entropy

MDPI

Article

Multi-Party Quantum Byzantine Agreement without Entanglement

Xin Sun [1], Piotr Kulicki [1,*] and Mirek Sopek [2]

[1] Department of the Foundations of Computer Science, John Paul II Catholic University of Lublin, 20-950 Lublin, Poland; xin.sun.logic@gmail.com
[2] MakoLab SA, 91-062 Lodz, Poland; sopek@makolab.com
* Correspondence: kulicki@kul.pl

Received: 4 September 2020; Accepted: 9 October 2020; Published: 14 October 2020

Abstract: In this paper, we propose a protocol of quantum communication to achieve Byzantine agreement among multiple parties. Our protocol's striking feature compared to the existing protocols is that we do not use entanglement to achieve the agreement. The role played by entangled states in other protocols is replaced in our protocol by a group of semi-honest list distributors. Such a replacement makes the implementation of our protocol more feasible. Moreover, our protocol is efficient in the sense that it achieves agreement in only three rounds which is a significant improvement with respect to the alternative agreement protocol not using entanglement. In the first round, a list of numbers that satisfies some special properties is distributed to every participant by list distributors via quantum secure communication. Then, in the second and third rounds, those participants exchange some information to reach an agreement.

Keywords: Byzantine agreement; quantum communication; distributed computing

1. Introduction

A fundamental problem in distributed computing is how to reach an agreement in the presence of faulty processes. For example, a database can be replicated on several computers, which ensures access to the database even if some of the computers are not functional. For the consistency of data, all computers must preserve the same contents. To achieve this goal, a protocol that ensures that all computers adopt the same update of the database is needed. This problem is intuitively formulated as the Byzantine generals problem:

"Three generals of the Byzantine army want to decide upon a common plan of action: either to attack (0) or to retreat (1). They can only communicate in pairs by sending messages. One of the generals, the commanding general, must decide on a plan of action and communicate it to the other generals. However, one of the generals might be a traitor, trying to keep the loyal generals from agreeing on a plan. How to find a way in which all loyal generals follow the same plan?"

If the generals communicate with each other only by pairwise classical channels, the Byzantine generals problem is provably unsolvable [1,2]. Even if pairwise quantum channels are used, it will not help to solve the problem [3]. However, a variation of the Byzantine agreement problem, called detectable Byzantine agreement (DBA), can be solved using quantum resources. A DBA protocol ensures that either all loyal generals agree upon a common plan or all abort. In addition, if all generals are loyal, then they agree upon a common plan.

In 2001, Fitzi et al. [4] presented a DBA protocol for three parties using pairwise quantum channels and entangled qutrits. Cabello [5] proposed a three-party DBA protocol based on a four-qubit singlet state. Iblisdir and Gisin [6] developed an improvement of the protocol of Fitzi et al. [4] by showing that the DBA problem can be solved by using two quantum key distribution channels and

three classical authenticated channels. Gaertner et al. [7] introduced a new DBA protocol based on four-qubit entangled state. An experimental implementation of the protocol was also presented by Gaertner et al. [7]. A device-independent quantum scheme for the Byzantine generals problem was provided by Rahaman et al. [8].

All the aforementioned DBA protocols only consider the situation of three parties. In actual distributed computing or in blockchains [9,10], the number of parties involved is significantly larger than three. Ben-Or and Hassidim [11], Tavakoli et al. [12] and Luo et al. [13] developed DBA protocols for multiple parties based on high-dimensional entangled states. These states are difficult to realize by the current technology. In this paper, we develop a new DBA protocol for multiple parties. The crucial feature of our protocol compared to the existing ones is that no entanglement is used in an essential way. The role played by entangled states in other protocols is replaced by a group of semi-honest list distributors in our protocol. Such a replacement makes our protocol easier to implement. The quantum technology that we use is Quantum Key Distribution (QKD) [14], which is a relatively mature technology and an active topic of research and has recently attracted the industry's interest. It is important to note that, in some QKD protocols (e.g., [15]), the phenomenon of entanglement is used directly on the physical level, and in others entanglement is used to assist and improve standard QKD protocols such as BB84 [16,17]. However, our quantum Byzantine Agreement protocol uses QKD on the data transmission level, so the entanglement usage on the physical level is not relevant on the essential level of our protocol.

To the best of our knowledge, the only existing work on DBA without using entanglement was presented by Fitzi et al. [18]. While the protocol of Fitzi et al. [18] requires $f + 5$ rounds to reach an agreement, where f is the number of faulty parties, our protocol is more efficient in the sense that agreement can be achieved in three rounds.

The structure of this paper is as follows. In Section 2, we introduce our protocol. Then, in Section 3, we analyze the properties of our protocol. We conclude the paper with future work in Section 4.

2. Quantum Byzantine Agreement without Entanglement

Let us begin with formal definitions of Byzantine agreement.

Definition 1. *[Byzantine agreement (BA) protocol [4]] A protocol among n parties such that one distinct party S (the sender) holds an input value $x_s \in D$ (for some finite domain D) and all other parties (the receivers) eventually decide on an output value in D is said to achieve Byzantine agreement if the protocol guarantees that all honest parties decide on the same output value $y \in D$ and that $y = x_s$ whenever the sender is honest.*

Definition 2. *[Detectable Byzantine agreement (DBA) protocol [4]] A protocol among n parties such that one sender S holds an input value $x_s \in D$ and all other receivers eventually decide on an output value in D is said to achieve detectable Byzantine agreement if the protocol guarantees the following:*

1. *Agreement: Either all honest parties abort the protocol or all honest parties decide on the same output value $y \in D$.*
2. *Validity: If all parties are honest, then they decide on the same output value $y = x_s$.*

We assume the parties are pairwise connected by a classical channel and a quantum channel. Both channels are error-free and synchronous. Being synchronous means that parties share a discrete global clock that starts out at time 0 and advances by increments of one. Communication proceeds in a sequence of rounds, with round k taking place between time $k - 1$ and time k. In each round, every party first sends the messages it needs to send to other parties, and then it receives the messages that were sent to it by other parties in the same round. The classical channels are further assumed to be authenticated, which means that every party can always send classical messages directly to every other party. The adversary can neither prevent those messages from being sent nor introduce new messages on the channel. The quantum channels, however, are insecure in the sense that quantum

messages may be tampered by the adversary. Those assumptions ensures that parties can pairwise establish secret keys by quantum key distribution. Indeed, we assume that secret keys with sufficient length are already established before our protocol starts. These assumptions also ensure the parties to create unconditionally secure signatures [19,20] for the communication of classical information. Thus, we assume that all classical messages in our protocol are signed by an unconditionally secure signature scheme.

Now, we introduce our protocol. It consists of three rounds. The aim of the first round is to distribute correlated lists of numbers among the parties involved in the protocol. We call such a list *reference list* since the parties refer to that lists to check whether the information they receive is trustworthy. Then, in the second and third rounds, parties use the reference lists to achieve consensus. We assume the existence of semi-honest parties to handle the task of reference list distribution. This assumption is similar to the one made by Luo et al. [13]. For a party to be semi-honest means that the party acts according to the description of the protocol, but may disclose information with a certain probability p, $0 < p < 1$.

2.1. Round 1: List Distribution

Let $\{P_1, \ldots, P_n, P_{n+1}, \ldots, P_{n+d}\}$ be a set of parties. Let further P_1 be the sender of the DBA protocol, $P_2, \ldots, P_n$ be receivers and $P_{n+1}, \ldots, P_{n+d}$ be list distributors. To distinguish the sender and the receivers from the distributors, we also call the former two *participants*. We assume that $P_{n+1}, \ldots, P_{n+d}$ are semi-honest. For every party $P_i \in \{P_{n+1}, \ldots, P_{n+d}\}$, the task of P_i is to use the technique of quantum secure communication (communicate with the encryption/decryption keys distributed by quantum key distribution) to send a list of numbers L_k^i (a reference list) to each $P_k \in \{P_1, \ldots, P_n\}$ such that the following is satisfied:

1. For all $k \in \{1, \ldots, n\}$, $|L_k^i| = m$ for some integer m which is a multiple of 6.
2. $L_1^i \in \{0,1,2\}^m$. $\frac{m}{3}$ numbers on L_1^i are 0. $\frac{m}{3}$ numbers on L_1^i are 1. $\frac{m}{3}$ numbers on L_1^i are 2.
3. For all $k \in \{2, \ldots, n\}$, $L_k^i \in \{0,1\}^m$.
4. For all $j \in \{1, \ldots, m\}$, if $L_1^i[j] = 0$, then $L_2^i[j] = \ldots = L_n^i[j] = 0$.
5. For all $j \in \{1, \ldots, m\}$, if $L_1^i[j] = 1$, then $L_2^i[j] = \ldots = L_n^i[j] = 1$.
6. For all $j \in \{1, \ldots, m\}$, if $L_1^i[j] = 2$, then for all $k \in \{2, \ldots, n\}$ the probability that $L_k^i[j] = 0$ and that $L_k^i[j] = 1$ are equal (i.e., the numbers of occurences of 0 and 1 are equal in the list).
7. For all $j, k \in \{1, \ldots, m\}$, $L_j^i = L_k^i$.

Distributors create their lists independently; thus, for different i and j, the lists L_1^i and L_1^j may be different (indeed, the probability that they are the same is quite small).

After the lists are distributed, $P_1, \ldots, P_n$ use sequential composition to form a longer list to be used in the next stage: $L_1 = L_1^{n+1} \ldots L_1^{n+d}, \ldots, L_n = L_n^{n+1} \ldots L_n^{n+d}$.

Obviously, $L_2 = L_3 = ... = L_n$. We call the longer lists *combined reference lists*. Notice that every distributor contributes $\frac{1}{d}$ to the combined reference lists.

2.2. Rounds 2 and 3: Reaching Agreement

Now, the parties $P_1, \ldots, P_n$ run the following steps to reach an agreement:

Round 2 P_1 sends a binary number $b_{1,k}$ to all P_k, $k \in \{2, \ldots, n\}$. Together with $b_{1,k}$, P_1 sends to P_k the list of numbers $ID_{1,k}$, which indicate all positions of $b_{1,k}$ on the list L_1. The length of $ID_{1,k}$ is to be $\frac{md}{3}$, where md is the length of L_1. P_1 uses $b_{1,k}$ as the final value it outputs.

Round 3 P_k checks the obtained message $(b_{1,k}, ID_{1,k})$ against his own reference list L_k. If the analysis of P_k shows that $(b_{1,k}, ID_{1,k})$ is consistent with L_k, then he sets $(b_{k,j}, ID_{k,j}) := (b_{1,k}, ID_{1,k})$ and sends $(b_{k,j}, ID_{k,j})$ to all other receivers P_j, $j \in \{2, \ldots, n\}$. Here, $(b_{1,k}, ID_{1,k})$ is consistent with L_k means that for all index $x \in ID_{1,k}$, $L_k[x] = b_{1,k}$. However, if $(b_{1,k}, ID_{1,k})$ is not consistent with L_k, then P_k immediately ascertains that P_1 is dishonest and sends to other

receivers $P_j, j \in \{2, \ldots, n\}$ message: $\perp$, meaning: "I have received an inconsistent message". To acknowledge the fact that every receiver knows his own output, we formally assume that each of them receives a message from himself.

Time 3 After all messages have been exchanged between the receivers, every P_k analyzes the data received from $P_2, \ldots, P_n$ and acts according to the following criteria:

(a) If there is a set of receivers H with $|H| \geq 2$ such that, for all $j \in H$, $(b_{j,k}, ID_{j,k})$ is consistent with L_k, and for some $i, j \in H$, $b_{i,k} \neq b_{j,k}$, then P_k sets his output value to be $\perp$.

(b) If there is a set of receivers H with $|H| \geq 2$ such that for all $j \in H$, $(b_{j,k}, ID_{j,k})$ is consistent with L_k and all $b_{j,k}$ are the same, and for all $i \notin H$, $(b_{i,k}, ID_{i,k})$ is not consistent with L_k, then P_k sets his output value to be $b_{j,k}$.

(c) If there is a set of receivers H with $|H| \geq 2$ such that for all $j \in H$, $(b_{j,k}, ID_{j,k})$ is consistent with L_k and all $b_{j,k}$ are the same, and for all $i \notin H$, the message sent by P_i is $\perp$, then P_k sets his output value to be $b_{j,k}$.

(d) In all other cases, P_k sets his value to be $\perp$.

Criteria (a)–(d) are crucial for our protocol. Let us now briefly explain the rationale behind them. In a nutshell, the most important factor here is the following claims:

Theorem 1. *For all $k, j \in \{2, \ldots, n\}$, P_k believes that P_j is honest whenever $(b_{j,k}, ID_{j,k})$ is consistent with L_k.*

Proof. We prove the theorem by showing that if P_j is dishonest then the probability that P_j sets $(b_{j,k}, ID_{j,k})$ to be consistent with L_k is extremely small, when $b_{j,k} \neq b_{1,j}$.

Suppose P_j is dishonest and $b_{j,k} \neq b_{1,j}$. Now, P_j wants to construct the message $(b_{j,k}, ID_{j,k})$ and send to P_k such that $(b_{j,k}, ID_{j,k})$ is consistent with L_k. Note that, in $L_j = L_k$, there are $\frac{md}{2}$ positions on which $b_{j,k}$ appears. However, on L_1, there are only $\frac{md}{3}$ positions on which $b_{j,k}$ appears. We say that a position x is a *discord* position iff $L_1[x] = 2$. If P_j selects a discord position x and puts it into $ID_{j,k}$, then with probability $\frac{1}{3}$ it will be that $L_k[x] \neq b_{j,k}$. To ensure that $(b_{j,k}, ID_{j,k})$ is consistent with L_k, P_j has to make a correct choice on all indexes. Therefore, the probability of making a correct choice on all indexes is $(\frac{2}{3})^{\frac{md}{3}}$, which is extremely small when md is relatively large. Therefore, if it is the case that $(b_{j,k}, ID_{j,k})$ is consistent with L_k, then P_k can conclude that P_j is honest. □

Theorem 2. *For all $k, j \in \{2, \ldots, n\}$, if P_j is honest and he sends $(b_{j,k}, ID_{j,k})$ to P_k, then $(b_{j,k}, ID_{j,k})$ is consistent with L_k.*

Proof. If P_j is honest and he sends $(b_{j,k}, ID_{j,k})$ to P_k, then $(b_{j,k}, ID_{j,k}) = (b_{1,j}, ID_{1,j})$ must be consistent with L_j. Since $L_j = L_k$, we know that $(b_{j,k}, ID_{j,k})$ is consistent with L_k. □

Thus, any receiver P_k can conclusively deduce about any other receiver P_j what follows:

- If P_j has sent a message consistent with L_k, then P_j is honest.
- If P_j has sent a message inconsistent with L_k, then P_j is dishonest.
- If P_j has sent $\perp$, then P_j may be honest or dishonest. However, if in this case P_j is honest, then P_1 must be dishonest.

The rationale of Criterion (a) follows from Theorem 1. P_k can conclude that P_i and P_j are honest when $(b_{i,k}, ID_{i,k})$ and $(b_{j,k}, ID_{j,k})$ are consistent with L_k. Now, if in addition $b_{i,k} \neq b_{j,k}$, P_k can safely conclude that the sender (P_1) is dishonest. Consequently all the messages are not trustworthy and the output $\perp$ is adequate for the situation.

As for Criterion (b), according to Theorem 1, we may conclude that all the receivers from the set H are honest and all others are not. Thus, H is the set of all honest receivers and their common

message is trustworthy. Criterion (c) is similar to Criterion (b). Receivers from H here are also honest. However, in this case, some participants who are not in H may also be honest. The honest ones finally will change their output value from $\perp$ to $b_{j,k}$. For safety reasons with respect to the agreement condition of DBA presented in Definition 2 by Criterion (d) in all other cases honest parties abort our protocol by setting their output to $\perp$.

3. Analysis of the Protocol

Now, let us analyze the performance of our protocol under an attack of an adversary. We make the following assumption about the adversary:

1. The adversary can control a fixed set of participants and let those participants send arbitrary messages at his will. A participant is dishonest if and only if he is controlled by the adversary. The amount of honest participants is $\geq$3.
2. The adversary can bribe the list distributors to disclose certain information. When being bribed, a list distributor will disclose information with probability p.
3. The adversary has unlimited computing power.

In short, the adversary is static, Byzantine and with unlimited computing power.

Theorem 3. *Our protocol satisfies agreement and validity under the attack of an adversary.*

Proof. It is easy to see that validity is satisfied. Indeed, if none of the participants is controlled by the adversary, then they behave as the protocol specifies. Even if the adversary collects information from a large number of list distributors, the correlated list of numbers will still be correctly distributed. All participants will send consistent messages and the same output value will be established.

We now turn to the proof of agreement. First, note that the adversary can hardly have complete information of the combined reference lists $(L_1, \ldots, L_n)$. By our assumption, every list distributor is semi-honest. They will disclose the content of the list that they distributed with probability $p < 1$, if the adversary bribes them. Since every list distributor contributes only $\frac{1}{d}$ to the lists, to collect complete information about $L_1, \ldots, L_n$, the adversary must bribe all d list distributors and still the probability of collecting complete information is p^d, which decreases exponentially as d grows. For those list distributors that the adversary does not bribe, the adversary cannot collect any information because the lists are distributed by quantum secure communication. The unlimited computing power the adversary has is not helpful in this case. Therefore, we conclude that the first stage of our protocol can be correctly and safely executed.

Now, we consider the second and third rounds. If the sender is honest, then there are at least two honest receivers. All honest receivers will receive the same consistent data from the sender. Those honest receivers will forward the same data to other participants. Therefore, according to Criterion (b) in our protocol, all honest participants will output the same value as the sender. If the sender P_1 is dishonest, then there are two cases:

1. All honest receivers receive consistent data. In this case, there are two sub-cases:
   (a) All honest receivers receive the same data. In this case, according to Criterion (b), all honest participants will output the same value.
   (b) Not all honest receivers receive the same data. Then, according to Criterion (a), all honest receivers will abort the protocol (output $\perp$).
2. Not all honest receivers receive consistent data. In this case, if there are still two receivers that receive the same and consistent data and all other receivers output $\perp$, then, according to Criterion (c), all honest receivers will output the same value. Otherwise, according to Criterion (a) or Criterion (d), all honest receivers will output $\perp$.

Therefore, in all possible cases, the agreement is achieved. □

The above proof also implies an interesting property of our protocol which is stronger than validity. We present it as a corollary.

Corollary 1. *Our protocol satisfies the following honest-success property under the attack of an adversary: if the sender is honest, then all honest parties decide on the same output as the sender.*

4. Conclusions and Future Work

We propose a protocol of quantum communication to achieve detectable Byzantine agreement among multiple parties. The significant feature of our protocol, compared to most existing protocols, is that it does not use entanglement. The success of our protocol relies on the distribution of sequences of reference lists, which in turn relies on the unconditional security of QKD. The way QKD is obtained is beyond the scope of the paper; we can just mention that in principle QKD can also be implemented without entanglement even if in some proposals the performance of QKD is improved by using two-qubit entangled state [16].

We also assume the participation of semi-honest list distributors in the protocol. This assumption is the price to pay for not using entanglement. Since low-dimensional entanglement can be implemented by current technology, in the future, we will study whether semi-honest distributors could be replaced by low-dimensional entanglement. One potential application of our DBA protocol is in the field of quantum blockchain [21–23]. In the future, we plan to apply our protocol to quantum blockchain to solve particular problems such as auction, lottery and multi-party secure computation.

Author Contributions: Conceptualization, X.S.; formal analysis, X.S. and P.K.; methodology, X.S. and M.S.; project administration, P.K.; supervision, P.K.; validation, P.K. and M.S.; writing—original draft, X.S.; writing—review & editing, P.K. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: The project is funded by the Minister of Science and Higher Education within the program under the name "Regional Initiative of Excellence" in 2019–2022, project number: 028/RID/2018/19, the amount of funding: 11 742 500 PLN.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pease, M.C.; Shostak, R.E.; Lamport, L. Reaching Agreement in the Presence of Faults. *J. ACM* **1980**, *27*, 228–234. [CrossRef]
2. Lamport, L.; Shostak, R.E.; Pease, M.C. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [CrossRef]
3. Fitzi, M.; Garay, J.A.; Maurer, U.; Ostrovsky, R. Minimal Complete Primitives for Secure Multi-Party Computation. *J. Cryptol.* **2005**, *18*, 37–61. [CrossRef]
4. Fitzi, M.; Gisin, N.; Maurer, U. Quantum Solution to the Byzantine Agreement Problem. *Phys. Rev. Lett.* **2001**, *87*, 217901.
5. Cabello, A. Solving the liar detection problem using the four-qubit singlet state. *Phys. Rev. A* **2003**, *68*, 012304.
6. Iblisdir, S.; Gisin, N. Byzantine agreement with two quantum-key-distribution setups. *Phys. Rev. A* **2004**, *70*, 034306.
7. Gaertner, S.; Bourennane, M.; Kurtsiefer, C.; Cabello, A.; Weinfurter, H. Experimental Demonstration of a Quantum Protocol for Byzantine Agreement and Liar Detection. *Phys. Rev. Lett.* **2008**, *100*, 070504.
8. Rahaman, R.; Wieśniak, M.; Żukowski, M. Quantum Byzantine agreement via Hardy correlations and entanglement swapping. *Phys. Rev. A* **2015**, *92*, 042302.
9. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 3 October 2020).

10. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; Caro, A.D.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, 23–26 April 2018; pp. 1–15. [CrossRef]
11. Ben-Or, M.; Hassidim, A. Fast quantum byzantine agreement. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005*; Gabow, H.N., Fagin, R., Eds.; ACM: New York, NY, USA, 2005; pp. 481–485. [CrossRef]
12. Tavakoli, A.; Cabello, A.; Żukowski, M.; Bourennane, M. Quantum Clock Synchronization with a Single Qudit. *Sci. Rep.* **2015**, *5*, 7982.
13. Luo, Q.; Feng, K.; Zheng, M. Quantum Multi-valued Byzantine Agreement Based on d dimensional Entangled States. *Int. J. Theor. Phys.* **2019**, *58*, 4025–4032. [CrossRef]
14. Bennetta, C.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, 9–12 December 1984; pp. 175–179.
15. Ekert, A.K. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* **1991**, *67*, 661–663. [CrossRef] [PubMed]
16. Koashi, M.; Preskill, J. Secure Quantum Key Distribution with an Uncharacterized Source. *Phys. Rev. Lett.* **2003**, *90*, 057902. [CrossRef] [PubMed]
17. Wang, X.B. A practically feasible entanglement assisted quantum key distribution protocol. *arXiv* **2003**, arXiv:quant-ph/0306156.
18. Fitzi, M.; Gottesman, D.; Hirt, M.; Holenstein, T.; Smith, A.D. Detectable byzantine agreement secure against faulty majorities. In Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, PODC 2002, Monterey, CA, USA, 21–24 July 2002; Ricciardi, A., Ed.; ACM: New York, NY, USA, 2002; pp. 118–126. [CrossRef]
19. Amiri, R.; Andersson, E. Unconditionally Secure Quantum Signatures. *Entropy* **2015**, *17*, 5635–5659. [CrossRef]
20. Amiri, R.; Abidin, A.; Wallden, P.; Andersson, E. Efficient Unconditionally Secure Signatures Using Universal Hashing. In *Proceedings of the Applied Cryptography and Network Security—16th International Conference, ACNS 2018, Leuven, Belgium, 2–4 July 2018*; Lecture Notes in Computer Science; Preneel, B., Vercauteren, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10892, pp. 143–162._8. [CrossRef]
21. Kiktenko, E.O.; Pozhar, N.O.; Anufriev, M.N.; Trushechkin, A.S.; Yunusov, R.R.; Kurochkin, Y.V.; Lvovsky, A.I.; Fedorov, A.K. Quantum-secured blockchain. *Quantum Sci. Technol.* **2018**, *3*, 035004.
22. Sun, X.; Wang, Q.; Kulicki, P.; Sopek, M. A Simple Voting Protocol on Quantum Blockchain. *Int. J. Theor. Phys.* **2019**, *58*, 275–281. [CrossRef]
23. Sun, X.; Sopek, M.; Wang, Q.; Kulicki, P. Towards Quantum-Secured Permissioned Blockchain: Signature, Consensus, and Logic. *Entropy* **2019**, *21*, 887. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).