the present problem:

1. Increasing the key size and using more than 2 primes for *N*

A simple solution to this issue would be to increase the size of the key and to use more primes instead of 2. J. Bernstein et al. suggested using 2^{31} primes and combining them into a 1-terabyte key using a very small *e*. The estimated time for the prime generation on the spare computing capacity of a 1400 core cluster was about 120 days. For a 2-terabyte key, the encryption took around 100 hours [39].

While the solution is not feasible for the average consumer due to the very long encryption/decryption times, a quantum computer would also require a large amount of qubits to break such a key, and, depending on the advancement of technology, assuming a quantum computer large enough to break a 2000-bit RSA will be built in the first place, could take substantially more time and funds.

2. Combining KSI (Keyless Signature Infrastructure) into PKI

A keyless signature infrastructure uses only hash functions to allow verification of the origin of data and create proof that the data was not modified since its creation.

Buldas et al. suggested that the keyless signature can be used as an evidence container so that even if the RSA key is broken it is still possible to verify the signature and reliability of the evidence because the keyless signature proves that the evidence was intact before it was broken. Hash functions are not quantum proof, however, they are quantum resistant. Therefore, to achieve the same level of security on a quantum computer as on a normal computer, the output will have to be larger [43].

3. Using lattice-based cryptography

Lattice-based cryptography is one of the important candidates for post-quantum cryptography because it can be used for encrypting data, for key-exchange protocols and signatures while remaining quantum-safe.

Hyeongcheol et al. proposed a quantum-resistant PKI scheme using the Ring Learning with Errors problem. The scheme is also decentralized and has a linear growth. It has the same properties as x509 v3 (non-repudiation, revocation, and scalability). It can also be maintained while offline because it does not require a central server [44].

III.3 PKI in the context of blockchain

It is necessary to demonstrate the identities of entities in the network and the traditional authentication of the IoT is generally based on the PKI, which requires a Central Authority. Blockchain, considered to be a distributed ledger technology, creates a trustless environment which can entirely remove the dependence on the Central Authority [30], eliminating the traditional PKI vulnerabilities, such as man-in-the-middle attacks or revocation of a certificate [45].

As far as transactions cost concern, it is cheaper when the user focuses only on public certificates. The result about the validity of transactions is faster in a classical PKI. However, blockchain comes up with some improvements like the establishment of a network for transactions from a given geographical area.

If sensitive or harmful transaction information is published by mistake, it can be deleted in a PKI. In contrast, data that has been stored in a blockchain is almost impossible to delete or alter. On the other hand, if one wants to track inspections logs, which cannot be modified, blockchain has a clear advantage.

A property that is only specific for blockchain is the possibility to include functionality of the smart contract. As far as smart contracts are concerned, they are self-executing scripts meaning that they can automatically conduct not only trade but also move value around without censorship or fraud. There is a great potential to digitalise international negotiations and trade transactions by exploiting this functionality [46].

In the context of smart contracts, an integrated framework for mobile blockchain is proposed to ensure key agreement between clients using Elliptic Curve Diffie-Hellman (ECDH) algorithm, which enables the use of smaller key sizes to maximize security level by a shared secret. Encrypted data from this framework is secure because an attacker cannot obtain the private keys of the communicating parties [47].

IV. Conclusions

Because the key distribution protocols serve as the first, and a very important, line of defense against attackers, it is important that they adapt extremely quickly to any potential threats. Adapting PKI however can be a very laborious task due to their sizes and complexity. This paper compares three different directions of cryptographic protocols, starting from the classical ones, which are insecure if one takes into account quantum computers, to the modern protocols based on quantum physics and blockchain. However, those also come with their own set of shortcomings which must be accounted for in order for them to be applicable in real world scenarios.

References

[1] Rahila Khan, Dr. C. Rama Krishna, "A review on key distribution protocols to achieve secure secret key distribution and mutual authentication"

[2] Maurer, Ueli. "Modelling a public-key infrastructure." *European Symposium on Research in Computer Security*. Springer, Berlin, Heidelberg, 1996.

[3] Adams, Carlisle, and Steve Lloyd. *Understanding public-key infrastructure: concepts, standards, and deployment considerations*. Sams Publishing, 1999.

[4] Kuhn, D. Richard, et al. *Introduction to public key technology and the federal PKI infrastructure*. National Inst of Standards and Technology Gaithersburg MD, 2001.

[5]https://brilliant.org/wiki/diffie-hellman-protocol/ – Last accessed 12 October 2020

[6] Manoj Ranjan Mishra, Jayaprakash Kar, "A STUDY ON DIFFIE-HELLMAN KEY EXCHANGE PROTOCOLS", International Journal of Pure and Applied Mathematics, Volume 114, No. 2, 2017, 179-189

[7]https://www.trendmicro.com/content/dam/trendmicro/global/en/migrated/security-intelligence-migration-spreadsheet/trendlabs-security-intelligence/2015/09/anglerek_dh_01.jpg (adapted version) – Last accessed 12 October 2020

[8]https://mathworld.wolfram.com/Diffie-HellmanProtocol.html – Last accessed 12 October 2020

[9]https://www2.kenyon.edu/Depts/Math/Aydin/Teach/Sp14/328/DH%20and%20ElGamal.pdf – Last accessed 2 January 2021

[10] Maurer, Ueli M. and Stefan Wolf. "The Diffie–Hellman protocol." Designs, Codes and Cryptography 19.2-3 (2000): 147-171.

[11] V. Cortier, "*Description of the Needham Schroeder public key protocol and its attack"* (https://members.loria.fr/VCortier/files/School/NS.pdf – Last accessed 14 October 2020)

[12] Lowe, Gavin. "Breaking and fixing the Needham-Schroeder public-key protocol using FDR." International Workshop on Tools and Algorithms for the Construction and Analysis of Systems. Springer, Berlin, Heidelberg, 1996.

[13]http://archive.dimacs.rutgers.edu/Workshops/Security/program2/boyd/node13.html –Last accessed 14 October 2020

[14] W. Diffie, P. van Oorschot and M. Wiener, "Authentication and Authenticated Key Exchange", Designs, Codes and Cryptography, 2, 1992, pp.107-125.

[15]http://www.hit.bme.hu/~buttyan/courses/Revkomaom/kex.pdf (adapted version) – Last accessed 17 October 2020

[16] Anita Narwal and Sunita Tomar, "*Kerberos Protocol: A Review"*, International Journal of Engineering Research & Technology (IJERT), Vol. 4 Issue 04, April-2015

[17]https://www.varonis.com/blog/kerberos-authentication-explained/ – Last accessed 9 January 2021

[18] Benioff, Paul (1980). "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". Journal of Statistical Physics.

[19] C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, volume 175, page 8. New York, 1984

[20] Elboukhari, Mohamed, Mostafa Azizi, and Abdelmalek Azizi. "Quantum Key Distribution Protocols: A Survey." *International Journal of Universal Computer Science* 1.2 (2010).

[21] Sato, Kazunobu, et al. "Novel applications of ESR/EPR: quantum computing/quantum information processing." EPR of Free Radicals in Solids II. Springer, Dordrecht, 2012. 163-204.

[22]https://www.researchgate.net/profile/Kamer_Vishi/publication/324115273/figure/fig1/AS:609979241345024@1522441792172/Key-exchange-in-the-BB84-protocol-implemented-with-polarization-of-photons-adapted-from.png – Last accessed 8 December 2020

[23] Mina Mihai-Zicu, and Emil Simion. "A Scalable Simulation of the BB84 Protocol Involving Eavesdropping."

[24] Padamvathi, V., B. Vishnu Vardhan, and A. V. N. Krishna. "Quantum cryptography and quantum key distribution protocols: a survey." 2016 IEEE 6th International Conference on Advanced Computing (IACC).IEEE,2016.

[25]https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/ – Last accessed 4 January 2021

[26] Rambabu Saini, "Quantum Cryptography Enhancement of QKD EPR Protocol and Identity Verification", International Journal of Engineering Sciences & Research, Oct. 2012

[27] Monrat, Ahmed Afif, Olov Schelén, and Karl Andersson. "A survey of blockchain from the perspectives of applications, challenges, and opportunities." IEEE Access 7 (2019): 117134-117151.

[28] Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE international congress on big data (BigData congress). IEEE, 2017.

[29] Yavari, Mostafa, et al. "An Improved Blockchain-Based Authentication Protocol for IoT Network Management." Security and Communication Networks 2020 (2020).

[30] Li, Dongxing, et al. "A blockchain-based authentication and security mechanism for iot." 2018 27th International Conference on Computer

Communication and Networks (ICCCN). IEEE, 2018.
[31] Garriga, Martin, et al. "Blockchain and cryptocurrencies: A classification and comparison of architecture drivers." Concurrency and Computation: Practice and Experience (2020): e5992.
[32] Li, Chuntang, et al. "Quantum Blockchain: A Decentralized, Encrypted and Distributed Database Based on Quantum Mechanics." Journal of Quantum Computing 1.2 (2019): 49.
[33] Ramezan, Gholamreza, and Cyril Leung. "A blockchain-based contractual routing protocol for the Internet of Things using smart contracts." Wireless Communications and Mobile Computing 2018 (2018).
[34] Li, Dagang, et al. "Meta-key: A secure data-sharing protocol under blockchain-based decentralized storage architecture." IEEE Networking Letters 1.1 (2019): 30-33.
[35] Stolbunov, Anton. "Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves." Advances in Mathematics of Communications 4.2 (2010): 215.
[36] Jao, David, and Luca De Feo. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies." International Workshop on Post-Quantum Cryptography. Springer, Berlin, Heidelberg, 2011.
[37] Koziel, Brian, et al. "NEON-SIDH: efficient implementation of supersingular isogeny Diffie-Hellman key exchange protocol on ARM." International Conference on Cryptology and Network Security. Springer, Cham, 2016.
[38] Buchanan, William, and Alan Woodward. "Will quantum computers be the end of public key encryption?." Journal of Cyber Security Technology 1.1 (2017): 1-22
[39] Bernstein, Daniel J., et al. "Post-quantum RSA." International Workshop on Post-Quantum Cryptography. Springer, Cham, 2017.
[40] Ziegler, Lynn. "Online security, cryptography, and quantum computing." (2015).
[41]https://www.sciencemag.org/news/2020/09/ibm-promises-1000-qubit-quantum-computer-milestone-2023 – Last accessed 10 November 2020
[42] Chen, Lily, et al. Report on post-quantum cryptography. Vol. 12. US Department of Commerce, National Institute of Standards and Technology, 2016.
[43] Buldas, Ahto, Risto Laanoja, and Ahto Truu. "Keyless signature infrastructure and PKI: hash-tree signatures in pre-and post-quantum world." International Journal of Services Technology and Management 23.1-2 (2017): 117-130.
[44]An, Hyeongcheol, and Kwangjo Kim. "QChain: Quantum-resistant and decentralized PKI using blockchain." Proc. SCIS. 2018.
[45] Talamo, Maurizio, et al. "A Blockchain based PKI Validation System based on Rare Events Management." Future Internet 12.2 (2020): 40.
[46] Rødseth, Ørnulf Jan, et al. "PKI vs. Blockchain when Securing Maritime Operations." (2019).
[47] Owoh, Nsikak Pius, and Manmeet Mahinderjit Singh. "Applying Diffie-Hellman Algorithm to Solve the Key Agreement Problem in Mobile Blockchain-based Sensing Applications." Int. J. Adv. Comput. Sci. Appl 10.3 (2019): 59-68.