Research
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

Beating the Fault-Tolerance Bound and Security Loopholes for Byzantine Agreement with a Quantum Solution

Chen-Xun Weng[1†], Rui-Qi Gao[1†], Yu Bao[1], Bing-Hong Li[1], Wen-Bo Liu[1], Yuan-Mei Xie[1], Yu-Shuo Lu[1], Hua-Lei Yin[1,2*], and Zeng-Bing Chen[1*]

[1]National Laboratory of Solid State Microstructures and School of Physics, Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, China. [2]Department of Physics and Beijing Key Laboratory of Opto-electronic Functional Materials and Micro-nano Devices, Key Laboratory of Quantum State Construction and Manipulation (Ministry of Education), Renmin University of China, Beijing 100872, China.

*Address correspondence to: hlyin@ruc.edu.cn (H.-L.Y); zbchen@nju.edu.cn (Z.-B.C.)
†These authors contributed equally to this work.

Citation: Weng CX, Gao RQ, Bao Y, Li BH, Liu WB, Xie YM, Lu YS, Yin HL, Chen ZB. Beating the Fault-Tolerance Bound and Security Loopholes for Byzantine Agreement with a Quantum Solution. *Research* 2023;6:Article 0272. https://doi.org/10.34133/research.0272

Submitted 10 October 2023
Accepted 1 November 2023
Published 21 November 2023

Copyright © 2023 Chen-Xun Weng et al. Exclusive licensee Science and Technology Review Publishing House. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License 4.0 (CC BY 4.0).

Byzantine agreement, the underlying core of blockchain, aims to make every node in a decentralized network reach consensus. Classical Byzantine agreements unavoidably face two major problems. One is 1/3 fault-tolerance bound, which means that the system to tolerate f malicious players requires at least $3f+1$ players. The other is the security loopholes from its classical cryptography methods. Here, we propose a Byzantine agreement framework with unconditional security to break this bound with nearly 1/2 fault tolerance due to multiparty correlation provided by quantum digital signatures. It is intriguing that quantum entanglement is not necessary to break the 1/3 fault-tolerance bound, and we show that weaker correlation, such as asymmetric relationship of quantum digital signature, can also work. Our work strictly obeys two Byzantine conditions and can be extended to any number of players without requirements for multiparticle entanglement. We experimentally demonstrate three-party and five-party consensus for a digital ledger. Our work indicates the quantum advantage in terms of consensus problems and suggests an important avenue for quantum blockchain and quantum consensus networks.

Introduction

Byzantine agreement requires solving the fundamental consensus problem initially posed in 1982 known as the Byzantine Generals Problem, which can ensure the smooth functioning of a decentralized system under the attacks of malicious players [1,2]. This problem can be translated into a "commanding general-lieutenants" model, where the commanding general is randomly selected from among all the Byzantine generals and the others become lieutenants to reach consensus on the commanding's order (see Supplementary Materials Section A for details). For a strict Byzantine agreement, there are two necessary interactive consistency (IC) Byzantine conditions. The first is that all loyal lieutenants obey the same order (IC_1), and the second is that every loyal lieutenant obeys the order of the commanding general if the commanding general is loyal (IC_2). Only when both conditions are satisfied can the system reach consensus. For an N-party system, however, classical Byzantine agreement (CBA) protocols [3–9] that tolerate f malicious players require $N \geq 3f+1$ players; namely, the fault-tolerance bound is 1/3 [10–13]. Thus, the three-party consensus problem is naturally unsolvable for CBA even using the authentication classical channel [14]. The other issue is the security loopholes of CBA's widely used public-key encryption and one-way hash function [15], which are seriously threatened by quantum computing [16–25].

Quantum Byzantine agreement (QBA) is a promising approach for consensus problems. For three-party consensus, the first quantum solution using three-qutrit singlet states was proposed in 2001 [26] and was experimentally demonstrated using four-photon polarization-entangled states in 2008 [27]. This protocol and its subsequent protocols [28–32] using some special entanglement, called detectable QBA framework, unavoidably weaken the two original Byzantine conditions with extra assumptions, which leads to a certain probability of aborting the protocol. More seriously, these rudimentary solutions are restricted to the three-party scenario and can only reach a one-bit message consensus [26,27,29–32]. Some achievements have been made toward scalable multiparty QBA [33–36], but their fault tolerance is 1/3. In addition, QBA protocols require sophisticated techniques, such as multiparticle entanglement generation and distribution and entanglement swapping, which are difficult for practical implementations. Furthermore, the security of detectable QBA has not been proven rigorously [37,38].

Intriguingly, quantum entanglement is not necessary to break the 1/3 fault-tolerance bound and weaker correlation can also work. Here, different from detectable QBA, we propose a strict

information-theoretical secure Byzantine agreement framework that exploits the recursion structure [39] and quantum digital signatures (QDS) [40–42] to address the limitation of fault-tolerance bound and security loopholes (see Table 1). It completely breaks the 1/3 fault-tolerance bound with a fault tolerance of $N \geq 2f+1, \forall f \in \mathbb{N}^+$ while strictly obeying IC_1 and IC_2 due to multiparty correlation provided by QDS. Our work is highly adaptable, because it can be achieved by any type of QDS, including the original proposal of GC01-QDS [40] and its variants such as orthogonal encoding [43–51] and non-orthogonal encoding [52–55], and OTUH-type QDS [56,57]. As QDS advances by leaps and bounds, it only requires coherent states instead of complex multiparticle entanglement and quantum memory [41,43,52]. Generating and maintaining entanglement is a sticking point in experimental setups, and the ability to relax this requirement can reduce the complexity of consensus systems and serve as a foundation for further research. Furthermore, our protocol is able to achieve consensus on multiple messages. In addition, we implement proof-of-principle experiments of the three-party and five-party consensus with three different QDS protocols, BB84 GC01-QDS [43], OTUH-QDS [56], and OTUH-QDS without perfect keys [57].

Results

Protocol definition

Before stating our QBA framework, we introduce the multicast round. In a multicast round with n players, there is a primary, and the others are backups. The primary multicasts his or her message to the backups by the following operation, as shown in Fig. 1. One of the backups is selected as the forwarder, and the other unchosen backups become verifiers. The primary, the forwarder, and one of the verifiers perform a three-party QDS to transmit the message. QDS is divided into two stages—distribution and messaging stage. The distribution stage is to distribute correlated quantum keys to the players. The messaging stage uses some classical operations and quantum keys to complete digital signatures. The messaging stage contains three steps: signing, forwarding, and verification. The primary signs the message and then sends the message and corresponding signature to the forwarder. After that, the forwarder will forward the message and signature to the verifier. Only when both the forwarder and verifier accept the signature, the signing is successful, i.e., the primary cannot deny the fact that she signed the message (nonrepudiation), and the message cannot be forged by others including the forwarder (unforgeability) (see Materials and Methods for details). For a chosen forwarder, the verifiers take turns participating in such three-party QDS. The above process will be repeated until all backups have acted as the forwarder one time. In the end, each backup records a list of $n-1$ messages, consisting of one message directly received from the primary and $n-2$ messages forwarded by other backups. We call this list of messages broadcasting list in the later. Note that a complete multicast round consists of three steps: (1) sign and multicast, (2) forward, and (3) verify and record.

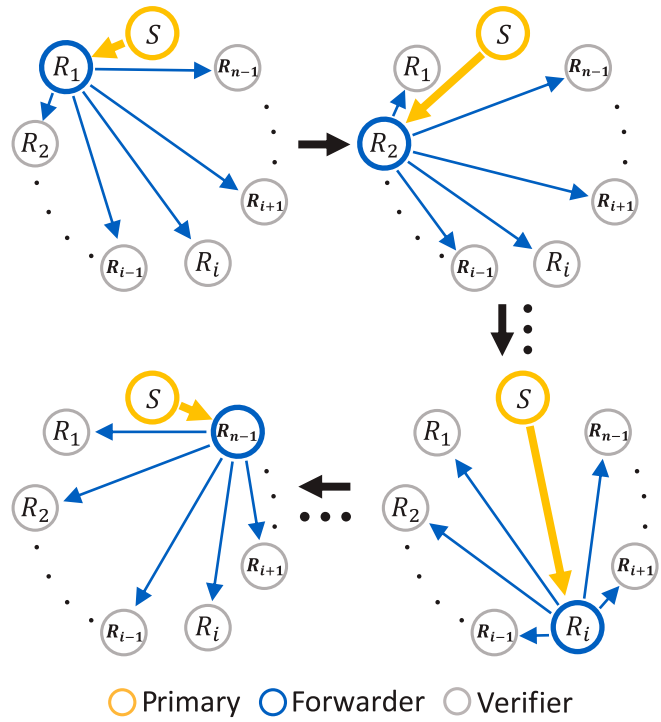


Fig. 1. Schematic of a multicast round including n players with primary S. The primary S signs and then multicasts his message to the backups. The forwarder is chosen from among the backups, and unchosen backups act as verifiers. The primary, the forwarder, and one of the verifiers perform a three-party QDS. The backups take turns acting as the forwarder. The arrow indicates the direction of the message delivery.

Table 1. Comparison between our work and detectable QBA framework. D-QBA, detectable QBA; N/A, not applicable; IC_1 and IC_2, two interactive consistency Byzantine conditions

Performance	This work	D-QBA [26–32]
Security analysis	Yes	N/A
Fault tolerance	$N \geq 2f+1, \forall f \in \mathbb{N}^+$	$N=3, f=1$
Message	Multiple	Binary
Decentralization	Yes	N/A
Entanglement	No	Yes
Strictly obey IC_1 and IC_2	Yes	No

Generally, our QBA framework consists of two phases, namely, broadcasting phase and gathering phase. Suppose there is a system of total N players including f malicious ones. The commanding general (initial primary) is denoted as S, and the lieutenants are denoted as R_i, for $i = 1, 2, \cdots, N-1$. The flow chart of the two phases is shown in Table 2. The broadcasting phase is designed for R_i to exchange the message received from S with each other, and the gathering phase is designed for R_i to deduce the original message of S according to the information gathered by themselves.

Broadcasting phase. The broadcasting phase consists of successive multicast rounds. For clarity, we denote the multicast round as $\mathcal{MR}^d_\zeta$, where ζ represents the route of delivering the message and d is the depth of the multicast round. The first multicast round started by the commanding general S is denoted as $\mathcal{MR}^1_S$. In $\mathcal{MR}^1_S$, S signs and then multicasts his message m^1_S to all the lieutenants R_i. In the multicast round of next depth, $\mathcal{MR}^2_{S\to R_i}$, R_i acts as a primary, and then signs and

Table 2. Protocol definition

Broadcasting phase

The broadcasting phase begins with $d = 1$ and ends with $d = f$. We consider a general case $\mathcal{MR}_{\zeta}^{d}$ (d=1, 2, 3, ⋯, f):

1. Sign and multicast The primary signs and then multicasts the message m_{ζ}^{d} to the N–d backups via QDS as shown in Fig. 1.

2. Consistency check If $d = 1$, there is no consistency check and the players will skip to step 3. If $1 < d \leq$ f, upon receiving the message from the primary, forwarder R_j checks the consistency between it and the message that he received from the primary at the previous depth $d - 1$. (R_j visits the players who do not appear in route ζ.) If consistency check is passed, perform step 3. Otherwise, he requests that this primary perform step 1 again until he receives a consistent message.

3. Forward R_j forwards the message to verifiers R_k (R_k visit the players except R_j and those who have appeared in route ζ).

4. Verify and record Forwarder R_j and verifier R_k verify the message and corresponding signature. When both of them accept, the signature is successful and they add this valid message m_{ζ}^{d} to their own broadcasting lists B_{ζ}^{d,R_j} and B_{ζ}^{d,R_k}, respectively.

5. Recursion The forwarder R_j acts as the primary of $\mathcal{MR}_{\zeta\to R_j}^{d+1}$, and then repeat the above four steps. The recursion process ends up when $d = f$.

Gathering phase

For the lieutenants R_i ($i = 1, 2, ..., N - 1$):

1. Input In the bottom layer $d = f$, R_i obtains the initial gathering lists $G_{\zeta}^{f,R_i} = B_{\zeta}^{f,R_i}$.

2. Recursion When $1 \leq d < f$, the gathering lists at the corresponding depth and route are $G_{\zeta}^{f,R_i} = \cup_{R_p}\left\{m_{\zeta\to R_p}^{d+1,R_i}\right\}$, where $m_{\zeta\to R_p}^{d+1,R_i} = \text{majority}\left(G_{\zeta\to R_p}^{d+1,R_i}\right)$ and R_p visits all players except those who have appeared in the route ζ.

3. Output $m_{S}^{1,R_i} = \text{majority}\left(G_{S}^{1,R_i}\right)$.

multicasts the message $m_{S\to R_i}^{2}$, which is he received from S, to the other lieutenants. The process will be repeated until $d = f$. We denote a list for the lieutenants R_i to record the messages received by him in $\mathcal{MR}_{\zeta}^{d}$ as aforementioned broadcasting list B_{ζ}^{d,R_i}.

In the broadcasting phase, the consistency check occurs between step 1 and step 3. Consider a general case: In $\mathcal{MR}_{\zeta}^{d}$, the primary signs and multicasts the message m_{ζ}^{d} to the backups, assuming that R_j acts as the forwarder and R_k acts as a verifier. In the multicast round at next depth $d + 1$, $\mathcal{MR}_{\zeta\to R_j}^{d+1}$, R_j will act as a primary and R_k will act as the forwarder. The messages R_j delivers to R_k in the two rounds, $\mathcal{MR}_{\zeta}^{d}$ and $\mathcal{MR}_{\zeta\to R_j}^{d+1}$, must be consistent, because R_k can check the consistency of the two messages. If the two messages are inconsistent, R_k will reject them and ask R_j to repeat the process until the two messages are consistent.

Gathering phase. The deterministic function we used in the gathering phase is called the majority function. It outputs the value of the majority element in the input set (see Materials and Methods). In $\mathcal{MR}_{\zeta}^{d}$, the gathering list held by the lieutenant R_i, denoted as G_{ζ}^{d,R_i}, is used for R_i to deduce the message delivered by the primary of $\mathcal{MR}_{\zeta}^{d}$. In the bottom layer $d = f$, R_i directly sets his or her own gathering list to $G_{\zeta}^{f,R_i} = B_{\zeta}^{f,R_i}$ and outputs $m_{\zeta}^{f,R_i} = \text{majority}\left(G_{\zeta}^{f,R_i}\right)$. Then, m_{ζ}^{f,R_i} becomes an element of the gathering list of $d = f - 1$. Considering general case where $1 \leq d < f$, all elements of R_i's gathering list G_{ζ}^{d,R_i} are deduced from the lists $G_{\zeta\to R_p}^{d+1,R_i}$ in multicast round $\mathcal{MR}_{\zeta\to R_p}^{d+1}$ (R_p visits all players who do not appear in route ζ). When $p = i$, this element is directly set as the message that R_i directly received from the primary in $\mathcal{MR}_{\zeta}^{d}$. With the recursive process, the gathering phase ends up when $d = 1$, and then R_i outputs $m_{S}^{1,R_i} = \text{majority}\left(G_{S}^{1,R_i}\right)$ as the final decision. Note that the broadcasting lists record the messages that are the lieutenants themselves actually received during the broadcast phase, and the gathering lists record the messages that are the lieutenants deduced according to the information of the previous depth. Only when $d = f$, the gathering lists are the same as the broadcasting lists, i.e., $G_{\zeta}^{f,R_i} = B_{\zeta}^{f,R_i}$.

Experimental implementation

We show proof-of-principle experimental implementation of our QBA framework for reaching consensus on a decentralized digital ledger, one of the most important application of blockchain. The digital ledger is a 1.10-MB document that is a virtual transaction including time, clients, merchants, commodity, and the amount. It is converted into a binary string of bits. We denote the correct message as *m1*, and the incorrect messages as *m2*, *m3*, and so on.

To show a high degree of adaptability of our work, we implement three-party consensus with single-bit GC01-QDS [43], one-time universal$_2$ hashing (OTUH) QDS [56], and OTUH-QDS without perfect keys [57], respectively (see Supplementary Materials Section B for details). In addition, we utilize OTUH-QDS to realize the five-party consensus. The key idea of single-bit GC01-QDS is to first generate two pairs of raw quantum keys and then exchange half of the Bob's and Charlie's quantum keys with each other, which is called symmetrization step, to construct the correlation. On the contrary, OTUH-QDS is to first construct the three-party correlation $X_a = X_b \oplus X_c$ and $Y_a = Y_b \oplus Y_c$ among the quantum keys of Alice (signer), Bob (forwarder), and Charlie (verifier). Alice signs the message with

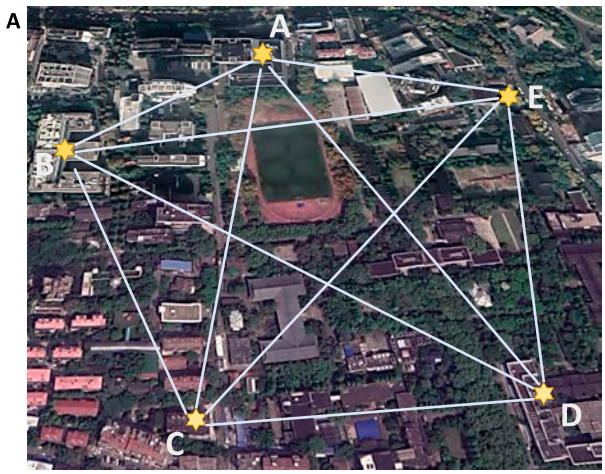
X_a and Y_a, and then Bob and Charlie exchange their keys to complete the verification. These correlated raw quantum keys can be achieved by any quantum key generation process [58–67]. Here, we utilize four-intensity decoy-state BB84 key generation process for the three QDS protocols [58].

There are five independent players S and R_i ($i = 1, 2, 3, 4$). The correlated quantum keys of different pairwise players are pre-distributed in the laboratory via fiber spool, i.e., the distribution stage of QDS is completed in the laboratory. They do not disclose any information of their own quantum keys to others, and then bring the keys to five different buildings in Fig. 2 to simulate the real-life situation where the users are geographically separated, and then use these quantum keys to complete digital signatures, i.e., classical operations of the messaging stage are performed in the real locations. Note that, to simplify the proof-of-principle experiment, we employ the above method due to the immaturity of real-life multi-node quantum networks. It is anticipated that as quantum networks progress in maturity, enabling their widespread deployment and utilization, our QBA framework can be seamlessly integrated into practical quantum networks without the necessity of laboratory-based quantum key preparation.

S, R_1, and R_2 perform three-party consensus, and all the five players perform five-party consensus. According to IC_1 and IC_2, we consider whether S is honest or not. Here, we exemplify the three-party and five-party consensus in Fig. 3. Moreover, we show the consensus rates of our QBA framework when adopting these three QDS in Table 3. OTUH-QDS, which can sign a multi-bit message each time, leads to much higher efficiency of QBA framework that the system can reach consensus 11.95 times per second, while single-bit GC01-QDS only reaches 4.5×10^{-8} times consensus per second under the same security parameter (see Supplementary Materials Section D for calculation details).

(a) The commanding general S is honest in three-party consensus. There is only one malicious player R_2 and thus only one layer $d = 1$ in the three-party consensus. In $\mathcal{MR}_S^1$, S sends correct message $m1$ via multicasting. R_1 records $m1$ when he acts as a forwarder, and records $m1$ received from R_2 when he acts as a verifier. The malicious R_2 must honestly forward $m1$ when he acts as a forwarder due to the unforgeability of QDS. Hence, as shown in Fig. 3A, the gathering list of honest R_1, which is also the broadcasting list, is $G_S^{1,R_1} = B_S^{1,R_1} = \{m1, m1\}$. The final output of R_1 is $m_S^{1,R_1} = \text{majority}\left(G_S^{1,R_1}\right) = m1$, which is consistent with the message sent by honest S. That satisfies IC_1.

(b) The commanding general S is dishonest in three-party consensus. There is only one malicious player S. In $\mathcal{MR}_S^1$, S sends conflicting messages $m1$ and $m2$ to honest R_1 and R_2, respectively. R_1 records $m1$ when he acts as a forwarder, and records $m2$ received from R_2 when he acts as a verifier. R_2



B

Link	A-B	A-C	A-D	A-E	B-C
Length (km)	25	35	40	50	25
Loss (dB)	6.35	8.90	9.35	10.35	6.35
QBER (%)	1.59	1.03	0.85	1.70	1.62
Key rate (kbps)	11.42	7.80	4.15	3.73	12.09

Link	B-D	B-E	C-D	C-E	D-E
Length (km)	35	40	25	25	35
Loss (dB)	8.90	9.35	6.35	6.35	8.90
QBER (%)	0.80	0.88	1.59	1.64	0.86
Key rate (kbps)	8.23	5.52	12.53	12.41	8.90

C

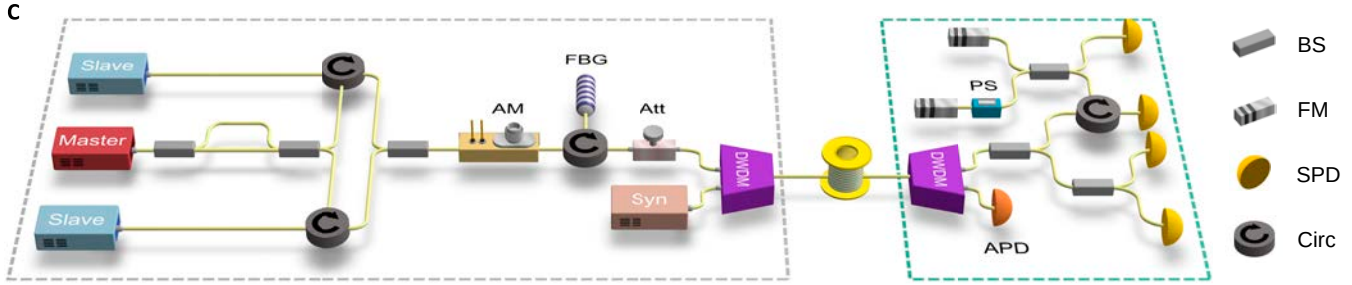


Fig. 2. Experimental implementation. (A) The five players bring their own pre-distributed correlated quantum keys to five different buildings and then perform the classical operations of the messaging stage. A, S; B, R_1; C, R_2; D, R_4; and E, R_5. In the messaging stage, the messages and corresponding signatures can be transmitted via authenticated classical channel. (B) Main parameters of BB84 key generation in the laboratory. QBER, quantum bit error rate. (C) Experimental setup of the four-intensity decoy-state quantum key generation system with a time-phase encoding. We take nodes A and B as an example. A uses a master laser, two slave lasers, and an asymmetric interferometer to prepare optical pulses in the Z and X bases. An intensity modulator is used for the decoy-state modulation. Before passing through a set of filters, a monitor and an attenuator are utilized to regulate the photon number per pulse. B uses a biased beam splitter for the passive basis detection. The pulses either go directly to the time detector or pass through an asymmetric interferometer. A synchronization signal is distributed from node A to node B through a wavelength division multiplexed quantum channel. BS, beam splitter; Circ, circulator; IM, intensity modulator; FBG, fiber Bragg grating; Att, attenuator; DWDM, dense wavelength division multiplexer; FM, Faraday mirror; PS, phase shifter; SPD, single-photon detector.

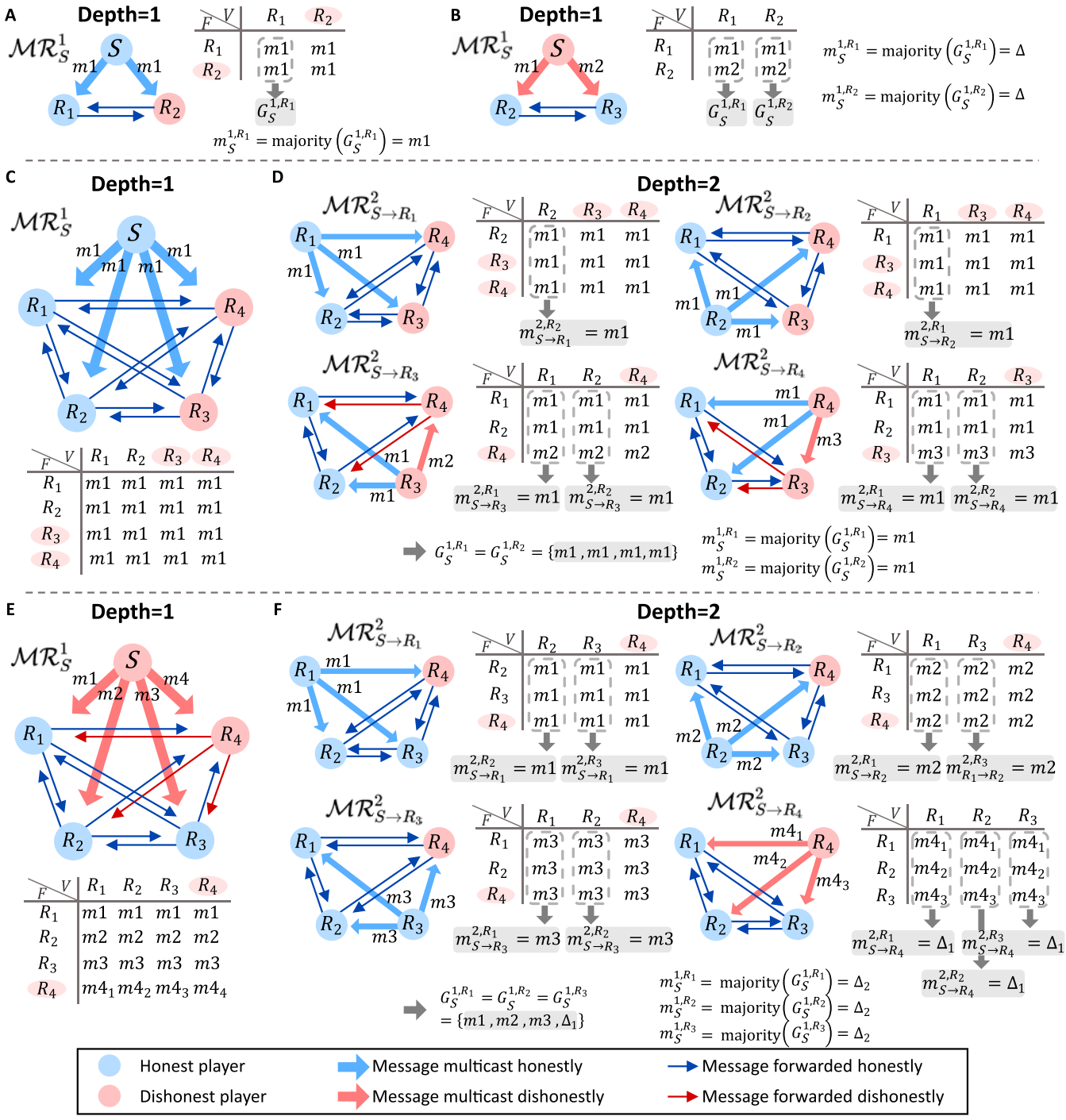


Fig. 3. Experimental results for three-party and five-party consensus. We use "F" to represent "forwarder" and "V" to represent "verifier" in the tables of lists. Each column of a table is a broadcasting list for the corresponding player. In the bottom layer $d=f$, R_i sets his or her gathering list as $G_\zeta^{f,R_i} = B_\zeta^{f,R_i}$ and performs the gathering phase to deduce the final output. (A) Multicast round at $d=1$ in three-party consensus with an honest primary. (B) Multicast round at $d=1$ in three-party consensus with a dishonest primary. Δ=majority($m1$,$m2$). (C and D) Multicast rounds at $d=1$ ($d=2$) in five-party consensus with an honest initial primary. (E and F) Multicast rounds at $d=1$ ($d=2$) in five-party consensus with a dishonest initial primary. Δ_1=majority($m4_1$,$m4_2$,$m4_3$) and Δ_2=majority($m1$,$m2$,$m3$,Δ_1).

records $m2$ when he acts as a forwarder, and records $m1$ received from R_1 when he acts as a verifier. Hence, as shown in Fig. 3B, the gathering list of honest R_1, which is also the broadcasting list, is $G_S^{1,R_1} = B_S^{1,R_1} = \{m1, m2\}$. The gathering list of honest R_2 is $G_S^{1,R_2} = B_S^{1,R_2} = \{m1, m2\}$, which is the same as that of R_1. The final outputs of R_1 and R_2 are both $m_S^{1,R_1} = m_S^{1,R_2} = \text{majority}(\{m1, m2\}) = \Delta$ Although the dishonest primary S sends conflicting messages, honest R_1 and R_2 obtain the same output Δ. That satisfies IC_2.

(c) The commanding general S is honest in five-party consensus. There are two malicious players R_3 and R_4 ($f=2$), and there are two layers $d=1$ and $d=2$ in five-party consensus.

Table 3. Consensus rates of our QBA framework adopting different QDS in the three-party consensus. The agreement rate, *CR*, is defined as the number of times a system can reach consensus per second. It can be expressed as $CR = \frac{SR}{C}$, where C is the communication complexity of the system and SR is the signature rate of adopted QDS (see Materials and Methods)

Different kinds of QDS	Consensus rate
GC01-QDS [43]	4.5×10^{-8}
OTUH-QDS [56]	11.95
OTUH-QDS without perfect keys [57]	6.12

Broadcasting phase. At depth $d=1$, in $\mathcal{MR}^1_S$, S broadcasts the correct message m_1 via multicast process. In $\mathcal{MR}^1_S$, every one must forward the correct message m_1 to honest players because S is honest and there are two honest players in QDS. Therefore, the broadcasting lists of honest R_1 and R_2 are both $B_S^{1,R_1}=B_S^{1,R_2}=\{m1,m1,m1,m1\}$ (see Fig. 3C).

The results of the multicast processes at depth $d=2$ are shown in Fig. 3D. In $\mathcal{MR}^2_{S\to R_1}$, because of the honest primary R_1 and Lemma 1 (see Materials and Methods for two lemmas), every one must forward the correct message m_1, and the broadcasting list of honest R_2 is $B^{2,R_2}_{S\to R_1}=\{m1,m1,m1\}$. In $\mathcal{MR}^2_{S\to R_2}$, because of the honest primary R_2 and Lemma 1, every one must forward the correct message *m1*, and the broadcasting list of honest R_1 is $B^{2,R_1}_{S\to R_2}=\{m1,m1,m1\}$ In $\mathcal{MR}^2_{S\to R_3}$, because of the dishonest primary R_3 and Lemma 2, only when dishonest R_3 and R_4 collude together can R_4 successfully forward the conflicting message *m2* to honest players. The broadcasting lists of honest R_1 and R_2 are both $B^{2,R_1}_{S\to R_3}=B^{2,R_2}_{S\to R_3}=\{m1,m1,m2\}$. Similar to the analysis of $\mathcal{MR}^2_{S\to R_3}$, only when dishonest R_3 and R_4 collude together, R_4 can successfully forward the conflicting message m_2 to honest players. The broadcasting lists of honest R_1 and R_2 are both $B^{2,R_1}_{S\to R_3}=B^{2,R_2}_{S\to R_3}=\{m1,m1,m3\}$.

Gathering phase. At the bottom depth $d=2$, each player obtains the initial gathering lists $G^{2,R_i}_{\zeta}=B^{2,R_i}_{\zeta}$.

For R_1 at depth $d=2$,

$$\begin{aligned}
G^{2,R_1}_{S\to R_2} &= B^{2,R_1}_{S\to R_2}=\{m1,m1,m1\},\\
G^{2,R_1}_{S\to R_3} &= B^{2,R_1}_{S\to R_3}=\{m1,m1,m2\},\\
G^{2,R_1}_{S\to R_4} &= B^{2,R_1}_{S\to R_4}=\{m1,m1,m3\},\\
m^{2,R_1}_{S\to R_1} &= m1,\\
m^{2,R_1}_{S\to R_2} &= \text{majority}\left(G^{2,R_1}_{S\to R_2}\right)=m1,\\
m^{2,R_1}_{S\to R_3} &= \text{majority}\left(G^{2,R_1}_{S\to R_3}\right)=m1,\\
m^{2,R_1}_{S\to R_4} &= \text{majority}\left(G^{2,R_1}_{S\to R_4}\right)=m1,
\end{aligned} \tag{1}$$

where the message $m^{2,R_1}_{S\to R_1}$ is directly received from S when R_1 acts as the forwarder. R_1 deduced that $m^{2,R_1}_{S\to R_2}$ is the message that R_2 received from S, $m^{2,R_1}_{S\to R_3}$ is the message that R_3 received from S, and $m^{2,R_1}_{S\to R_4}$ is the message that R_4 received from S. Then, $m^{2,R_1}_{S\to R_1}$, $m^{2,R_1}_{S\to R_2}$, $m^{2,R_1}_{S\to R_3}$, and $m^{2,R_1}_{S\to R_4}$ constitute the gathering list of R_1 at depth $d=1$, where $G^{1,R_1}_S=\left\{m^{2,R_1}_{S\to R_1},m^{2,R_1}_{S\to R_2},m^{2,R_1}_{S\to R_3},m^{2,R_1}_{S\to R_4}\right\}=\{m1,m1,m1,m1\}$. Thus, at depth $d=1$, R_1 obtains the final output

$$\begin{aligned}
m^{1,R_1}_S &= \text{majority}\left(G^{1,R_1}_S\right)\\
&= \text{majority}(\{m1,m1,m1,m1\})\\
&= m1.
\end{aligned} \tag{2}$$

For R_2 at depth $d=2$,

$$\begin{aligned}
G^{2,R_2}_{S\to R_1} &= B^{2,R_2}_{S\to R_1}=\{m1,m1,m1\},\\
G^{2,R_2}_{S\to R_3} &= B^{2,R_2}_{S\to R_3}=\{m1,m1,m2\},\\
G^{2,R_2}_{S\to R_4} &= B^{2,R_2}_{S\to R_4}=\{m1,m1,m3\},\\
m^{2,R_2}_{S\to R_1} &= \text{majority}\left(G^{2,R_2}_{S\to R_1}\right)=m1,\\
m^{2,R_2}_{S\to R_2} &= m1,\\
m^{2,R_2}_{S\to R_3} &= \text{majority}\left(G^{2,R_2}_{S\to R_3}\right)=m1,\\
m^{2,R_2}_{S\to R_4} &= \text{majority}\left(G^{2,R_2}_{S\to R_4}\right)=m1,
\end{aligned} \tag{3}$$

where the message $m^{2,R_2}_{S\to R_2}$ is directly received from S when R_2 acts as the forwarder. R_2 deduced that $m^{2,R_2}_{S\to R_1}$ is the message that R_1 received from S, $m^{2,R_2}_{S\to R_3}$ is the message that R_3 received from S, and $m^{2,R_2}_{S\to R_4}$ is the message that R_4 received from S. Then, $m^{2,R_2}_{S\to R_1}$, $m^{2,R_2}_{S\to R_2}$, $m^{2,R_2}_{S\to R_3}$, and $m^{2,R_2}_{S\to R_4}$ constitute the gathering list of R_2 at depth $d=1$, where $G^{1,R_2}_S=\left\{m^{2,R_2}_{S\to R_1},m^{2,R_2}_{S\to R_2},m^{2,R_2}_{S\to R_3},m^{2,R_2}_{S\to R_4}\right\}=\{m1,m1,m1,m1\}$. Thus, at depth $d=1$, R_2 obtains the final output

$$\begin{aligned}
m^{1,R_2}_S &= \text{majority}\left(G^{1,R_2}_S\right)\\
&= \text{majority}(\{m1,m1,m1,m1\})\\
&= m1.
\end{aligned} \tag{4}$$

The final outputs of R_1 and R_2 are both $m^{1,R_1}_S=m^{1,R_2}_S=\text{majority}\left(\{m_1,m_1,m_1m_1\}\right)=m_1$, which is consistent with the initial message sent by the honest primary S. This result satisfies IC_1.

(d) The commanding general S is dishonest in five-party consensus. There are two malicious players S and R_4 ($f=2$), and there are two layers $d=1$ and $d=2$ in five-party consensus.

Broadcasting phase. At depth $d=1$, in $\mathcal{MR}^1_S$, the dishonest primary S can broadcast the different messages *m1*, *m2*, and *m3* to honest R_1, R_2, and R_3, respectively, via multicast process. Additionally, due to Lemma 2, malicious S and R_4 can collude together and then R_4 can deliberately forward different messages $m4_1$, $m4_2$, and $m4_3$ to honest R_1, R_2, and R_3, respectively. Therefore, the broadcasting lists of honest R_1, R_2, and R_3 are $B_S^{1,R_1}=\{m1,m2,m3,m4_1\}$, $B_S^{1,R_2}=\{m1,m2,m3,m4_2\}$, and $=B_S^{1,R_3}=\{m1,m2,m3,m4_3\}$, respectively (see Fig. 3E).

The results of the multicast processes at depth $d=2$ are shown in Fig. 3F. In $\mathcal{MR}^2_{S\to R_1}$, because of the honest primary

R_1 and Lemma 1, every one must forward R_1's message m_1, and the broadcasting lists of honest R_2 and R_3 are both $B^{2,R_2}_{S\to R_1} = B^{2,R_3}_{S\to R_1} = \{m1, m1, m1\}$. In $\mathcal{MR}^2_{S\to R_2}$, because of the honest primary R_2 and Lemma 1, every one must forward R_1's message $m2$, and the broadcasting lists of honest R_1 and R_3 are both $B^{2,R_1}_{S\to R_2} = B^{2,R_3}_{S\to R_2} = \{m2, m2, m2\}$. In $\mathcal{MR}^2_{S\to R_3}$, because of the honest primary R_2 and Lemma 1, every one must forward R_3's message $m3$, and the broadcasting lists of honest R_1 and R_2 are both $B^{2,R_1}_{S\to R_3} = B^{2,R_2}_{S\to R_3} = \{m3, m3, m3\}$ In $\mathcal{MR}^2_{S\to R_4}$, because the dishonest primary R_4 can successfully broadcast the conflicting messages $m4_1$, $m4_2$, and $m4_3$ to honest players R_1, R_2, and R_3, respectively. The broadcasting lists of honest R_1, R_2, and R_3 are all $B^{2,R_1}_{S\to R_4} = B^{2,R_2}_{S\to R_4} = B^{2,R_3}_{S\to R_4} = \{m4_1, m4_2, m4_3\}$.

Gathering phase. At the bottom depth $d = 2$, each player obtains the initial gathering lists $G^{2,R_i}_{\zeta} = B^{2,R_i}_{\zeta}$.

For R_1 at depth $d = 2$,

$$\begin{aligned}
G^{2,R_1}_{S\to R_2} &= B^{2,R_1}_{S\to R_2} = \{m2, m2, m2\},\\
G^{2,R_1}_{S\to R_3} &= B^{2,R_1}_{S\to R_3} = \{m3, m3, m3\},\\
G^{2,R_1}_{S\to R_4} &= B^{2,R_1}_{S\to R_4} = \{m4_1, m4_2, m4_3\},\\
m^{2,R_1}_{S\to R_1} &= m1,\\
m^{2,R_1}_{S\to R_2} &= \text{majority}\left(G^{2,R_1}_{S\to R_2}\right) = m2,\\
m^{2,R_1}_{S\to R_3} &= \text{majority}\left(G^{2,R_1}_{S\to R_3}\right) = m3,\\
m^{2,R_1}_{S\to R_4} &= \text{majority}\left(G^{2,R_1}_{S\to R_4}\right) = \Delta_1,
\end{aligned} \tag{5}$$

where the message $m^{2,R_1}_{S\to R_1}$ is directly received from S when R_1 acts as the forwarder. R_1 deduced that $m^{2,R_1}_{S\to R_2}$ is the message that R_2 received from S, $m^{2,R_1}_{S\to R_3}$ is the message that R_3 received from S, and $m^{2,R_1}_{S\to R_4}$ is the message that R_4 received from S. Then, $m^{2,R_1}_{S\to R_1}$, $m^{2,R_1}_{S\to R_2}$, $m^{2,R_1}_{S\to R_3}$, and $m^{2,R_1}_{S\to R_4}$ constitute the gathering list of R_1 at depth $d = 1$, where $G^{1,R_1}_{S} = \left\{m^{2,R_1}_{S\to R_1}, m^{2,R_1}_{S\to R_2}, m^{2,R_1}_{S\to R_3}, m^{2,R_1}_{S\to R_4}\right\} = \{m1, m2, m3, \Delta_1\}$. Thus, at depth $d = 1$, R_1 obtains the final output

$$\begin{aligned}
m^{1,R_1}_{S} &= \text{majority}\left(G^{1,R_1}_{S}\right)\\
&= \text{majority}\left(\{m1, m2, m3\Delta_1\}\right)\\
&= \Delta_2.
\end{aligned} \tag{6}$$

For R_2 at depth $d = 2$,

$$\begin{aligned}
G^{2,R_2}_{S\to R_1} &= B^{2,R_2}_{S\to R_1} = \{m1, m1, m1\},\\
G^{2,R_2}_{S\to R_3} &= B^{2,R_2}_{S\to R_3} = \{m3, m3, m3\},\\
G^{2,R_2}_{S\to R_4} &= B^{2,R_2}_{S\to R_4} = \{m4_1, m4_2, m4_3\},\\
m^{2,R_2}_{S\to R_1} &= \text{majority}\left(G^{2,R_2}_{S\to R_2}\right) = m1,\\
m^{2,R_2}_{S\to R_2} &= m2,\\
m^{2,R_2}_{S\to R_3} &= \text{majority}\left(G^{2,R_2}_{S\to R_3}\right) = m3,\\
m^{2,R_2}_{S\to R_4} &= \text{majority}\left(G^{2,R_2}_{S\to R_4}\right) = \Delta_1,
\end{aligned} \tag{7}$$

where the message $m^{2,R_2}_{S\to R_2}$ is directly received from S when R_2 acts as the forwarder. R_2 deduced that $m^{2,R_2}_{S\to R_1}$ is the message that R_1 received from S, $m^{2,R_2}_{S\to R_3}$ is the message that R_3 received from S, and $m^{2,R_2}_{S\to R_4}$ is the message that R_4 received from S. Then, $m^{2,R_2}_{S\to R_1}$, $m^{2,R_2}_{S\to R_2}$, $m^{2,R_2}_{S\to R_3}$, and $m^{2,R_2}_{S\to R_4}$ constitute the gathering list of R_2 at depth $d = 1$, where $G^{1,R_2}_{S} = \left\{m^{2,R_2}_{S\to R_1}, m^{2,R_2}_{S\to R_2}, m^{2,R_2}_{S\to R_3}, m^{2,R_2}_{S\to R_4}\right\} = \{m1, m2, m3, \Delta_1\}$. Thus, at depth $d = 1$, R_2 obtains the final output

$$\begin{aligned}
m^{1,R_2}_{S} &= \text{majority}\left(G^{1,R_2}_{S}\right)\\
&= \text{majority}\left(\{m1, m2, m3, \Delta_1\}\right)\\
&= \Delta_2.
\end{aligned} \tag{8}$$

For R_3 at depth $d = 2$,

$$\begin{aligned}
G^{2,R_3}_{S\to R_1} &= B^{2,R_2}_{S\to R_1} = \{m1, m1, m1\},\\
G^{2,R_3}_{S\to R_2} &= B^{2,R_2}_{S\to R_2} = \{m2, m2, m2\},\\
G^{2,R_3}_{S\to R_4} &= B^{2,R_2}_{S\to R_4} = \{m4_1, m4_2, m4_3\},\\
m^{2,R_3}_{S\to R_1} &= \text{majority}\left(G^{2,R_3}_{S\to R_2}\right) = m1,\\
m^{2,R_3}_{S\to R_2} &= \text{majority}\left(G^{2,R_3}_{S\to R_2}\right) = m2,\\
m^{2,R_3}_{S\to R_3} &= m3,\\
m^{2,R_3}_{S\to R_4} &= \text{majority}\left(G^{2,R_3}_{S\to R_4}\right) = \Delta_1,
\end{aligned} \tag{9}$$

where the message $m^{2,R_2}_{S\to R_2}$ is directly received from S when R_3 acts as the forwarder. R_3 deduced that $m^{2,R_3}_{S\to R_1}$ is the message that R_1 received from S, $m^{2,R_3}_{S\to R_2}$ is the message that R_2 received from S, and $m^{2,R_3}_{S\to R_4}$ is the message that R_4 received from S. Then, $m^{2,R_3}_{S\to R_1}$, $m^{2,R_3}_{S\to R_2}$, $m^{2,R_3}_{S\to R_3}$, and $m^{2,R_3}_{S\to R_4}$ constitute the gathering list of R_3 at depth $d = 1$, where $G^{1,R_3}_{S} = \left\{m^{2,R_3}_{S\to R_1}, m^{2,R_3}_{S\to R_2}, m^{2,R_3}_{S\to R_3}, m^{2,R_3}_{S\to R_4}\right\} = \{m1, m2, m3, \Delta_1\}$. Thus, at depth $d = 1$, R_3 obtains the final output

$$\begin{aligned}
m^{1,R_3}_{S} &= \text{majority}\left(G^{1,R_3}_{S}\right)\\
&= \text{majority}\left(\{m1, m2, m3, \Delta_1\}\right)\\
&= \Delta_2.
\end{aligned} \tag{10}$$

The final outputs of honest R_1, R_2, and R_3 are all $m^{1,R_1}_{S} = m^{1,R_2}_{S} = m^{1,R_3}_{S} = \Delta_2$. Although the dishonest primary S sends conflicting messages, the honest players R_1, R_2, and R_3 obtain the same output Δ_2. This result satisfies IC_2.

In conclusion, our experimental results show that in real three-party and five-party consensus, our protocol can not only satisfy the two original Byzantine conditions IC_1 and IC_2 but also achieve the fault tolerance of $N \geq 2f + 1$, which breaks the 1/3 fault-tolerance lower bound.

Discussion

The 1/3 fault-tolerance bound cannot be beaten for any arbitrary pairwise communication [10–13]; not even quantum channels can

help solve this problem. If the nodes of a system are linked by the channels that are independent of each other, the bound is unable to be beaten. Intriguingly, when quantum entanglement is introduced into the system, it is possible to surpass this bound because quantum entanglement provides the correlation and removes the independence [26]. Although detectable QBA framework is designed according to multi-particle entanglement [26–32], they cannot extend to more than three participants and unavoidably weaken the Byzantine conditions, because multi-particle entangled states are very hard to prepare and maintain, and these protocols do not fully utilize the correlation to protect the unforgeability and nonrepudiation that leads to a certain probability of failure.

Is quantum entanglement necessary or can weaker multiparty correlation work in Byzantine agreement? Fortunately, QDS is a useful tool for solving this problem due to its asymmetric relationship among three players. In addition, three-party QDS is naturally decentralized due to its structure without a fully trusted third party. The two essential properties of QDS, unforgeability and nonrepudiation, effectively curtail the malevolent activities of malicious players within the system, preventing them from deliberately delivering conflicting messages. Asymmetric relationship of QDS makes the channels no longer independent of each other. Consequently, our protocol can break the 1/3 fault-tolerance bound.

Note that the most important thing to break the fault-tolerance bound is to provide a decentralized multiparty correlation to remove the independence of pairwise channels. Intriguingly, quantum entanglement and asymmetric relationship of QDS both satisfy the above requirement. In addition, if we can find a three-party information-theoretically secure classical digital signature scheme that has the same decentralized structure as QDS [68], our framework can also break the bound. However, up to now, we do not find classical correlation greater than QDS without any additional assumptions because these classical schemes require extra assumptions such as the existence of a trusted third party and authenticated broadcast channels, which disobeys decentralization of Byzantine agreement [69]. By bridging two prominent research themes, the Byzantine agreement and QDS, our work paves the way for practical quantum blockchain and quantum consensus networks.

In the end, we want to highlight that although our QBA framework utilizes QDS, it is still unable to surpass the famous blockchain trilemma (see Materials and Methods).The blockchain trilemma highlights the intricate balance required among three fundamental attributes: decentralization, security, and scalability. Within our QBA framework, we have achieved notable success in security, surpassing the 1/3 fault-tolerance bound while maintaining information-theoretical security. Moreover, our approach has full decentralization. However, it falls short in scalability, exhibiting exponential communication complexity, as shown in Eq. 11. Consequently, our research remains bound by the constraints posed by the blockchain trilemma. Intriguing inquiries linger as we ponder the blockchain trilemma's resilience to quantum resources. Is it an irrefutable theorem or an assailable postulate? The possibility of quantum resources challenging the blockchain trilemma beckons us toward further scholarly exploration.

Materials and Methods

Quantum digital signatures

QDSs with information-theoretical security have two major properties, nonrepudiation and unforgeability. They are all divided into the two stages, distribution stage and messaging stage. The distribution stage is to distribute the correlated raw quantum keys of Alice–Bob and Alice–Charlie for the messaging stage. The correlated quantum keys can be achieved by some classical operations, such as symmetrization step used in CV-QDS [49], BB84 GC01-QDS [43], and MDI-QDS [45,51], test bits used in SARG04-QDS [52,54,55], and secret sharing used in OTUH-QDS [56,57]. The messaging stage is to complete the digital signature to determine whether it is successful or not.

The brief process of messaging stage can be described as follows. Alice is a "signer." Bob is a "forwarder." Charlie is a "verifier." Alice signs a message with her quantum keys and then transmits the message and corresponding signature to Bob. Bob forwards the message and corresponding signature to the verifier Charlie. Then, Bob and Charlie will check the message and corresponding signature, respectively. The process of QDS is successful when and only when both Bob and Charlie accept the message and the corresponding signature. The signature rate, *SR*, is defined as the number of times the players can perform the QDS per second.

Nonrepudiation. Nonrepudiation refers to a situation in which the signer cannot successfully dispute the authorship of his signature. This means that Alice cannot deny the fact that she signed the message if the signature is accepted by both Bob and Charlie.

Unforgeability. Unforgeability refers to a situation in which no one can forge a message and its corresponding signature. This means that if Bob forwards a forged message and signature, it will be impossible for him to successfully make Charlie accept the forged message and signature.

Majority function

The majority function we apply in our protocol aims to output the element that appears most often for an input set. For example, when the input set is $M = \{m_1, m_1, m_1, m_2, m_2\}$, the output will be majority$(M) = m_1$. In a few cases, more than one element appears most frequently in the input set, and the systems that calculate the majority function on the input set are often deliberately biased toward one of them that we set initially. For example, when the input set is $M = \{m_1, m_1, m_1, m_2, m_2, m_2\}$, the output will be majority$(M) = m_1\ (m_2)$, which is determined by the biased output $m_1\ (m_2)$ that we set before calculation. Note that for the same input sets with different players, the majority function outputs the same value, which we will denote as Δ.

Communication complexity and consensus rate

To measure the consumed resources, we define the number of times the QDS process is implemented to reach consensus as the communication complexity, denoted as *C*. The total communication complexity of our QBA protocol can be expressed by

$$\begin{aligned} C &= A_{N-1}^2 + (N-1)A_{N-2}^2 + (N-2)A_{N-3}^2 \\ &\quad + \cdots + (N-1)(N-2)\cdots\left(N-f+1\right)A_{N-f}^2 \\ &= \sum_{m=0}^{f-1} A_{N-1}^{2+m}, \end{aligned} \tag{11}$$

where $A_a^b = \frac{a!}{(a-b)!}$ is b-permutations of a, f is the number of dishonest players, and N is the number of all players. Here, A_{n-1}^{2+m} represents the communication complexity at depth

$m+1$. In reality, we need to perform $\left[\frac{N-1}{2}\right]$ recursions for a real-life consensus system with unknown f, where $[x]$ is the greatest integer less than or equal to x.

Consider the simple case that the system uses the QDS protocol, which has the same signature rate, denoted as SR, in all the multicast rounds. We define the consensus rate of our QBA protocol as

$$CR = \frac{SR}{C} = \frac{SR}{\sum_{m=0}^{f-1} A_{N-1}^{2+m}}, \tag{12}$$

where C is the communication complexity of the system. CR is the important index to indicate the efficiency of QBA. To get the higher consensus rate, we need to adopt the QDS protocol that has the higher signature rate. We can find that as the total number of players and the number of malicious players increase, the communication complexity will increase, which leads to the decrease of the consensus rate.

Two important lemmas in security analysis

In our protocol, the performance of honest (dishonest) players follows the same rule. Therefore, the players can be divided into two groups, the honest and dishonest. Also, the elements of a gathering list can be divided in the same way. Therefore, we can simplify the protocol with a perfect binary tree model where one tree node represents the set of multicast rounds with honest or dishonest primaries. The left (right) child tree node represents the multicast rounds with honest (dishonest) primaries of the next depth. In what follows, when we say a tree node is honest (dishonest), it means that the primaries in this tree node is honest (dishonest). We can obtain the important Lemma 1 and Lemma 2. The proofs for two lemmas and complete security analysis for our protocol can be found in Supplementary Materials Section C in detail.

Lemma 1: Suppose that B is a right child tree node of a parent node A who is honest, and C is the left child tree node of B. The messages delivered in C are consistent with those of A, which protects the consistency of the delivered messages.

Lemma 2: Suppose that B is a right child tree node of a parent node A who is honest, and E is the right child tree node of B. The message multicast in E can be inconsistent with those of A, which disrupts the consistency of the delivered messages.

Experimental setting

The master laser generates phase-randomized 1.6-ns-wide laser pulses with a repetition rate of 100 MHz at 1550.12 nm. The system frequency is 100 MHZ, but due to the 400-ns dead time every 10 us, the effective frequency of optical pulse is 96 MHz. Two pairs of pulses with relative phases 0 and π at a 2-ns time delay generated by an asymmetric interferometer are injected into two slave lasers through the optical circulator, respectively. By controlling the trigger electrical signal of two slave lasers, Alice randomly prepares quantum states in the Z (time) and X (phase) bases by using 400-ps-wide slave laser pulses. The programmable delay chip with a 10-ps timing resolution is used to calibrate the time consistency. The spectral consistency is naturally satisfied because of the laser seeding technique [70]. A 50-GHz nominal bandwidth fiber Bragg grating is used to remove extra spurious emission and precompensate for the pulse broadening in the fiber transmission. The 2-ns-wide synchronization pulses with repetition rates of 100 kHz are transmitted via the quantum channel using multiplexed wavelength division. The intensities are set as $\mu = 0.40$, $\nu = 0.20$, $\omega = 0.4$, and 0 with the corresponding probabilities $p_\mu = 0.60$, $p_\nu = 0.20$, $p_\omega = 0.15$, and $p_0 = 0.05$, respectively. If trigger signal is not provided to the slave laser, the vacuum state is generated. The amplitude modulator generates two different intensities, and the intensity of ω is double that of ν ($\omega = 2\nu$) since it has two pulses in the X basis. At the receiving end, a 30:70 biased beam splitter is used to perform passive basis detection after a wavelength division demultiplexer. A probability of 30% is measured in the phase basis, and the probability of 70% is used to receive in the time basis. A Faraday–Michelson interferometer is used for the phase measurement, in which phase drift is compensated in real time by using the phase shifter. The total insertion losses of the time and phase bases are 4.25 and 8 dB, respectively. The efficiency of single-photon detectors is 20% at a 160 dark count per second. To decrease the after-pulse probability, we set the dead times to 10 μs for the links.

Blockchain trilemma

Blockchain possesses three crucial attributes: decentralization, security, and scalability. Decentralization forms the fundamental core of blockchain technology, emphasizing its inherent nature. Security stands as a paramount concern in any blockchain system, while scalability presents a formidable challenge. However, the blockchain trilemma emerges from the inherent difficulty of achieving a harmonious balance among these three essential elements. Blockchains are often forced to make trade-offs that prevent them from achieving all the three aspects. Note that the trilemma is just a model to conceptualize the various challenges facing blockchain technology. There is no strict proof that the three aspects cannot be achieved. However, to date, there are no protocols able to break down the trilemma. The design needs to weaken the requirements for a certain feature.

Acknowledgments

We thank Y. Fu and M.-G. Zhou for their valuable discussions.
Funding: This study was supported by the National Natural Science Foundation of China (no. 12274223), the Natural Science Foundation of Jiangsu Province (no. BK20211145), the Fundamental Research Funds for the Central Universities (no. 020414380182), the Key Research and Development Program of Nanjing Jiangbei New Area (no. ZDYD20210101), and the Program for Innovative Talents and Entrepreneurs in Jiangsu (no. JSSCRC2021484).
Author contributions: H.-L.Y. and Z.-B.C. conceived the research. C.-X.W., R.-Q.G., and H.-L.Y. developed the quantum Byzantine agreement. R.-Q.G., C.-X.W., and H.-L.Y. provided the security proof. Y.B., Y.-S.L., and H.-L.Y. performed the quantum communication network. R.-Q.G., C.-X.W., and H.-L.Y. performed quantum consensus of digital ledger. C.-X.W., R.-Q.G., and H.-L.Y. cowrote the manuscript, with input from the other authors. All authors have discussed the results and proofread the manuscript.
Competing interests: The authors declare that they have no competing interests.

Data Availability

Data generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Supplementary Materials

Sections A to D
Figs. S1 to S4

References

1. Lamport L, Shostak R, Pease M. The Byzantine Generals Problem. *ACM Trans Program Lang Syst*. 1982;4:382–401.
2. Extance A. The future of cryptocurrencies: Bitcoin and beyond. *Nature*. 2015;7571(526):21–23.
3. Castro M, Liskov B. Practical Byzantine fault tolerance. In: OSDI Vol. 99. 1999:173–86.
4. Castro M, Liskov B. Practical Byzantine fault tolerance and proactive recovery. *ACM Trans Comput Syst*. 2002;20:398–461.
5. Aublin PL, Mokhtar SB, Quéma V. RBFT: Redundant Byzantine fault tolerance. Paper presented at: 2013 IEEE 33rd International Conference on Distributed Computing Systems; 2013; Philadelphia, PA, USA.
6. Miller A, Xia Y, Croman K, Shi E, Song D. The honey badger of BFT protocols. Paper presented at: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16. Association for Computing Machinery; 2016; New York, NY, USA.
7. Yin M, Malkhi D, Reiter MK, Gueta GG, Abraham I. HotStuff: BFT consensus with linearity and responsiveness. Paper presented at: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing. PODC '19. Association for Computing Machinery; 2019; New York, NY, USA.
8. Guo B, Lu Z, Tang Q, Xu J, Zhang Z. Dumbo: Faster asynchronous BFT protocols. Paper presented at: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. CCS '20. Association for Computing Machinery; 2020:803–818.
9. Lu Y, Lu Z, Tang Q, and Wang G. Dumbo-MVBA: Optimal multi-valued validated asynchronous Byzantine agreement, revisited. Paper presented at: Proceedings of the 39th Symposium on Principles of Distributed Computing. PODC '20. Association for Computing Machinery; 2020; New York, NY, USA.
10. Pease M, Shostak R, Lamport L. Reaching agreement in the presence of faults. *J ACM*. 1980;27:228–234.
11. Dolev D, Halpern JY, Strong HR. On the possibility and impossibility of achieving clock synchronization. *J Comput Syst Sci*. 1986;32:230–250.
12. Fischer MJ, Lynch NA, Merritt M. Easy impossibility proofs for distributed consensus problems. *Distrib Comput*. 1986;1:26–39.
13. Fitzi M, Garay J, Maurer U, Ostrovsky R. *Advances in Cryptology-CRYPTO 2001*: Proceedings of the 21-st Annual International Cryptology Conference, Santa Barbara, CA, 2001. Lecture Notes in Computer Science (Springer, Berlin, New York, 2001).
14. Kiktenko EO, Pozhar NO, Anufriev MN. Quantum-secured blockchain. *Quant Sci Technol*. 2018;3(3):Article 035004.
15. Menezes AJ, Van Oorschot PC, Vanstone SA. *Handbook of applied cryptography*. Boca Raton (FL): CRC Press; 2018.
16. Shor P. Algorithms for quantum computation: Discrete logarithms and factoring. Paper presented at: Proceedings 35th Annual Symposium on Foundations of Computer Science; 1994; Santa Fe, NM, USA.
17. Grover LK. Quantum mechanics helps in searching for a needle in a haystack. *Phys Rev Lett*. 1997;79(2):325.
18. Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, Biswas R, Boixo S, Brandao FGSL, Buell DA, et al. Quantum supremacy using a programmable superconducting processor. *Nature*. 2019;574(7779):505–510.
19. Fedorov AK, Kiktenko EO, Lvovsky AI. Quantum computers put blockchain security at risk. *Nature*. 2018;563(7732): 465–467.
20. Wei S, Li H, Long G. A full quantum eigensolver for quantum chemistry simulations. *Research*. 2020;2020:1486935.
21. Fernández-Caramès TM, Fraga-Lamas P. Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. *IEEE Access*. 2020;8:21091–21116.
22. Zhou MG, Cao XY, Lu YS, Wang Y, Bao Y, Jia ZY, Fu Y, Yin HL, Chen ZB. Experimental quantum advantage with quantum coupon collector. *Research*. 2022;2022:9798679.
23. Huang S, Yin HL, Chen ZB, Wu S. Quantum-accelerated algorithms for generating random primitive polynomials over finite fields. arXiv preprint arXiv:2203.12884. 2022.
24. Pan N, Chen T, Sun H, Zhang X. Electric-circuit realization of fast quantum search. *Research*. 2021;2021:9793071.
25. Long GL. Toward applications of cloud quantum computation. *Sci Chin Phys Mech Astron*. 2022;65:Article 110361.
26. Fitzi M, Gisin N, Maurer U. Quantum solution to the Byzantine agreement problem. *Phys Rev Lett*. 2001;87(21):Article 217901.
27. 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(7):070504.
28. Fitzi M, Gottesman D, Hirt M, Holenstein T, Smith A. Detectable Byzantine agreement secure against faulty majorities. Paper presented at: Proceedings of the Twenty-First Annual Symposium on Principles of Distributed Computing; 2002; New York, NY, USA.
29. Iblisdir S, Gisin N. Byzantine agreement with two quantum-key-distribution setups. *Phys Rev A*. 2004;70(3):034306.
30. Neigovzen R, Rodó C, Adesso G, Sanpera A. Multipartite continuous-variable solution for the Byzantine agreement problem. *Phys Rev A*. 2008;77(6):062307.
31. Rahaman R, Wieśniak M, Żukowski M. Quantum Byzantine agreement via Hardy correlations and entanglement swapping. *Phys Rev A*. 2015;92(4):042302.
32. Smania M, Elhassan AM, Tavakoli A, Bourennane M. Experimental quantum multiparty communication protocols. *npj Quantum Inf*. 2016;2:16010.
33. Ben-Or M, Hassidim A. Fast quantum Byzantine agreement. In: *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*. New York (NY): Association for Computing Machinery, 2005:481–485.
34. Taherkhani MA, Navi K, Van Meter R. Resource-aware system architecture model for implementation of quantum aided Byzantine agreement on quantum repeater networks. *Quantum Sci Technol*. 2018;3(1):Article 014011.
35. Sun X, Kulicki P, Sopek M. Multi-party quantum Byzantine agreement without entanglement. *Entropy*. 2020;22(10):1152.
36. Wang W, Yu Y, Du L. Quantum blockchain based on asymmetric quantum encryption and a stake vote consensus algorithm. *Sci Rep*. 2022;12(1):8606.
37. Gao F, Guo FZ, Wen QY, Zhu FC. Comment on "Experimental demonstration of a quantum protocol for Byzantine agreement and liar detection". *Phys Rev Lett*. 2008;101(20):Article 208901.

38. Gaertner S, Bourennane M, Kurtsiefer C, Cabello A, Weinfurter H. Gaertner et al. Reply. *Phys Rev Lett*. 2008;101(20):Article 208902.
39. Kleinberg J, Tardos E. *Algorithm design*. Noida (Uttar Pradesh, India): Pearson Education India; 2006.
40. Gottesman D and Chuang I. Quantum digital signatures. arXiv preprint quant-ph/0105032. 2001.
41. Dunjko V, Wallden P, Andersson E. Quantum digital signatures without quantum memory. *Phys Rev Lett*. 2014;112(4):040502.
42. Roehsner MC, Kettlewell JA, Batalhão TB, Fitzsimons JF, Walther P. Quantum advantage for probabilistic one-time programs. *Nat Commun*. 2018;9(1):5225.
43. Amiri R, Wallden P, Kent A, Andersson E. Secure quantum signatures using insecure quantum channels. *Phys Rev A*. 2016;93(3):032325.
44. Puthoor IV, Amiri R, Wallden P, Curty M, Andersson E. Measurement-device-independent quantum digital signatures. *Phys Rev A*. 2016;94(2):Article 022328.
45. Roberts G, Lucamarini M, Yuan Z, Dynes JF, Comandar LC, Sharpe AW, Shields AJ, Curty M, Puthoor IV, Andersson E. Experimental measurement-device-independent quantum digital signatures. *Nat Commun*. 2017;8(1):1098.
46. Collins RJ, Amiri R, Fujiwara M, Honjo T, Shimizu K, Tamaki K, Takeoka M, Sasaki M, Andersson E, Buller GS. Experimental demonstration of quantum digital signatures over 43 dB channel loss using differential phase shift quantum key distribution. *Sci Rep*. 2017;7:3235.
47. An XB, Zhang H, Zhang CM, Chen W, Wang S, Yin ZQ, Wang Q, He DY, Hao PL, Liu SF, et al. Practical quantum digital signature with a gigahertz BB84 quantum key distribution system. *Opt Lett*. 2019;44(1):139–142.
48. Thornton M, Scott H, Croal C, Korolkova N. Continuous-variable quantum digital signatures over insecure channels. *Phys Rev A*. 2019;99(3):Article 032341.
49. Richter S, Thornton M, Khan I, Scott H, Jaksch K, Vogl U, Stiller B, Leuchs G, Marquardt C, Korolkova N. Agile and versatile quantum communication: Signatures and secrets. *Phys Rev X*. 2021;11(1):Article 011038.
50. Qin JQ, Jiang C, Yu YL, Wang XB. Quantum digital signatures with random pairing. *Phys Rev Appl*. 2022;17(4):Article 044047.
51. Yin HL, Wang WL, Tang YL. Experimental measurement-device-independent quantum digital signatures over a metropolitan network. *Phys Rev A*. 2017;95(4):042338.
52. Yin HL, Fu Y, Chen ZB. Practical quantum digital signature. *Phys Rev A*. 2016;93(3):032316.
53. Yin HL, Fu Y, Liu H, Tang QJ, Wang J, You LX, Zhang WJ, Chen SJ, Wang Z, Zhang Q, et al. Experimental quantum digital signature over 102 km. *Phys Rev A*. 2017;95(3):Article 032334.
54. Lu YS, Cao XY, Weng CX, Gu J, Xie YM, Zhou MG, Yin HL, Chen ZB. Efficient quantum digital signatures without symmetrization step. *Opt Express*. 2021;29(7):10162–10171.
55. Weng CX, Lu YS, Gao RQ, Xie YM, Gu J, Li CL, Li BH, Yin HL, Chen ZB. Secure and practical multiparty quantum digital signatures. *Opt Express*. 2021;29(17):27661–27673.
56. Yin HL, Fu Y, Li CL, Weng CX, Li BH, Gu J, Lu YS, Huang S, Chen ZB. Experimental quantum secure network with digital signatures and encryption. *Natl Sci Rev*. 2023;10(4):nwac228.
57. Li BH, Xie YM, Cao XY. One-time universal hashing quantum digital signatures without perfect keys. *Phys Rev Appl*. 2023;20(4):044011.
58. Yin HL, Liu P, Dai WW, Ci ZH, Gu J, Gao T, Wang QW, Shen ZY. Experimental composable security decoy-state quantum key distribution using time-phase encoding. *Opt Express*. 2020;28(20):29479–29485.
59. Xu F, Ma X, Zhang Q, Lo HK, Pan JW. Secure quantum key distribution with realistic devices. *Rev Mod Phys*. 2020;92(2):Article 025002.
60. Pirandola S, Andersen UL, Banchi L, Berta M, Bunandar D, Colbeck R, Englund D, Gehring T, Lupo C, Ottaviani C, et al. Advances in quantum cryptography. *Adv Opt Photon*. 2020;12(4):1012–1236.
61. Liu WB, Li CL, Xie YM, Weng CX, Gu J, Cao XY, Lu YS, Li BH, Yin HL, Chen ZB. Homodyne detection quadrature phase shift keying continuous-variable quantum key distribution with high excess noise tolerance. *PRX Quantum*. 2021;2(4):Article 040334.
62. Lo HK, Curty M, Qi B. Measurement-device-independent quantum key distribution. *Phys Rev Lett*. 2012;108(13): Article 130503.
63. Lucamarini M, Yuan ZL, Dynes JF, Shields AJ. Overcoming the rate–distance limit of quantum key distribution without quantum repeaters. *Nature*. 2018;557(7705):400–403.
64. Xie YM, Lu YS, Weng CX, Cao XY, Jia ZY, Bao Y, Wang Y, Fu Y, Yin HL, Chen ZB. Breaking the rate-loss bound of quantum key distribution with asynchronous two-photon interference. *PRX Quantum*. 2022;3(2):Article 020315.
65. Fu Y, Yin HL, Chen TY, Chen ZB. Long-distance measurement-device-independent multiparty quantum communication. *Phys Rev Lett*. 2015, 114;9:090501.
66. Cao Z, Lu Y, Chai G, Yu H, Liang K, Wang L. Realization of quantum secure direct communication with continuous variable. *Research*. 2023;6:0193.
67. Shen A, Cao XY, Wang Y, Fu Y, Gu J, Liu WB, Weng CX, Yin HL, Chen ZB. Experimental quantum secret sharing based on phase encoding of coherent states. *Sci Chin Phys, Mech Astron*. 2023;66:Article 260311.
68. Amiri R, Abidin A, Wallden P, Andersson E. Efficient unconditionally secure signatures using universal hashing. In: Preneel B and Vercauteren F, editors. *Applied cryptography and network security*. Cham (Switzerland): Springer International Publishing; 2018:143–162.
69. Wallden P, Dunjko V, Kent A, Andersson E. Quantum digital signatures with quantum-key-distribution components. *Phys Rev A*. 2015;4(91):042304.
70. Comandar L, Lucamarini M, Fröhlich B, Dynes JF. Quantum key distribution without detector vulnerabilities using optically seeded lasers. *Nat Photonics*. 2016;10(5):312–315.